
Deep Learning on SPY Stock Prices Using Improved Multi-Step LSTM

Yulong Eric Lian

Amherst Regional High School

Amherst, MA, 01002. USA

email: yulonglian88@gmail.com

Abstract

Stock price prediction has been a widely pursued topic by researchers in recent years due to the great impact that significant research can have on the economy. LSTM is commonly used for stock price prediction as it has strong time series predictive capabilities. However, it is limited by its loss function, which only takes one parameter (predicted stock price) into account. This paper proposes a multivariate multi-step, vector output predictive model using LSTM, with both the stock price and the relative return as inputs of the neural network. Furthermore, a novel loss function that combines both the standard mean squared error, and the relative return mean squared error to hone accuracy is introduced. The model's predictive capabilities are demonstrated on the S&P 500 Index. This improved LSTM model reaches a test MSE of 0.0076, which is a result that is significantly stronger than results demonstrated by standard LSTM stock prediction networks, and which outperforms most of the LSTM models mentioned in the literature.

Keywords: SPY, Stock, Multistep, LSTM, Deep Learning, S&P 500, Multivariate

JEL Code: C45, C53, E17, G17

How to cite:

Lian, Y. E. (2022). Deep learning on SPY stock prices using improved multi-step LSTM *International Journal of Finance Research*, 3(2). 135 – 146. DOI: <https://doi.org/10.47747/ijfr.v3i2.785>

1. Introduction

The inherent volatility of the stock market makes prediction a very difficult process. As such, the individual trader cannot profitably depend on their own intuition to predict stock. Therefore, a precise method of prediction is in high demand. This paper will showcase an improvement to the standard LSTM model for stock prediction by predicting Standard and Poor's 500 (SPY) stock index. The S&P500 is a stock market index that tracks the largest 500 companies in America. It is widely acknowledged by investors to be the single strongest indicator of American stock market

performance. Therefore, the ability to accurately predict the S&P 500's stock value is of paramount importance to the government, investors, and investment companies.

Technical indicators, including Bollinger Bands and Relative Strength Index (RSI), have been very popular for traders to predict stock price. We can see from Figure 1 that stock prices of SPY 500 has been on a steady incline in the past decade, with a surge in late 2021. It should be noted that the S&P 500 is less likely to be affected by outstanding events which may drastically influence one company, as it is a conglomeration of the largest companies and is only affected by nationwide trends. This makes the data relatively less volatile than most stocks.

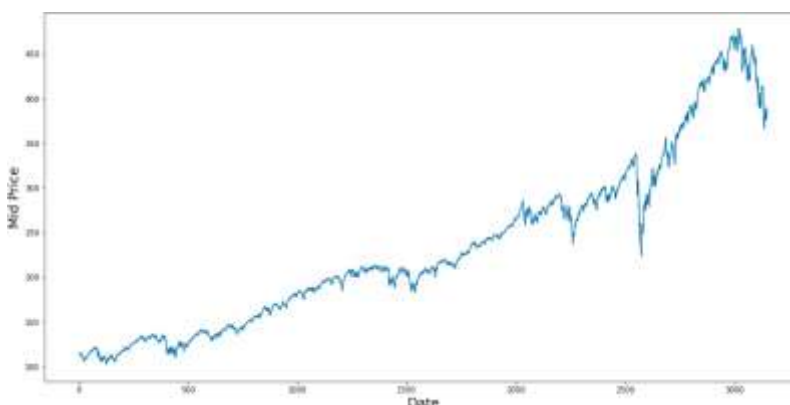


Figure 1: SPY stock price from 2010-01-01 to 2022-07-08

Bollinger band is an efficient indicator for traders to predict the short-term trend of a stock. Bollinger bands are placed at one standard deviation above and below the simple moving average (SMA) of the price. This makes it a strong trend-following indicator. We can see that up to now, there has been a strong decline to the present. The RSI also indicate that there will be a short-term incline starting in July 2022.



Figure 2: Bollinger Bands of SPY stock from 2021-Jan to 2022-July.

