International Journal of Development and Economic Sustainability
Vol.12, No.3, pp.36-45, 2024
Print ISSN: 2053-2199 (Print),
Online ISSN: 2053-2202(Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

Application of Long Short-Term Memory (LSTM) in Stock Price Prediction

David Olumuyiwa Sunday

doi: https://doi.org/10.37745/ijdes.13/vol12n33645 Published June 19, 2024

Citation: Sunday D.O. (2024) Application of Long Short-Term Memory (LSTM) in Stock Price Prediction, *International Journal of Development and Economic Sustainability*, Vol.12, No.3, pp.36-45

ABSTRACT: *No doubt, the stock market has been one of the most volatile and this makes it very difficult to carry out forecast or prediction about the movement in share prices. However, the advent of machine learning is gradually changing the narratives as there are a number of neural networks which could be used to develop models to effectively predict the stock prices. This study examined the application of Long Short-Term Memory algorithms in predicting stock prices using time series data on Apple shares from 2017 to 2022. The closing price of the stocks was adopted and the data split into 98% for training and 2% for test because the study only wanted to make prediction for the last 20 days. TensorFlow together with its subset Keras libraries were employed with other libraries like Dropout for regularization and sklearn for normalization. NumPy to build array, pandas for data analysis were also imported with matplotlib for the purpose of plotting the result graph. The LSTM model was trained to learn from the sequential data and predict the share price. The outcome revealed that the machine learnt very effectively and has a predictive ability for share prices. The model was able to predict the trend or pattern in prices almost accurately.*
KEYWORDS: machine learning, neural network, recurring neural network, long short-term memory

INTRODUCTION

Over the years, it has become increasingly difficult, if not impossible, to estimate or predict the prices of stocks because of lack of future information about stock market which is highly volatile and very unpredictable. People constantly seek for more accurate and efficient ways of stock trading knowing fully well that it is not enough to depend primarily on experience and personal judgement to make stocks price prediction (Bin, Ahmed & Megahed 2017; Veeresh 2021). This unpredictability of the stock market has led to a number of researches in this area without any consensus by the scholars (Borovkova & Tsiamas 2019).

However, these challenges in making predictions about market activities is steadily reducing with the evolution of machine learning algorithms which could be trained to make forecast and predictions in stock prices. Qiu, Wang and Zhou (2020) mentioned that industry and the academic world have focused on financial market forecasting using machine learning. Scholars are now continually examining the role of machine learning in the stock market and many hedge funds and stock broking firms employ people with knowledge of machine algorithms (Zhang 2022). The application of machine learning in predicting stock prices has become the central point of many studies due to the rapid evolvement of artificial intelligence (Veeresh 2021).

Today many organisations have deployed computer systems to optimise their processes with machine learning as one of the transformational forces, especially in finance (Tanya 2021). With this invention, machine learning has played vital role in the financial sector in areas such as loan and insurance underwriting, credit scores, risk assessment, portfolio management, Algorithm trading and fraud detection (Emerson et al., 2019; Fagella 2020). Issam, Ruijiang and Martin (2015) explained machine learning as a growing aspect of computational algorithms that are created to mimic human intelligence by learning from the immediate environment. Also, Lopez (2016) opined that machine learning techniques consider non-linear relationships in data, and this makes it effective in applying machine learning to financial problems.

Neural network in deep learning has become very prevalent in the application of machine learning algorithms as a result of its capacity to adapt with self-learning and linear approximations (Qiu, Wang & Zhou 2020). According to Masood and Abbas (2024), neural networks has transformed machine learning by aiding computers in learning from data, identifying patterns and making smart decisions in the same way human brain does. Haykin (2009) described Neural Network (NN) as a form of deep learning that mimic the functions of the human brain with the use of some interconnected nodes or neurons which are the basic building blocks of an artificial intelligence system and deep learning framework.

Micheal (2019) classified Neural Network into feedforward or forward propagation (Artificial Neural Network and Convolutional Neural Network) and backpropagation (Recurrent Neural Network). In feedforward propagation, the data flows only in a forward direction from an input layer to the hidden layers and then to the output. The Recurrent Neural Network relates to backward propagation of errors and has the propensity to mimic the human brain's operations better than the feedforward networks. It is an enhancement to the other neural network because it has capacity to provide solutions to the issues associated with feedforward networks like the inability to handle sequential data, consideration for only current input, inability to memorise previous input, failure to consider experience in learning.

