Graduate Journal of Interdisciplinary Research, Reports & Reviews
(Grad.J.InteR^3)
ark:/85846/JPR3.2025AR.03010062

ISSN(E): 2584-2919
Vol.03, No.01 (2025)
pp.62-67

Report

Forecasting Stock Market Prices through Long Short-Term Memory Method

Bhavisshya Amit Goyal

21K School (British), Indiranagar, Bangalore - 560038, Karnataka, India.

* Email: bhavishyaamitgoyaal@gmail.com (corresponding author)

(Received:01-04-2025; Accepted: 15-05-2025; Published Online:27-05-2025)

ABSTRACT:
Prediction of futuristic price of stocks (shares) of listed companies at exchange has always been a fascinating area of interest for all kind of market participants. Whether short term traders. long term investors, or risk managers, everyone interest lies in forecasting the market accurately and in time. This study explores the application of Long Short-Term Memory (LSTM) networks, a class of Recurrent Neural Networks (RNNs), for predicting stock prices of major technology companies, Apple (AAPL), Google (GOOG), Microsoft (MSFT), and Amazon (AMZN) using historical data from 2005 to 2024. This work of mine “Stock Prediction using LSTM (Long Short-Term Memory”) method is a sincere effort in same direction, and I hope it will immensely help all market participants and serve them with more accuracy in forecasting share prices in near future period.
. **Keywords:** Forecasting stock market, stock market, shares, stocks

1. Introduction

This article guides a stepwise walkthrough through a new approach to stock market price forecasting using the Long-Short-Term Memory (LSTM) method While LSTM capability of capturing long-term dependent time frame of data, this paper provides a basis for insight into its use in stock market forecasting. By analyzing the intricacies of the LSTM's architecture and adapting it to stock market data, this study provides a comprehensive assessment of its potential for accurate forecasting in financial markets. Through a systematic analysis of real-world cases, this paper sheds light on the potential of LSTM as a powerful tool for improving stock valuations.

This study aims to implement and analyze LSTM models for stock price forecasting, addressing the following research questions:

i How effective are LSTM models at predicting stock market prices, compared to Traditional Statistical Techniques?

ii What are the challenges and limitations of using LSTM models for financial forecasting?

iii Can traders and investors get useful insights from LSTM models?

2. Review of Literature

Several researches have studied the application of deep learning techniques to financial prediction. Hochreiter and Schmidhuber [1] introduced LSTM-based networks to overcome the vanishing gradient problem inherent in recurrent neural networks (RNNs). Chen et al. [2] demonstrated that LSTM models outperformed traditional methods in forecasting stock prices, highlighting their ability to capture complex temporal patterns. Similarly, a study by Li et al. [3] integrated Symbolic Genetic Programming with LSTM, achieving significant improvements in prediction accuracy for Chinese stocks. LSTM is now widely used in areas such as speech recognition, natural language processing, and financial time-series predictions. Recent investigations have examined LSTM applications to stock market predictions, showing some good results.

```
pip install yfinance
Show hidden output

import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
plt.style.use("fivethirtyeight")
%matplotlib inline

# For reading stock data from yahoo
from pandas_datareader.data import DataReader
import yfinance as yf
from pandas_datareader import data as pdr


# For time stamps
from datetime import datetime


# The tech stocks we'll use for this analysis
tech_list = ['AAPL', 'GOOG', 'MSFT', 'AMZN']

# Set up End and Start times for data grab
tech_list = ['AAPL', 'GOOG', 'MSFT', 'AMZN']

end = datetime.now()
start = datetime(end.year - 1, end.month, end.day)

for stock in tech_list:
    globals()[stock] = yf.download(stock, start, end)


company_list = [AAPL, GOOG, MSFT, AMZN]
company_name = ["APPLE", "GOOGLE", "MICROSOFT", "AMAZON"]

for company, com_name in zip(company_list, company_name):
    company["company_name"] = com_name

df = pd.concat(company_list, axis=0)
df.tail(10)
```

Figure 1: code snippet

Challenges such as overfitting, the challenges posed by noisy data, and the aberration of the market continue to provide barriers.

