EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Stock Market Analysis using Long Short-Term Model

Pulkit Gupta[1, *], Suhani Malik[2], Kumar Apoorb[3], Syed Mahammed Sameer[4], Vivek Vardhan[5] and Prashanth Ragam[6]

[1,2,3,4,5,6]School of Computer Science & Engineering, Vellore Institute of Technology, Amaravati, Andhra Pradesh, India

Abstract

In today's world of value and improved investments, financial analysis has become a difficult task. The implementation of recurrent neural networks (RNN) and long short-term memory (LSTM) cells for stock market forecasting using time series of historical portfolio stock data is demonstrated in this study. In this study, we applied LSTM to predict stock market values using Yahoo Finance data along with Python modules Pandas and Matplotlib to evaluate the performance of the model. Our results show that the LSTM model is able to make accurate predictions of stock market prices and trends using historical data. The results of the correlation study showed a significant relationship between the daily return and the closing price of four randomly chosen companies. Overall, using LSTM, Yahoo Finance, Python Pandas, and Matplotlib modules to predict stock prices and provide useful information to investors was a successful strategy.

Keywords: Long Short-Term Model, Yahoo finance, stock market return, average

Received on 20 September 2023, accepted on 15 November 2023, published on 22 November 2023.

doi: 10.4108/eetsis.4446

*Corresponding author. Email: pulkit92003@gmail.com

1. Introduction

One of the world's most unreliable financial systems is the stock market. Stock prices are influenced by a wide range of variables, such as business performance, economic data, investor attitude, and geopolitical developments. Recurrent neural networks with the ability to learn and recall long-term dependencies include LSTM networks [1] Speech recognition, natural language processing, and financial forecasting are just a few of the time series prediction tasks. We will talk about the application of LSTM networks to stock market forecasting in this study [3]. In this paper we will have a look at four stock analysis namely Tesla Inc., Uber Inc., Nvidia Corp. and Meta Inc.

2. Stock Market & LSTM

The idea of stock analysis is predicated on the belief that the intrinsic value of a stock may be ascertained using the information currently accessible on the market. Multiple strategies have been developed by traders to forecast future stock prices: Fundamental stock analysis examines the business value of the stock by taking into account the variables such as revenue, operating costs, assets, goods, management, rivals, and the general business climate. Technical analysis is predicated on the idea that markets frequently move in cycles and that precise forecasts may be generated by examining historical price patterns and price trends. Statistical tools have been used by traders to examine price and volume data. They can now analyze bigger datasets thanks to recent developments in computing and machine learning. [3]

Recurrent neural networks are a type of long short-term memory. In an RNN, the output from the previous step is fed into the current step as input. RNN's performance becomes less effective as the gap length increases. LSTM can save the data for a very long time. It is utilized for time-series data processing, forecasting, and classification [6]. The RNN known as the Long Short-Term Memory (LSTM) network has been prominent in time series and sequential analysis. A hidden layer, an output layer, and an input layer make up each of them. A LSTM network's hidden layer is made up of memory cells and three gates that are in charge of updating the cell state. An input gate, an output gate, and a forget gate make up these three gates [4]. The vanishing gradient problem, which is a significant factor in stock price prediction as prior data being processed in the neural network, does not affect LSTM networks in the same way that it does RNNs. Recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) are especially made to handle sequential data, including time series, speech, and text [3]. LSTM networks are particularly suited for applications like language translation, speech recognition, and time series forecasting because they can learn long-term dependencies in sequential data [2]. By storing states in the cell state, LSTM gets around this restriction. Additionally, the LSTM has a forget gate that determines whether or not previous state information is pertinent. Cell state stores the information if the Forget gate output is 1; otherwise, it ignores the information if the output is 0. Additionally, used in LSTMs are input and output gates [5].