Another important technical indicator is the Relative Strength Index, often abbreviated as RSI, which is used as a momentum indicator, see Figure 3. Essentially, it measures how overbought or oversold a stock is. With a range of 0 to 100, it is usually considered overbought when above 70, and oversold when below 30. RSI from the past year indicates that the momentum has been fairly stable, fluctuating between 70 and 30. But by July 2022, the stock has an up-momentum in buying strength, because the stock price is low and affordable, which attracts more investors.



Figure 3: Relative Strength Index of SPY stock from 2021-Jan to 2022-July

The contribution of this research is that a Multivariate Multi-Step, vector output predictive model using LSTM, with both the stock price and the relative return as the input of the neural network is built. A novel improvement is that a new loss function which is based on the idea of the

physically-informed neural network is introduced. More precisely the formula of the relative return as a loss function is used. The improved loss function combines the standard mean squared error loss function with a relative re- turn mean squared error. The improved LSTM model is shown to be able to reach standards of accuracy beyond the prediction capability of standard LSTM stock prediction models. Furthermore, this can be applied to LSTM variants, such as previously discussed wavelet transformation LSTM neural networks. This improvement can also be applied on a large variety of stock, but is demonstrated convincingly on the SPY index.

2. Literature review

The success of Neural Networks (NN) in other applications such as speech and image recognition, and weather forecasting have prompted research to be done on stock market prediction. Studies such as Atsalakis and Valavanis (2009) now say that the use of neural networks is one of the most common methods for future financial stock prediction. Compared to stock predictions using more traditional models, such as Mustapa and Ismail (2019), who used an ARIMA-Garch model to predict the monthly stock price of S&P 500 (SPY) stock, stock predictions that implement LSTM have been shown to significantly outperform the predictions produced by ARIMA, Muncharaz (2020). Similarly, Borovkova and Tsiamas (2019), whom used ensembles of LSTM models to predict the direction of large cap US stocks with a large variety of technical indicators including open, close, bollinger bands, and relative strength index as network inputs, found that LSTM models significantly outperformed benchmark lasso and ridge regression models. Research has also been done on the comparison between LSTM and alternative methods of prediction, like Patel et al. (2015). They tackled the problem of predicting movement of Indian stock using Artificial Neural Network (ANN), Support Vector Machine (SVM), random forests (RF) and naive-Bayes by using two different inputs for all models; the first approach involving computation of ten technical parameters using stock trading data (open, high, low close prices), and the second approach representing technical parameters as trend deterministic data. They found that performance for all models improved when technical parameters were expressed as trend deterministic data. However, it has been shown that LSTM networks outperforms most alternative models of prediction. The authors of [8] used LSTM to predict directional movements of SPY stock. They found that LSTM networks outperformed classification methods that did not use memory, namely a random forest (RAF), a deep neural net (DNN), and a logistic regression classifier (LOG). And furthermore, Lakshminarayanan and McCrae (2019), who did an extensive comparison of stock market prediction on the Dow Jones Index with added parameters of crude oil and gold prices found that predictions with SVM with advanced moving averages compared to LSTM were inferior in all scenarios due to the LSTM's superior ability to efficiently remember and forget information. Research such as Kamalov et al. (2020) affirms that LSTM models generally produce very strong stock price prediction results. However, even compared to the models which yield the strongest results (CNN, MLP, RNN), LSTM is able to often perform better than these models. For example, Kamalov (2020) found that LSTM was more effective than multi-layer perceptron (MLP), Convolutional Neural Network (CNN), Random Forest (RF), and relative strength index methods at predicting significant changes in stock price.