3. Data Collection and Preprocessing

For this study, Yahoo Finance [4] was used to provide the historical stock price data, including the daily stock prices of Apple (AAPL), Google (GOOG), Microsoft (MSFT), and Amazon (AMZN), covering the period from January 1, 2005, to December 31, 2024. The dataset included daily open, high, low, close, adjusted close prices, and trading volume. Features were normalized using Min-Max scaling to ensure that all inputs to the LSTM model are on a comparable scale, facilitating faster convergence during training.
To enhance the model's predictive capabilities, additional features were engineered.

i **Closing Price:** Served as the key indicator of market sentiment and the primary variable for time-series prediction.

ii **Sales Volume:** Representing the total number of shares traded daily and was included to capture market activity and liquidity trends that often precede price movements.

iii **Daily Returns:** Calculated as the percentage change in closing prices.

iv **Moving Averages:** Computed 10-day, 20-day, and 50-day moving averages (MA) for each stock Apple, Google, Microsoft, and Amazon based on their respective closing price tickers, to capture short-, medium-, and long-term trend patterns respectively.

The code snippet demonstrates data extraction and preprocessing in figure 1.

```
AAPL.describe()
```

Price	Adj Close	Close	High	Low	Open	Volume
Ticker	AAPL	AAPL	AAPL	AAPL	AAPL	AAPL
count	252.000000	252.000000	252.000000	252.000000	252.000000	2.520000e+02
mean	199.457422	199.816349	201.458691	197.973611	199.685198	5.798851e+07
std	21.515374	21.326477	21.555429	20.956342	21.341770	3.027396e+07
min	164.585999	165.000000	166.399994	164.080002	165.350006	2.404830e+07
25%	182.899498	183.297504	184.887497	181.649998	183.305000	4.258440e+07
50%	192.569260	193.164993	194.690002	192.025002	193.360001	5.119855e+07
75%	221.869930	222.027496	224.149994	219.782505	221.652496	6.472080e+07
max	236.479996	236.479996	237.490005	234.449997	236.479996	3.186799e+08

```
GOOG.describe()
```

Price	Close	High	Low	Open	Volume
Ticker	GOOG	GOOG	GOOG	GOOG	GOOG
count	251.000000	251.000000	251.000000	251.000000	2.510000e+02
mean	172.559578	174.369473	170.833511	172.536373	1.909588e+07
std	15.212333	15.382551	15.052332	15.177810	8.176270e+06
min	132.085403	133.540187	131.079029	132.264771	6.809800e+06
25%	163.501740	165.105103	162.461693	163.941066	1.406835e+07
50%	172.028122	173.422883	169.809865	171.409650	1.685310e+07
75%	183.131569	185.192793	181.868985	183.774686	2.090825e+07
max	207.710007	208.699997	204.259995	204.500000	5.972800e+07

Figure 2: data extraction

```
MSFT.describe()
```

Price	Close	High	Low	Open	Volume
Ticker	MSFT	MSFT	MSFT	MSFT	MSFT
count	251.000000	251.000000	251.000000	251.000000	2.510000e+02
mean	421.336973	424.820298	417.658032	421.533867	2.043706e+07
std	15.443741	15.049774	15.274857	14.974740	7.538829e+06
min	384.649994	387.529999	381.000000	383.350006	7.164500e+06
25%	411.605911	414.643590	407.916915	411.521543	1.616870e+07
50%	418.703339	422.848822	415.533391	419.249358	1.871850e+07
75%	429.338013	431.708499	424.990802	429.567648	2.241160e+07
max	464.854340	465.639777	461.772294	464.297590	6.426370e+07

```
AMZN.describe()
```