Recurrent Neural Networks have been effectively applied in some sequential learning cases like language modelling, speech and voice recognition, and terminal predictions (Anil & Venkatesh 2020). It has the capacity to process sequential data because it connects previous information to the present which is useful in predicting what the next information will be (Tolo 2019). Also, the backpropagation feature makes it very useful in processing sequential data like speech and voice recognition, natural language translation, and financial time series modelling. Recurrent Neural Network has the capacity for long-term and dynamic dependencies in an input sequence. Vengertsev and Sherman (2020) explained that RNN is based on longer trained length of sequences unlike forward propagation.

Recurrent Neural Network is an emerging and promising technology in finance and has effective application in forecasting, especially in time series because they have the capacity to store information on both past and current data. It can be used in predicting share prices, inflation, corporate bankruptcy and corporate finance, security markets, forecasting financial distress.

The Long short-term memory (LSTM) is a variant of the Recurrent Neural Network (RNN) which has wide applications in Finance. It has a deep learning capacity and due to its distinctive features of backward propagation and internal memory, it is very useful in predicting financial time series (Xu, Zhang & Ma 2017), weather forecasting, speech and hand recognition (Dinesh et al., 2021). LSTM neural networks are very appropriate for stock price time series (Shih et al., 2017). The model makes use of historical data to analyse the volatility of a particular period and predict the trend of stock prices and this is made possible by its sequence modelling and long term memory capability (Zhang 2022)

Dinesh et al., (2021) stated that LSTM adopts the Recurrent Neural Network methodology with the capability to memorise and each cell consists of input, forget and output gates, with one tanh layer in each memory cell. When data goes into the LSTM network, only the needed data is retained while all unwanted data are neglected by the forget gate. The input gate is used to protect the memory content from any agitation by unwanted inputs (Hochreiter & Schmidhuber 1997). Again, Veeresh (2021) explained that forget gate produces numbers from 0 to 1 with 0 indicating a total hold and 1 showing complete neglect. The output gate determines the output in each cell based on the cell condition.

LSTM models are very effective in predicting sequence problems due to the ability to store past information or data (Danesh et al., 2022). This makes LSTM models very important in stock price prediction because previous data relating to stocks need to be stored and studied in order to be able to carry out forecast of the future prices correctly.

This paper focuses on the training of LSTM models, a type of Recurrent Neural Network, in the prediction of Apple stock price using historical data from 2017 – 2022 and how this could improve stock trading effectiveness.

Empirical Review

Related studies previously conducted by some scholars were reviewed with the objective of evaluating the application of LSTM models in time series data predictions.
Ta, Liu and Tadesse (2020) used Long Short-term Memory (LSTM) neural network to predict stock movement based on historical data and adopted equal-weighted method, Monte Carlo simulation and Mean Variant (MV) model to improve the portfolio performance. The outcome revealed that the proposed LSTM prediction model perform more efficiently with high accuracy for stock prediction than the constructed portfolios-based prediction models such as linear regression.

Zhang (2022) in his work, trained LSTM and SVM models to predict stock prices changes using S&P 500 stocks for the period between 2013 and 2018. The study showed that LSTM model produced a better prediction. Also, Fischer and Krauss (2017) in their various studies, applied the LSTM, Random Forests, deep networks and logistic regression models for stock trading prediction using daily S&P500 data from 1992 to 2015. The outcome revealed that LSTM model had the best predictions.

Nelson, Pereira, Oliveira (2017) collected data on some Sao Paolo exchange stocks from 2008 to 2015 and applied LSTM model based on 15-minute-interval observations for price forecast. The results showed the model achieved about 55.9% accuracy on the stock price prediction.

Hardinata, Warsito and Suparti (2018) applied Jordan Recurrent Neural Networks to classify and predict corporate bankruptcy based on financial ratios and the results showed that the model worked well for bankruptcy prediction with average success rate of 81%. Again, Tolo (2019) employed Recurrent Neural Network in predicting systemic financial crises between one to five years ahead. The author used Jorda-Schularick-Taylor dataset, which includes the crisis dates and relevant macroeconomic series of 17 countries over the period 1870-2016. The outcome of the study found Recurrent Neural Network architecture to be useful in predicting systemic financial crises.Paranhos (2023) used LSTM to forecast inflation using United States' time series data and the results showed that LSTM performed well in forecasting the periods of uncertainty.

METHODOLOGY

The study employed a total of 5-year sequential data on Apple (AAPL) stock price downloaded from yahoo finance covering the period 2017 to 2022. The closing price of the stocks was adopted and the data is not randomised but instead split into 98% for training and 2% for test because the study only wants to make prediction for the last 20 days.