Recently, research has been done on variations of and additions to the LSTM model which may produce better results than the standard LSTM model. In Bao et al. (2017), Bao et al. found that applying a wavelet transformation to a financial data set to de-noise it, then passing the de-noised data to stacked autoencoders to generate meaningful features, and finally feeding it to LSTM networks produced better predictions than simpler models. Then, in Liang et al. (2019), the authors proposed a multioptimal combination wavelet transform (MOCWT) method to de-noise the financial data. They found that the MOCWT method outperformed the original wavelet method. In Qiu et al. (2020), the authors proposed that a wavelet transform to reduce noise in historical stock data combined with a LSTM model and an attention mechanism would perform better than a standard LSTM model, a standard gated recurrent unit NN model, and a wavelet LSTM model on predicting SPY stock. The proposed wavelet transform LSTM model with attention outperformed all other tested models, with a test mean-squared error of less than 0.05. In a different path, there has also been research done on multi-variate multi-step LSTM. This classification of LSTM essentially defines the class of LSTM that not only predicts some dependent variable y based on multiple independent variables x (multivariate), but also predicts more than one time unit per time step. (multi-step). For example, yang2022multivariate first used LASSO and Random Forest models to filter the features and to produce a dataset of a predetermined size. Then, they compared an ARIMA model, a multivariate multistep LSTM model, and a hybrid multi-step ARIMA- LSTM model to predict tuberculosis outbreaks in the Liaoning Province, China. The result of the study was that the multivariate 2-step LSTM model reduced the error of the LSTM model by 15.94%, and the 3-step ARIMA-LSTM reduced the error of the LSTM model by 33.14%, show that multi-step LSTM produce very strong results when applied to time series prediction. In Unterluggauer et al. (2021), two multi-step LSTM models which produced 4 time step and 96 time step predictions respectively were tested on electric vehicle charging site volume prediction. Gathering real charging data from shopping centers, residential, public, and workplace charging sites, and forecasts of the aggregated charging load being in 15-min resolutions, they found that that both models produced strong predictions, with the 4 time step model (which corresponds to predicting an hour ahead) outperforming the 96 time step model (which corresponds to predict a day ahead). In Wu et al. (2021), they used a long short term memory model combined with Critical Point Search to produce a forecast for train delays. This multi-step model showed significant improvement when compared to standard LSTM models at predicting train delays. However, there has only been very limited research done on multi-step LSTM models done on stock prediction. Aasi et al. (2021) used a multivariate, multistep LSTM model. They combined this model with many data sources, such as Google Search Trends, e-News headlines, and Tweets which effectively gauge public sentiment on Apple (APPL) stock. Using internet search trends, reactions to current events, Twitter data, and historical stock returns as input data, they were able to predict 7 days per time step, and they found that their proposed model outperformed ARIMA and Random forest models trained on the same data set, along with most other LSTM models.

Specifically for predicting SPY stock price, there have been novel methods to enhance accuracy. It is known that neural networks must be trained on a large set of data. Sun et al. (2017) have shown that neural network models which were applied to image classification problems increase logarithmically in effectiveness with the size of data. To solve this problem, Lee and Kang (2020) used only data from individual companies instead of SPY stock price, because SPY is

computed by averaging top companies. This was done as an effort to increase the amount of data available to analyze, which successfully improved accuracy rate compared to other LSTM models.

3. Research Method

In this paper, the Long-short term memory (LSTM) neural network is used to forecast the SPY stock price. The LSTM recurrent neural networks are able to almost seamlessly model problems with multiple input variables. This is especially advantageous in time series forecasting, where classical linear methods can be difficult to adapt to multivariate or multiple input forecasting problems.

We begin with an introduction of the structure of a neural network. An artificial Neural Network usually contains 3 stages of layers: the input layer, the hidden layers, and the output layer. The layers are made of nodes: the input layer has the same amount of nodes as the dimension of the input data, and the nodes of the input layer connect to the nodes of the hidden layers via “synapses”, each synapse having a weight and a bias which are adjusted through training. The nodes in the hidden layer apply an “activation function” to the weighted sum of the inputs. The sigmoid function is used as the activation function, which is defined as

$$S(X) = \frac{1}{1 + e^{-x}}$$

, and is extensively used as an activation function in the input, output, and forget gates of LSTM because it “squishes” data into the range [0,1]. In the first forward pass, random values are assigned to the weights and biases. After the forward propagating process finishes and we obtain values for the output layer, the loss function is calculated and then the back propagating process begins. The backpropagation process goes from the output layer to the input layer, and uses a gradient descent method on the loss function to optimize the weights of each synapse. The forward and backwards propagation processes are then repeated many times. The amount of time this happens is usually measured in “epochs”, which is formally defined as the number of times an entire dataset is passed forward and backwards through the neural network.