3. Data Used & Proposed Methodology

First, we gather historical stock data for the chosen stocks from the Yahoo Finance API, including closing price, sales volume, and daily return. We pre-process the data by removing outliers and missing values from the data and normalizing it to the same scale. Pre-processing is done through MinMaxScaler which transforms the features by scaling them in the range between 0 and 1. The fit_tranformed (dataset) method is used to fit to the data and transform it. It returns the transformed array. By performing feature engineering, we produce new features from the existing data, such as sentiment analysis of stock-related news, technical indicators, and moving averages. A Model development is done to forecast stock closing prices, sales volume, and daily returns, create an LSTM model using Keras or TensorFlow. We do a comparative analysis to determine which stock has the largest expected return by comparing the predicted closing prices, sales volume, and daily return of the selected stocks. The closing price history shows the trend and performance of the stocks, plot the historical closing prices of the chosen stocks alongside the projected closing prices using the LSTM model. A trained model for price prediction is derived from the LSTM Model for later use, and periodically update the model with fresh data to boost prediction precision.

Below are the main findings of the paper:

Closing Price: The final price at which a specific stock is traded on a given day is known as the closing price of a stock. It shows the final price at which the stock was purchased or sold on that particular day during regular trading hours on the stock exchange.

Sales Volume: The total number of shares of a stock that were exchanged over a given time period, usually a day or a week, is referred to as the sales volume of that stock.

Adj. Closure: The term "Adj. Close" designates the final price at which a stock was traded on a specific trading day, adjusted for any corporate activities that may have taken place.

Moving Average: It is a computation that is used to reduce price volatility in a stock over a given time period. The average stock price over a predetermined number of time periods, such as days or weeks, is used to construct the moving average. When employed as a technical indicator, the moving average is frequently used to spot price trends in stocks

Daily Return: The percentage change in a stock's price from one day to the next is known as the daily return. It is computed as the difference between the stock's closing price on the current day and its closing price on the day before, divided by the stock's closing price on the day before.

Correlation: The degree to which the prices of two or more stocks move in relation to one another is referred to as correlation in the context of the stock market. A positive correlation means that the stock prices are moving together, whereas a negative correlation means that they are moving in the opposite direction. A correlation coefficient, whose values range from -1 to +1, is used to assess how closely two stocks are related.

Risk Quantification: A stock's risk quantification procedure involves calculating the likelihood and potential effects of the stock's volatility on investment returns.

First, we have a look at the descriptive statistics of each of the chosen stocks in a tabular form.

Here we use two functions namely describe () and info () to get the following output.

	Open	High	Low	Close	Adj Close	Volume
count	250.000000	250.000000	250.000000	250.000000	250.000000	2.500000e+02
mean	217.384293	222.589613	211.500453	217.015854	217.015854	1.144078e+08
std	52.077499	52.831450	51.090017	51.895759	51.895759	4.897508e+07
min	103.000000	111.750000	101.809998	108.099998	108.099998	4.186470e+07
25%	183.949997	186.382504	179.185001	183.177498	183.177498	7.847858e+07
50%	211.070000	221.084999	206.538338	214.340004	214.340004	9.778950e+07
75%	253.900833	257.456665	242.854164	252.255829	252.255829	1.447784e+08
max	313.006653	318.500000	305.579987	317.540009	317.540009	3.065906e+08

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 250 entries, 2022-05-02 to 2023-04-28
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Open          250 non-null    float64
 1   High          250 non-null    float64
 2   Low           250 non-null    float64
 3   Close         250 non-null    float64
 4   Adj Close     250 non-null    float64
 5   Volume        250 non-null    int64
 6   company_name  250 non-null    object
dtypes: float64(5), int64(1), object(1)
memory usage: 15.6+ KB
```

Figure 1. Descriptive statistics of TESLA using describe () and info ()

	Open	High	Low	Close	Adj Close	Volume
count	250.000000	250.000000	250.000000	250.000000	250.000000	2.500000e+02
mean	160.641480	163.887480	158.135400	161.010480	161.010480	3.368817e+07
std	33.798300	34.243701	33.287430	33.843637	33.843637	2.177520e+07
min	90.080002	90.459999	88.089996	88.910004	88.910004	1.200760e+07
25%	132.899998	136.827503	131.489994	133.284996	133.284996	2.306385e+07
50%	164.260002	167.625000	161.000000	163.610001	163.610001	2.800475e+07
75%	186.295006	190.797501	182.912498	185.180000	185.180000	3.660548e+07
max	239.889999	241.690002	236.770004	240.320007	240.320007	2.323166e+08

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 250 entries, 2022-05-02 to 2023-04-28
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Open              250 non-null    float64
 1   High              250 non-null    float64
 2   Low               250 non-null    float64
 3   Close             250 non-null    float64
 4   Adj Close         250 non-null    float64
 5   Volume            250 non-null    int64
 6   company_name      250 non-null    object
 7   MA for 10 days    241 non-null    float64
 8   MA for 20 days    231 non-null    float64
 9   MA for 50 days    201 non-null    float64
 10  Daily Return      249 non-null    float64
dtypes: float64(9), int64(1), object(1)
memory usage: 23.4+ KB
```