First, TensorFlow together with its subset Keras libraries, which have been developed primarily for this type of model was installed. Thereafter, libraries like Dropout for regularization and sklearn for normalization before data is sent to the keras were imported. Other libraries like math for mathematical operations, numpy to build array, pandas for data analysis were also imported with matplotlib for the purpose of plotting the result graph.

```
## having installed the Tensorflow, we then load other libraries as shown below
```

```
import math
import matplotlib.pyplot as plt
import keras
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.layers import *
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from keras.callbacks import EarlyStopping
```

Next, imported both the train and test data.

```
## we then load the Apple data. the data has been splitted into tarin and test
## and they are not randomised
```

```
train= pd.read_csv('Price train.csv')
test= pd.read_csv('Price test.csv')
```

Changed the array of the data from two-dimensional array to one dimensional as below:

```
train_open= train.iloc[:, 1:2].values
```

```
train_open
```

```
array([[ 43.587502],
       [ 43.637501],
       [ 43.290001],
       ...,
       [148.210007],
       [145.960007],
       [147.770004]])
```

rescaled the data from 0 to 1

```
#Scaling the values between 0 to 1
from sklearn.preprocessing import MinMaxScaler
ss= MinMaxScaler(feature_range=(0,1))
train_open_scaled= ss.fit_transform(train_open)
```

```
train_open_scaled[60]
```

```
array([0.04876053])
```

The previous 60 days open data steps was employed to forecast the next values for 61st and 62nd day based on the opening and closing price of the share and the knowledge gained. The data was converted based on the input in LSTM layers. The input features were built with time lag of 1 day.

```
# Feature selection
xtrain=[]
ytrain=[]
for i in range(60,len(train_open_scaled)):
 xtrain.append(train_open_scaled[i-60:i,0])
 ytrain.append(train_open_scaled[i,0])
xtrain, ytrain = np.array(xtrain), np.array(ytrain)
#We will now reshape the data into the following format (#values, #time-steps, #1 dimensional output).
```

```
#Reshaping the train data to make it as input for LTSM layer input_shape(batchzise,timesteps,input_dim)
xtrain= np.reshape(xtrain,(xtrain.shape[0],xtrain.shape[1],1))
```

```
xtrain.shape
```

```
(1177, 60, 1)
```

Adam Optimizer was used to train the model and the Mean Square Error loss function adopted for measuring the mean square error loss. Then 0.2% dropout was added to the layers to avert the problems of overfitting in the model. Lastly, the model was trained by fitting the training set and setting the batch size and the number of loops it will run as 30 and 200 epochs. The error function moved from 0.0265 to 0.0012 which is the minimum it could get.

```
#initialising the model
model= Sequential()
#First Input layer and LSTM layer with 0.2% dropout
model.add(LSTM(units=50,return_sequences=True,kernel_initializer='glorot_uniform',input_shape=(xtrain.s
],1)))
model.add(Dropout(0.2))
# Where:
# return_sequences: Boolean. Whether to return the last output in the output sequence, or the full sequ
# Second LSTM layer with 0.2% dropout
model.add(LSTM(units=50,kernel_initializer='glorot_uniform',return_sequences=True))
model.add(Dropout(0.2))
#Third LSTM layer with 0.2% dropout
model.add(LSTM(units=50,kernel_initializer='glorot_uniform',return_sequences=True))
model.add(Dropout(0.2))
#Fourth LSTM layer with 0.2% dropout, we wont use return sequence true in last layers as we dont want t
model.add(LSTM(units=50,kernel_initializer='glorot_uniform'))
model.add(Dropout(0.2))
#Output layer , we wont pass any activation as its continous value model
model.add(Dense(units=1))
#Compiling the network
model.compile(optimizer='adam',loss='mean_squared_error')
#fitting the network
model.fit(xtrain,ytrain,batch_size=30,epochs=200)
```



```
Epoch 1/200
40/40 [==============================] - 10s 66ms/step - loss: 0.0431
Epoch 2/200
40/40 [==============================] - 3s 69ms/step - loss: 0.0056
Epoch 3/200
40/40 [==============================] - 3s 64ms/step - loss: 0.0061
Epoch 4/200
40/40 [==============================] - 3s 65ms/step - loss: 0.0048
```

International Journal of Development and Economic Sustainability
Vol.12, No.3, pp.36-45, 2024
Print ISSN: 2053-2199 (Print),
Online ISSN: 2053-2202(Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

```
40/40 [==============================] - 3s 75ms/step - loss: 0.0012
Epoch 199/200
40/40 [==============================] - 3s 74ms/step - loss: 0.0010
Epoch 200/200
40/40 [==============================] - 3s 74ms/step - loss: 0.0012

51: <keras.callbacks.History at 0x1a28a29c520>
```