Recurrent Neural Networks (RNN) are neural networks that have “memory”. They are able to use previous inputs to help predict the current input and output. While in feedforward neural networks all input is taken in at once, RNN goes through each input in a sequence and does calculations for each input before producing an output called a “hidden state”. This “hidden state” is then combined with the next input to calculate the new hidden state. In this way, RNN’s are able to take information from previous inputs into account. Here, the Multivariate, Multi-step, Vector-output LSTM neural network, which is one of the most complex RNN model, is used.

The data feeding into the LSTM gates are the input at the current time step and the hidden state of the previous time step. They are processed by three fully connected layers with a Sigmoid activation function to compute the values of the input, forget, and output gates. Define x_t as the

input at timestep t , C_t is the cell state, h_t is the hidden state, f_t is the output. The forward process of LSTM to update these values is shown in below equations:

$$\begin{aligned} X'_t &= [x_t, h_{t-1}] \\ S_t &= \tanh(W_s x'_t + b_s) \\ C_t &= i_t \cdot S_t + f_t \cdot C_{t-1} \\ &= o_t \tanh(C_t) \end{aligned}$$

where i_t, o_t, f_t are called the input, output and forget gates given by:

$$\begin{aligned} i_t &= \sigma(W_i x'_t + b_i) \\ o_t &= \sigma(W_o x'_t + b_o) \\ f_t &= \sigma(W_f x'_t + b_f) \end{aligned}$$

We call C_t the *memory cell* at time t , and it is arguably the main component of the LSTM architecture. We can think of the LSTM in the following manner:

$$C_t = i_t \odot S_t + f_t \odot C_{t-1}$$

$$\text{memory}_t = \text{read} \odot \text{input}_t + \text{remember} \odot \text{memory}_{t-1}$$

The training process will try to learn the parameters W_i, W_a, W_f, W_o . We need to unfold the input training sequence into a feed-forward neural network, as shown in Figure 4. Here Γ_f, Γ_i and Γ_o

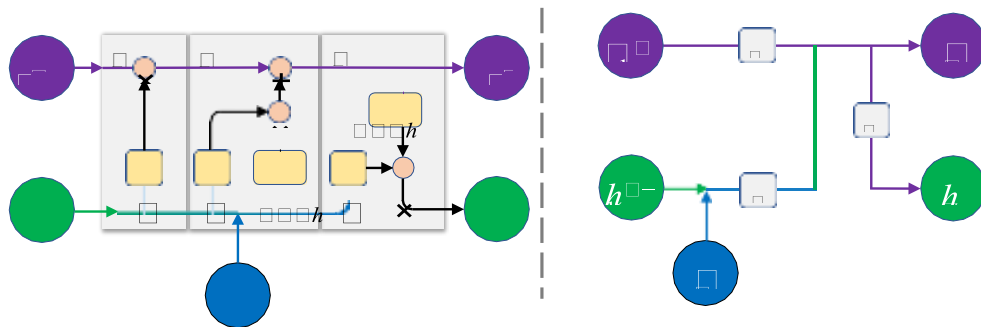


Figure 4: Network structure of LSTM neural network

represents the forget gate, input gate, and the output gate.

LSTM can be used to build deep learning models that have the ability to learn complex patterns in massive data stacks. To improve the accuracy of forecasting the movement of the stock market index, a multivariate time-series forecasting model is created by an improved LSTM, which implements a multivariate time-series analysis method into the model for simultaneous forecasting of parallel time-series using series correlation analysis.

4. Result and discussion

The data used in this paper consists of daily adjusted price extracted from yahoo finance of SPY

stock from 2010-01-01 to 2022-07-08. We denote the stock prices as a time series $S_t, t = 0, \dots, T$. To improve the forecasting of the stock price, we also use another time series: the relative return.