Price	Close	High	Low	Open	Volume
Ticker	AMZN	AMZN	AMZN	AMZN	AMZN
count	251.000000	251.000000	251.000000	251.000000	2.510000e+02
mean	195.063466	197.118845	192.866303	195.189960	3.909345e+07
std	19.784831	19.879338	19.561573	19.721167	1.566125e+07
min	161.020004	162.960007	151.610001	154.210007	1.500750e+07
25%	180.775002	182.995003	179.320000	181.114998	2.982875e+07
50%	187.000000	188.940002	185.419998	187.429993	3.569020e+07
75%	208.825005	212.434998	206.500000	209.024994	4.234970e+07
max	242.059998	242.520004	238.029999	239.020004	1.414484e+08

Figure 3: data extraction

4. LSTM Model Architecture

The LSTM model implemented in this study consists of multiple layers, including LSTM units, dropout layers

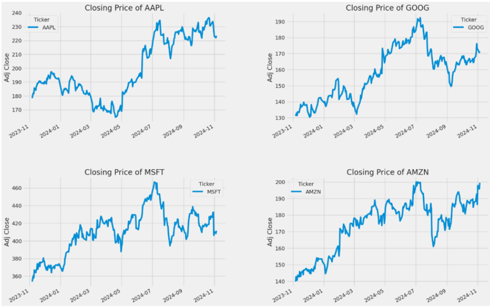

Figure 4: Closing Prices

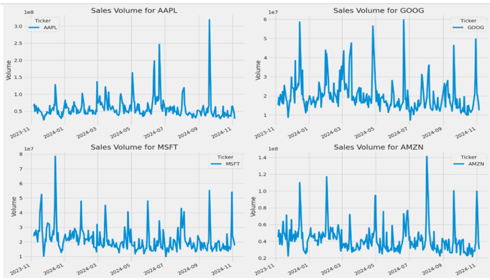

Figure 5: Sales Volume

to mitigate overfitting, and dense layers for output generation. The model architecture is outlined below:

5. Results

The Long Short-Term Memory (LSTM) model was employed to predict the closing price of Apple Inc. (AAPL) stock using historical data. The results, as depicted in the graph number 12, it shows the model's performance in three distinct phases.

5.1. Training Phase (Blue Line)

i The LSTM model was trained on the historical price data of the stock, covering a long-term trend of Apple's stock performance in the market.

ii The training data is from 2005 to 2024, so, it captures the price fluctuations, long-term trends, and volatility.

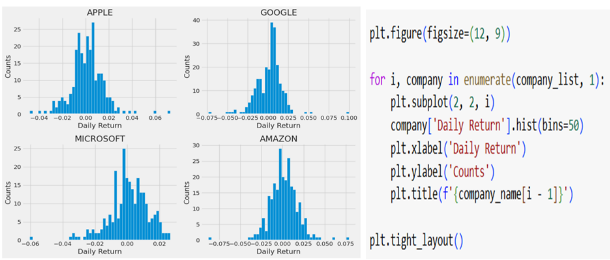

Figure 6: Daily Return

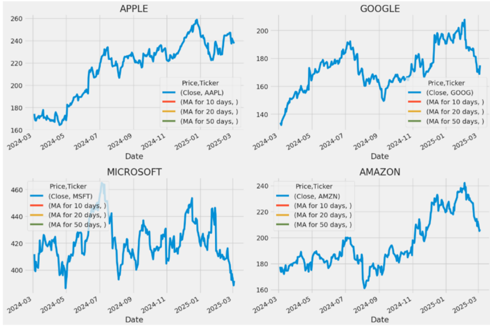

Figure 7: Moving Average-1

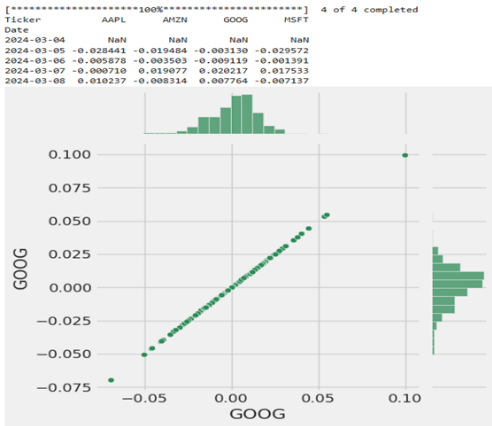

Figure 8: Moving Average-2

5.2. Validation Phase (Red Line):

i A subset of the dataset was kept apart from the rest at the time of training to be used to evaluate the model's learning process