Concatenation of the train and test datasets was carried out and the last 60 values of the training set with the test set were taken.

```
test_open= test.iloc[:, 1:2].values #taking open price
total= pd.concat([train['Open'],test['Open']],axis=0) # Concating train and test and then will take las
test_input = total[len(total)-len(test)-60:].values
test_input= test_input.reshape(-1,1) # reshaping it to get it transformed
test_input= ss.transform(test_input)
```

the range of prediction which is 20 days in this case (since we are dealing with data that has daily interval) was identified while input for LSTM prediction was also created.

```
xtest= []
for i in range(60,80):
 xtest.append(test_input[i-60:i,0]) #creating input for lstm prediction
```

the model was then used to predict the new price as follows:

```
xtest= np.array(xtest)
```

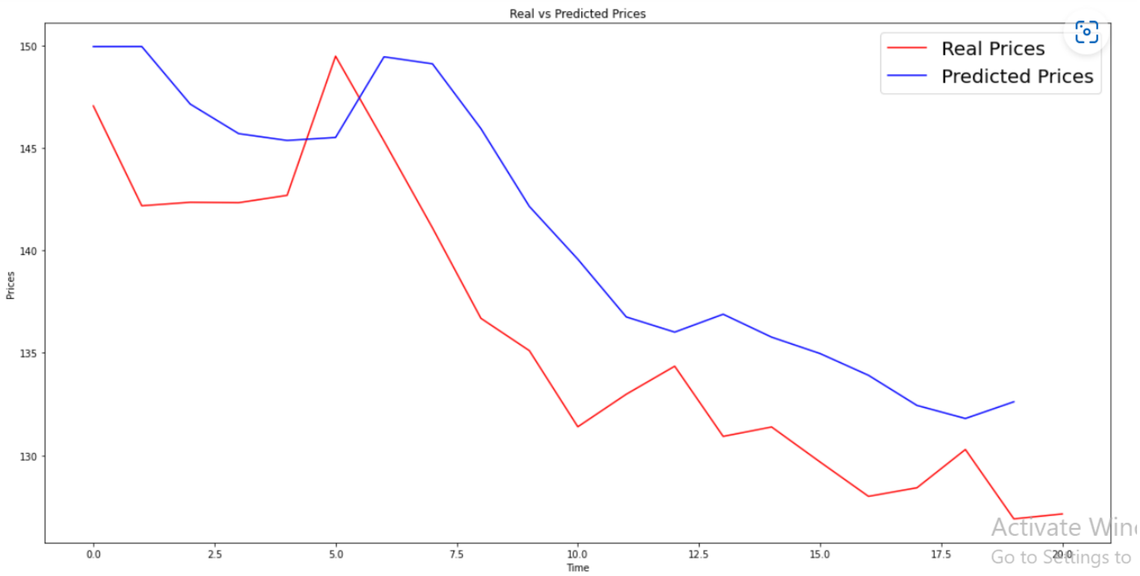
```
xtest= np.reshape(xtest,(xtest.shape[0],xtest.shape[1],1))
predicted_value= model.predict(xtest)
```

```
1/1 [==============================] - 2s 2s/step
```

The results of the prediction was then visualized as shown below:

```
predicted_value= ss.inverse_transform(predicted_value)
```

```
plt.figure(figsize=(20,10))
plt.plot(test_open,'red',label='Real Prices')
plt.plot(predicted_value,'blue',label='Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Prices')
plt.title('Real vs Predicted Prices')
plt.legend(loc='best', fontsize=20)
```



Interpretation of the model results

The graph above shows the results of the comparison of Apple's actual and predicted prices using LSTM model trained on historical Apple stock prices. The model was able to predict the trend or pattern in prices almost accurately but could not predict the exact price. This showed that the machine is learning and has a predictive ability for share prices. This outcome aligns with the findings of Dinesh et al., (2021) and Nabipour et al., (2020). It also conforms with the conclusion made in the work of Zhang (2022) that LSTM models produced better prediction.

CONCLUSION

This study examined how effective the LSTM algorithms could be adopted in predicting stock prices using time series data on Apple shares from 2017 to 2022. The LSTM model was trained to learn from the data and predict the share price. The outcome provided further affirmations to some of the existing studies that LSTM has a good predictive capacity, especially the trend of share prices.