The relative return is the profit or loss an investor expects from an investment. The expected value of the relative return predicts future returns. This financial concept can be useful when there is a robust pool of historical data on the returns of a particular investment. The formula to calculate relative return is as follows:

$$R_{t+1} = \frac{S_{t+1} - S_t}{S_t}$$

The time series of relative returns ($R_t \geq 1$) will be used in the training to improve the accuracy rate of the forecast. More precisely, we take the 2-dimension time series ($S_t, R_t \geq 1$) as the input of the LSTM neural network. The output vector has two steps. More precisely, given $[(S_t, R_t), t = 1, \dots, s]$, we need to predict $[S_{s+1}, R_{s+1}), (S_{s+2}, R_{s+2})]$. Thus the neural network defines two functions f_S and f_R , such that

$$(\hat{S}_{T+1}, \hat{S}_{T+2}) = f_S(S_{t+1}, S_{t+2}, \dots, S_{t+s}; R_{t+1}, R_{t+2}, \dots, R_{t+s}) (\hat{R}_{T+1}, \hat{R}_{T+2}) = f_R(S_{t+1}, S_{t+2}, \dots, S_{t+s}; R_{t+1}, R_{t+2}, \dots, R_{t+s})$$

where $s > 0$ is the sample size, which is taken to be $s = 6$ in our simulations. Our deep learning of the SPY stock price is based on the construction of an ensemble model that can predict multiple financial time-series data simultaneously.

80% of the data was used for training, and 20% of the data for testing. Six preceding values were used to predict two values.

A unique feature of this model is that we input both the stock data and the relative return values into the model. Hence, why the output shape for the input layer is (6,2).

Layers	Output Shape	Parameters
Input Layer	(6, 2)	0
LSTM1	(6, 500)	1006000
LSTM2	(200)	560800
dense1	(32)	6432
dense2	(4)	132
Reshape	(2,2)	0

We reshape the model to dimension size (2,2) at the end so that it will output the two predicted stock price values, but also the two corresponding relative return values.

The standard LSTM model uses some integer x preceding values to predict 1 value. The standardMSE loss function, defined as

$$L_{mse}(y_{true}, y_{pred}) = \frac{1}{N} \sum_{i=1}^N (y_{true}(i) - y_{pred}(i))^2$$

is then used. However, as shown in the literature, deep learning neural networks' ability to predict stock prices are limited.

To improve the accuracy, we propose a novel loss function, by implementing the relative return function. We first construct the multivariate, multi-step LSTM, with the 2-dimension output $(S_{t+s+1}, S_{t+s+2}; R_{t+s+1}, R_{t+s+2})$. The new loss function is defined as

$$L_D = \frac{1}{s} \sum_{k=1}^s \left(R_{t+s+2} - \frac{S_{t+s+2} - S_{t+s+1}}{S_{t+s+1}} \right)^2$$

which uses the definition of the relative return R_t .

Our new total loss function is

$$L_{loss} = \lambda L_{mse} + (1 - \lambda) L_D$$

where $\lambda > 0$ is a regulation parameter. In our training process, we take $\lambda = 0.6$ in the pre-trainings, and then $\lambda = 0.4$ in further trainings, in order to improve the accuracy rates.

The motivation to do this is because if relative return is added to the loss function, the improved model will have to learn weights and biases that will be optimal not only for the relative return loss function, but also for standard stock price loss function. Because this effectively cuts down on the set of acceptable weight and bias values, this model should achieve higher accuracy after extensive training.

Training loss continued to decrease incrementally until finally stabilizing at around 10,000 epochs. The improved LSTM model is evaluated using MSE, which is a standard accuracy criterion in machine learning. Our training MSE after 10,000 epoch is $2.6 \cdot 10^{-6}$, and our test MSE is 0.00764.