Figure 2. Descriptive statistics of META using describe () and info ()

	Open	High	Low	Close	Adj Close	Volume
count	250.000000	250.000000	250.000000	250.000000	250.000000	2.500000e+02
mean	28.137860	28.766100	27.532404	28.162000	28.162000	2.719001e+07
std	3.819757	3.831437	3.809756	3.835610	3.835610	1.510520e+07
min	20.370001	21.125999	19.895000	20.459999	20.459999	5.200400e+06
25%	24.960000	25.712501	24.282501	24.920000	24.920000	1.799392e+07
50%	28.405000	29.162499	27.920000	28.645000	28.645000	2.403790e+07
75%	31.022501	31.701249	30.492500	31.165000	31.165000	3.083808e+07
max	37.430000	37.580002	35.650002	36.830002	36.830002	1.156018e+08

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 250 entries, 2022-05-02 to 2023-04-28
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Open              250 non-null    float64
 1   High              250 non-null    float64
 2   Low               250 non-null    float64
 3   Close             250 non-null    float64
 4   Adj Close         250 non-null    float64
 5   Volume            250 non-null    int64
 6   company_name      250 non-null    object
 7   MA for 10 days    241 non-null    float64
 8   MA for 20 days    231 non-null    float64
 9   MA for 50 days    201 non-null    float64
 10  Daily Return      249 non-null    float64
dtypes: float64(9), int64(1), object(1)
memory usage: 23.4+ KB
```

Figure 3. Descriptive statistics of UBER using describe () and info ()

	Open	High	Low	Close	Adj Close	Volume
count	250.000000	250.000000	250.000000	250.000000	250.000000	2.500000e+02
mean	180.261000	184.428080	176.682000	180.969400	180.899985	5.231800e+07
std	43.894524	44.238161	43.631431	44.156148	44.175998	1.371971e+07
min	109.709999	117.349998	108.129997	112.269997	112.222221	1.679340e+07
25%	151.177502	154.414997	148.072498	151.537502	151.428009	4.316798e+07
50%	169.680000	174.080002	166.305000	169.834999	169.707306	5.006610e+07
75%	198.254997	203.415001	192.459999	202.009998	201.859192	5.894245e+07
max	279.660004	281.100006	273.570007	279.649994	279.649994	1.178865e+08

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 250 entries, 2022-05-02 to 2023-04-28
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Open              250 non-null    float64
 1   High              250 non-null    float64
 2   Low               250 non-null    float64
 3   Close             250 non-null    float64
 4   Adj Close         250 non-null    float64
 5   Volume            250 non-null    int64
 6   company_name      250 non-null    object
 7   MA for 10 days    241 non-null    float64
 8   MA for 20 days    231 non-null    float64
 9   MA for 50 days    201 non-null    float64
 10  Daily Return      249 non-null    float64
dtypes: float64(9), int64(1), object(1)
memory usage: 23.4+ KB
```

Figure 4. Descriptive statistics of NVIDIA using describe () and info ()

4. Results

Using the LSTM trained model prediction and python pandas and matplotlib libraries we obtain the following:

Figure 5. Closing price of the stocks

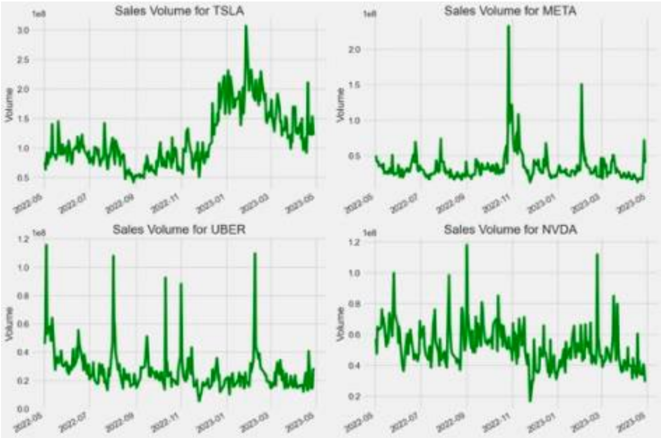

Figure 6. Sales Volume of the stocks

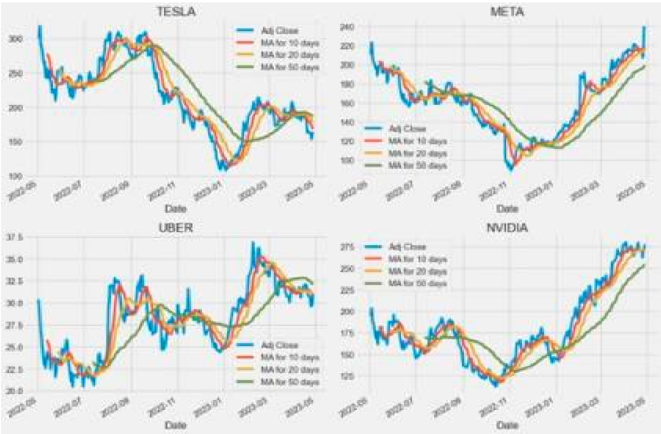

Figure 7. Adj. Closure and Moving Avg

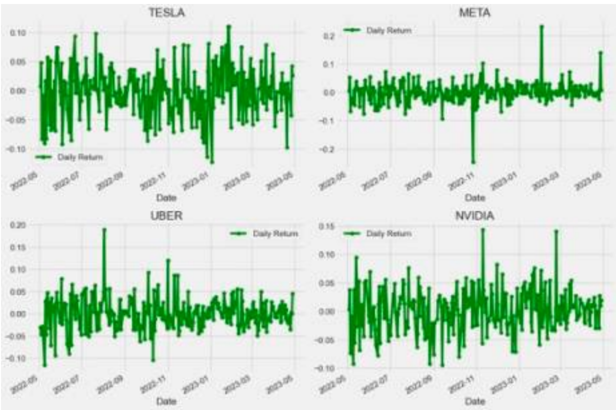

Figure 8. Daily Return of the stocks

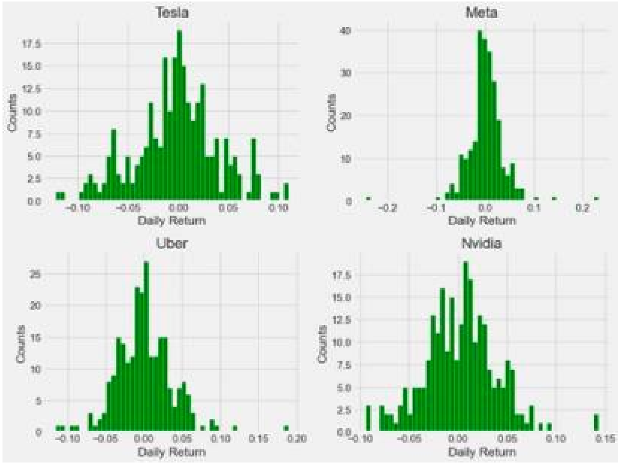

Figure 9. Daily return histogram

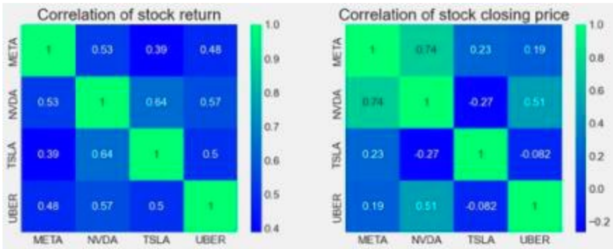

Figure 10. Correlation of Stocks

Figure 11. Close Price History

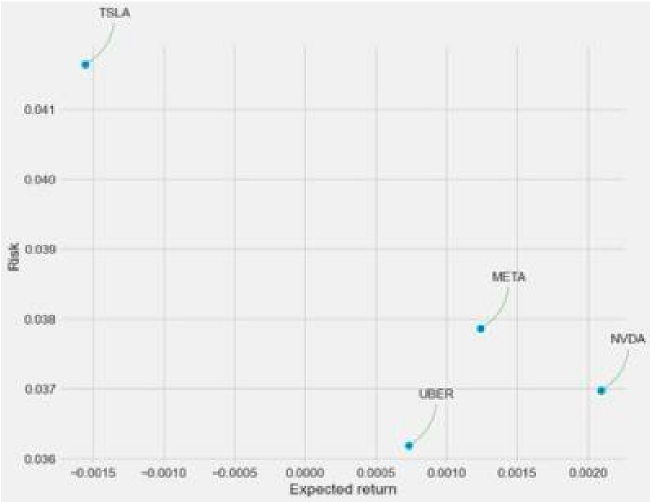

Figure 12. Risk Quantification

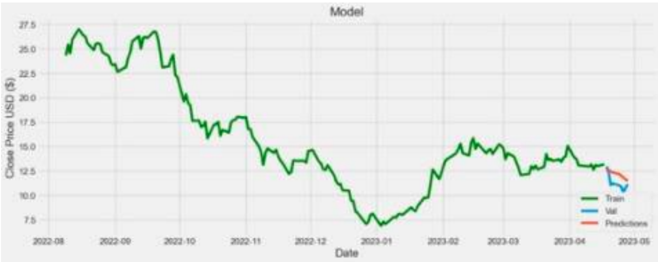

Figure 13. Future Prediction Model

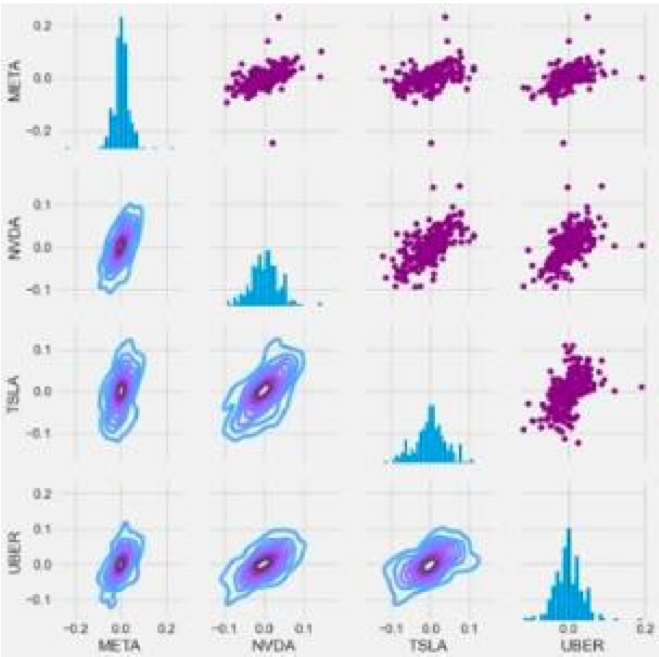

Figure 14. Correlation and kernel Density Plot

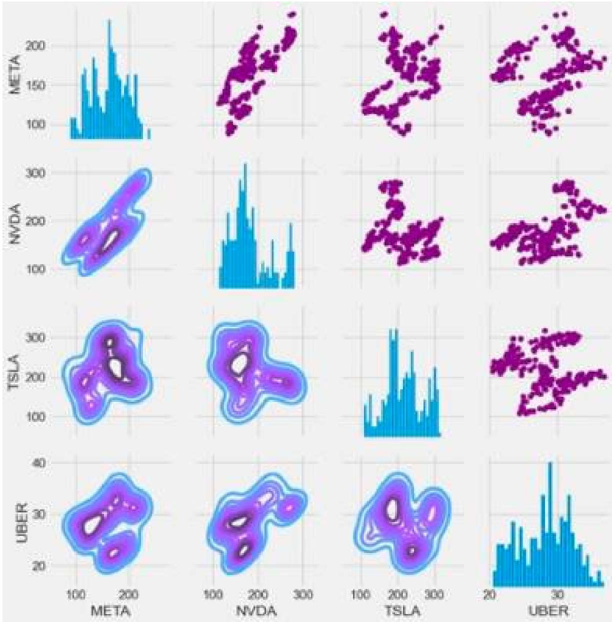

Figure 15. KDE & Histogram Plot via Daily Return

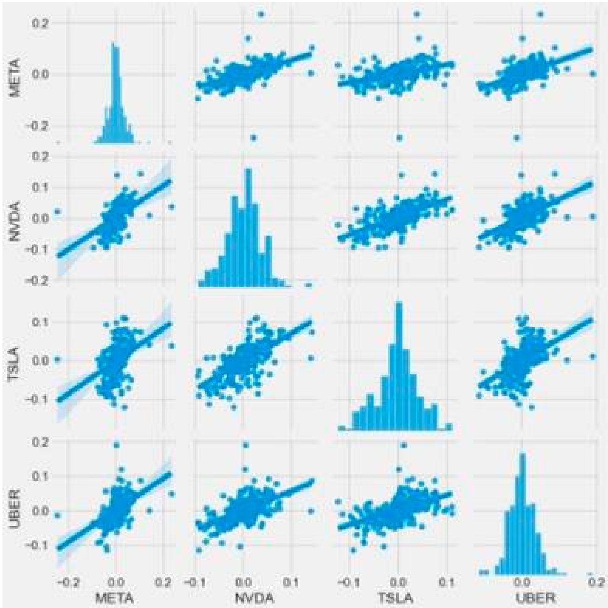

Figure 16. Visual Analysis of all Comparisons

5. Conclusions