ii The validation data (approximately from early 2023 to late 2024) demonstrates a strong correlation with the historical trend, confirming the model's reliability.

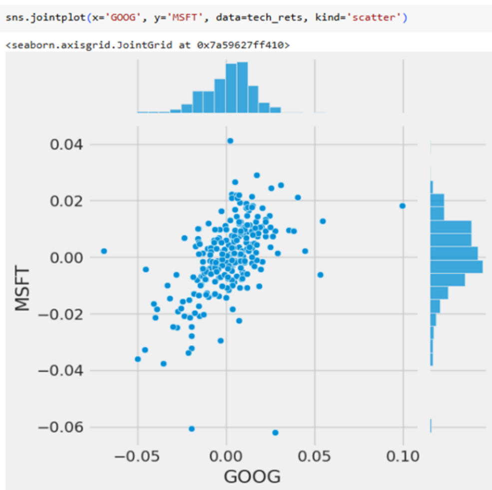

Figure 9: Moving Average-3

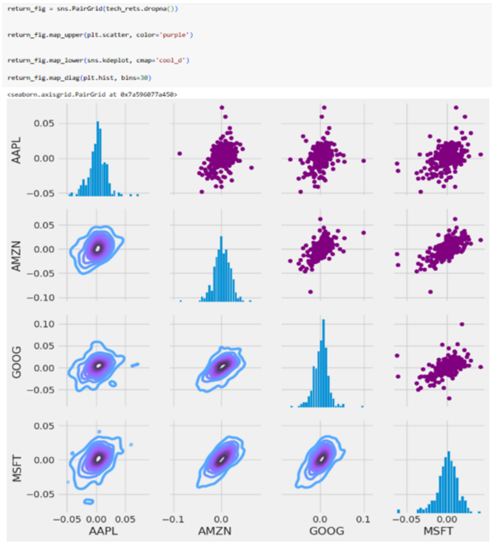

Figure 10: Moving Average-4

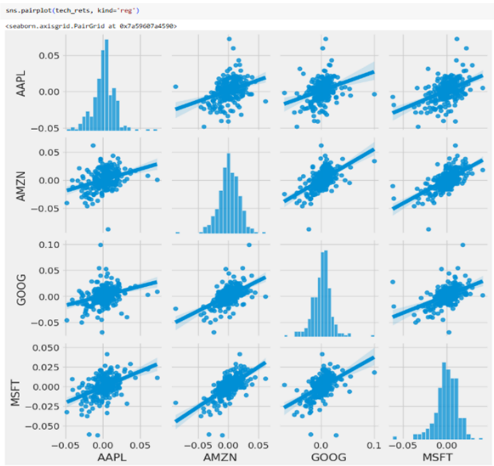

Figure 11: Moving Average-5

5.3. *Prediction Phase (Yellow Line):*

i With the help of the LSTM, the model made predictions for future stock prices beyond the time period of validation.

ii The predicted values closely follow the validation set, suggesting that the LSTM model effectively captures patterns and trends in Apple's stock price.

```
stocks = ['AAPL', 'MSFT', 'GOOG', 'AMZN']
data = yf.download(stocks, start="2023-09-01", end="2024-10-01")
closing_df = data['Close']

[*********************100%***********************]  4 of 4 completed

plt.figure(figsize=(12, 10))

plt.subplot(2, 2, 1)
sns.heatmap(tech_rets.corr(), annot=True, cmap='summer')
plt.title('Correlation of stock return')

plt.subplot(2, 2, 2)
sns.heatmap(closing_df.corr(), annot=True, cmap='summer')
plt.title('Correlation of stock closing price')
```