Figure 5: Prediction of stock prices compared with test set

As we can see, both the training and test MSE are objectively strong. The improved LSTM model outperforms the LSTM model used to predict SPY in [3], which reaches a test mean-squared error of 0.12.

In Liang et al. (2019), for the S&P 500 dataset, data from January 3, 2000 to May 16, 2019 is used. This is the comparison of the test MSE of the LSTM, Wavelet-LSTM, GRU, and Wavelet-LSTM + attention models that are found in Liang et al. (2019) with our improved LSTM model:

	MSE
LSTM	0.1208
WLSTM	0.1067
GRU	0.1000
WLSTM + attention	0.0546
Improved LSTM	0.0076

5. Conclusions

This paper's proposed improvement to the loss function produces an accurate prediction which significantly outperforms the standard LSTM stock prediction model. The results of training, thus, indicates that adding limiting parameters to the loss function by adding a relative return MSE allows for a more accurate prediction.

The proposed architecture of the LSTM (typically used in state-of-art language modeling deep learning tasks) can be considered very promising as it has proven to be able to predict well in comparison with other strategies employed and tested in the literature. However, the relative return limiting parameter may not be optimal. Furthermore, if one includes a set of relevant technical indicators in the training process, it may provide a positive influence to the performance of the model. In the same way, employing all the features available in the financial time series such as High, Low, Open, Close and Volume prices, which are the most relevant variables, enriches the model. More research may be done on different limiting parameters and additional features.

References

Aasi, B., Imtiaz, S. A., Qadeer, H. A., Singarajah, M., and Kashef, R. (2021). Stock price prediction using a multivariate multistep lstm: A sentiment and public engagement analysis model. In *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–8. IEEE.

- Atsalakis, G. S. and Valavanis, K. P. (2009). Surveying stock market forecasting techniques—part ii: Soft computing methods. *Expert Systems with applications*, 36(3):5932–5941.
- Bao, W., Yue, J., and Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944.
- Borovkova, S. and Tsiamas, I. (2019). An ensemble of lstm neural networks for high-frequency stock market classification. *Journal of Forecasting*, 38(6):600–619.
- Kamalov, F. (2020). Forecasting significant stock price changes using neural networks. *Neural Computing and Applications*, 32(23):17655–17667.
- Kamalov, F., Smail, L., and Gurrib, I. (2020). Stock price forecast with deep learning. In *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pages 1098–1102. IEEE.
- Lakshminarayanan, S. K. and McCrae, J. P. (2019). A comparative study of svm and lstm deep learning algorithms for stock market prediction. In *AICS*, pages 446–457.
- Lee, J. and Kang, J. (2020). Effectively training neural networks for stock index prediction: Predicting the s&p 500 index without using its index data. *PloS one*, 15(4):e0230635.
- Liang, X., Ge, Z., Sun, L., He, M., and Chen, H. (2019). Lstm with wavelet transform based data preprocessing for stock price prediction. *Mathematical Problems in Engineering*, 2019.
- Muncharaz, J. O. (2020). Comparing classic time series models and the lstm recurrent neural network: An application to s&p 500 stocks. *Finance, Markets and Valuation*, 6(2):137–148.
- Mustapa, F. H. and Ismail, M. T. (2019). Modelling and forecasting s&p 500 stock prices using hybrid arima-garch model. In *Journal of Physics: Conference Series*, volume 1366, page 012130. IOP Publishing.
- Patel, J., Shah, S., Thakkar, P., and Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, 42(1):259–268.
- Qiu, J., Wang, B., and Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS one*, 15(1):e0227222.
- Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.
- Unterluggauer, T., Rauma, K., Järventausta, P., and Rehtanz, C. (2021). Short-term load forecasting at electric vehicle charging sites using a multivariate multi-step long short-term memory: A case study from finland. *IET Electrical Systems in Transportation*, 11(4):405–419.
- Wu, J., Wu, Q., Zhou, L., Cai, C., Du, B., Zhai, Y., Wei, W., Shen, J., and Zhou, Q. (2021).

Multivariate multi-step train delay forecasting: A hybrid lstm-cps solution. Technical report.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>)