In order to forecast stock market prices, we used LSTM, Yahoo Finance, Python Pandas, and Matplotlib modules. In order to predict future values for the stock's daily return, sales volume, moving average, and closing price, we trained an LSTM model using historical data. The results of the correlation i.e Fig 4.6 which reveals the correlation values between -1 to 1, the research revealed a substantial association between the daily return as well as the closing price of the four companies. Overall, the use of LSTM, Yahoo Finance, Python Pandas, and Matplotlib modules to anticipate stock market prices and give investors useful information was found to be a successful strategy.

References

[1] S. Mehtab, J. Sen and S. Dasgupta, "Robust Analysis of Stock Price Time Series Using CNN and LSTM-Based Deep Learning Models," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1481-1486, doi: 10.1109/ICECA49313.2020.9297652.

[2] P. S. Sisodia, A. Gupta, Y. Kumar and G. K. Ameta, "Stock Market Analysis and Prediction for Nifty50 using LSTM Deep Learning Approach," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar,India,2022,pp.156-161,doi:10.1109/ ICIPTM54933.2022.9754148.

[3] J. Eapen, D. Bein and A. Verma, "Novel Deep Learning Model with CNN and BiDirectional LSTM for Improved Stock Market Index Prediction," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2019, pp. 0264-0270, doi: 10.1109/CCWC.2019.8666592.

[4] Adil Moghar, Mhamed Hamiche, “Stock Market Prediction Using LSTM Recurrent Neural Network”, Procedia Computer Science, Volume 170, 2020, Pages 1168-1173,ISSN:1877-0509, https://doi.org/10.1016/j.procs.2020.03.049.

[5] G. Bathla, "Stock Price prediction using LSTM and SVR," 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 2020, pp. 211-214, doi: 10.1109/PDGC50313.2020.9315800.

[6] Adil Moghar, Mhamed Hamiche, “Stock Market Prediction Using LSTM Recurrent Neural Network”, Procedia Computer Science, Volume 170, 2020, Pages 1168-1173,ISSN:1877-0509,