The result also shows that emergence of deep learning algorithms like LSTM with great predictive capabilities has implications on the theory of Efficient Market Hypothesis (EMH) which states that future asset prices could not be predicted, and no individual could profit by making predictions of future prices since no such future information is available. This has led to volumes of research on this subject, though without any consensus yet (Borovkova & Tsiamas 2019; Zhang 2022)

Recommendation

Based on the findings and conclusions of this study, the following recommendations were proffered:

Organisations, especially brokerage firms, should invest in the use of machine learning as this has capacity to improve efficiency in stock predictions. The use of machine learning requires data for learning and so organisations should ensure data is available and accurately kept to facilitate effective learning outcome.
Also, governments, companies and individuals should deepen research into the use of machine learning to drive innovation and the level of competitiveness, even beyond stock price predictions to other areas in finance like portfolio management, risk management, fraud detection, loan appraisal, among others.

References

Anil, K. & Venkatesh, S. (2021). Forward propagating through time. Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 5189-5200.

Bin, W., Ahmed, M. A. & Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. Expert Systems with Applications, 79(2),153–163.

Borovkova, S. & Tsiamas, I (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. Journal of Forecasting, 38, 600–619

Dinesh, S., Rahul, S., Sandeep, O., Relangi, K. & Rama Raju, A. M. S. (2021). Stock price prediction using LSTM. Mukt Shabd Journal, Volume 10(6), 436-442

Emerson, S., Kennedy, R., O'Shea, L. & O'Brien, J. (2019). Trends and applications of machine learning in quantitative finance . 8th International Conference on Economics and Finance Research (ICEFR 2019).

Fagella, D. (2020). Machine learning in finance. Present and future applications. https://emerj.com/ai-sector-overviews/machine-learning-in-finance/.

Fischer, T. & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

Hardinata, L., Warsito, B. & Suparti, I. (2018). Bankruptcy prediction based on financial ratios using Jordan Recurrent Neural Networks. A case study in Polish companies. Journal of Physics, Series 1025.

Haykin, S. (2009). Neural networks and learning machines. Prentice Hall/Pearson, New York.

Hochreiter, S. & Schmidhuber, J. (1997). Long Short-term Memory. Neural computation, 9(8),1735-1780

Issam, E. N., Ruijiang, L. & Martin, J. M. (2015). Machine learning in radiation oncology. Theory and applications. Springer Nature. ISBN : 978-3-319-18304-6.

Lopez, M. D. (2016). Building diversified portfolios that outperform out-of-sample. Journal of Portfolio Management, 31.

Masood, S. & Abbas, A. (2024). Neural network and deep learning. A comprehensive overview of modern techniques and applications. Journal of Environmental Sciences and Technology, 2(2), 8-20.

Micheal, H. (2019). Neural networks and deep learning. Retrieved from https://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf on January 8, 2023

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E. & Shahab, S. (2020). Deep learning for stock market prediction. 10.20944/preprints202003.0256.v1.

Nelson, D.M., Pereira, A.M., & Oliveira, R.A. (2017). Stock market's price movement prediction with LSTM neural networks. International Joint Conference on Neural Networks, 1419-1426.

Paranhos, L. (2023). Predicting inflation with recurrent neural networks. Bank of England.

Qiu, J., Wang, B & Zhou, C (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. PLoS ONE 15(1).

Shih, C. H., Yan, B. C., Liu, S. H. & Chen, B. (2017). Investigating siamese LSTM networks for text categorization. Asia-Pacific Signal & Information Processing Association Annual Summit & Conference, Kuala Lumpur, Malaysia, 641-646.

Ta, V., Liu, C. & Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using Long-Short Term Memory Network in quantitative trading. Applied science 10(2),437.

Tanya, B. (2021). Machine learning in finance: Application, benefits, challenges. https://forbytes.com/blog/machine-learning-in-finance/.

Tölö, E. (2019). Predicting systemic financial crises with recurrent neural networks. Bank of Finland Research Discussion Papers, No. 14/2019.

Veeresh, K.V. (2021). Applications of machine learning in finance using LSTM algorithm. International Journal of Innovative Research in Science, Engineering and Technology, 10(5), 4889-4895

Vengertsev, D. & Sherman, E. (2020). Recurrent neural network properties and their verification with Monte Carlo Techniques. CEUR Workshop Proceedings, 2560, 178-185.

Xu T., Zhang J. & Ma, Z. . (2017). Deep LSTM for large vocabulary continuous speech recognition. 4.

Zhang, T. (2022). Testing the market efficiency by LSTM and SVM. Proceedings of the 2022 2nd International Conference on Economic Development and Business Culture.