Figure 12: Correlation code

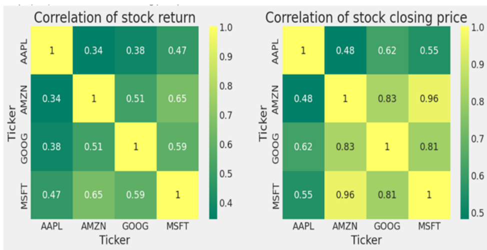

Figure 13: Correlation in stock return

```
from keras.models import Sequential
from keras.layers import Dense, LSTM

# Build the LSTM model
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape= (x_train.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
model.fit(x_train, y_train, batch_size=1, epochs=10)
```

Figure 14: LSTM Model

6. Challenges

While LSTMs outperform traditional methods, several challenges remain:

i **Market Volatility:** Sudden economic shifts or geopolitical events can cause stock price fluctuations.

ii **Data Quality:** The presence of noise in financial data can affect prediction accuracy.

iii **Computational Complexity:** Training deep LSTM models is resource-intensive and requires optimization techniques.

Although these are the hurdles associated with LSTMs, however, they provide trade insights by quantifying

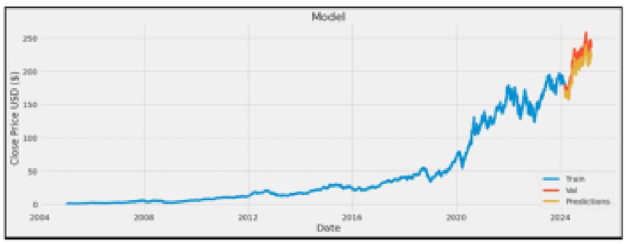

Figure 15: The Long Short-Term Memory

patterns and providing prediction errors. Future improvements may include the implementation of sentiment analysis, hybridization of the models between LSTMs and reinforcement learning, and experimentation with attention mechanisms for increased prediction accuracy.

7. Conclusion

This study highlights the effectiveness of LSTM networks in stock market prediction. Future research directions include:

i Integrating real-time news and sentiment analysis for improved accuracy.

ii Exploring hybrid architectures such as convolutional neural networks (CNNs) combined with LSTMs.

iii Conducting hyperparameter tuning and testing alternative optimizers for model refinement. These enhancements will further solidify LSTM-based stock predictions, making them more reliable for investment and trading strategies.

Authorship contribution: Author have made significant contributions to the research presented in this report.
Funding: No funding by any agency.
Conflict of interest: Author have no conflict of interest with anyone.
Declaration: It is an original data and has not been sent or published anywhere.
Similarity Index: The authors hereby confirm that there is no similarity index in abstract and conclusion while overall is less than 10% where individual source contribution is 2% or less than it.

References

[1] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[2] Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock price prediction. *Expert Systems with Applications*, *42*(8), 3903–3910. https://doi.org/10.1016/j.eswa.2014.11.047

[3] Li Qi, et al. (2024). Integrating Symbolic Genetic Programming With Lstm for Forecasting Cross-Sectional Price Returns: A Comparative Analysis of Chinese And Japanese Stock Market. *Advances in Artificial Intelligence and Machine Learning*, *4*(4), 2981-3005.

[4] Yahoo Finance. (n.d.). Historical market data. Retrieved from https://www.yfinance.com

Copyright and License

How to Cite?

Bhavisshya Amit Goyal (2025). Forecasting Stock Market Prices through Long Short-Term Memory Method. *Graduate Journal of Interdisciplinary Research, Reports and Reviews*, 3(01), 62-67. Retrieved from https://jpr.vyomhansjournals.com/index.php/gjir/article/view/54