

## Forecasting Financial Time Series Using Deep Learning Networks: Evidence from Long-Short Term Memory and Gated Recurrent Unit

**Mohammadreza Ghadimpour**

MSc., Department of Financial Engineering, Faculty of Industrial Engineering, Khajeh Nasir Toosi University of Technology, Tehran, Iran. (Email: mrghadimpour@gmail.com)

**Seyed Babak Ebrahimi\***

\*Corresponding Author, Assistant Professor, Department of Financial Engineering, Faculty of Industrial Engineering, Khajeh Nasir Toosi University of Technology, Tehran, Iran. (Email: b\_ebrahimi@kntu.ac.ir)

Iranian Journal of Finance, 2022, Vol. 6, No.4, pp. 81-94.

Publisher: Iran Finance Association

doi: <https://doi.org/10.30699/IJF.2022.313164.1286>

Article Type: Original Article

© Copyright: Author(s)

Type of License: Creative Commons License (CC-BY 4.0)

Received: November 30, 2021

Received in revised form: April 05, 2022

Accepted: April 30, 2022

Published online: October 10, 2022



### Abstract

The ability to predict the stock market and analyze market trends is invaluable to researchers and anyone interested in investing. However, this task is a challenging problem due to a large number of parameters and unpredictable noise that may affect the stock price. To overcome this issue, researchers have employed numerous approaches such as Moving Average (MA), Support Vector Machine (SVM), and Neural Networks. With technological advances, deep learning methods have become popular in processing time-series data. In this paper, we compare two recently introduced deep learning models, namely

a Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), in forecasting daily movements of the Standard & Poor (S&P 500) index using the daily closing price of this index from 14/5/1991 to 14/5/2021. Results show that both models are effective and accurate in stock market prediction. In this case study, the mean squared error (MSE) and mean absolute error (MAE) for the GRU model are slightly lower than the LSTM model; hence, GRU outperformed the LSTM model despite its simpler structure. The results of this study are applicable in various instances where it is challenging to identify patterns among large volumes of unstructured data, such as medical data analysis, text mining, and financial time series modeling.

**Keywords:** Machine Learning; Recurrent Neural Network; Long Short-Term Memory; Gated Recurrent Unit; Financial time series.

## Introduction

Forecasting the future price of a financial asset has always been an interesting subject for researchers and investors who want to beat the market, take higher returns, and reduce investment risks. However, predicting stock market movement is not an easy task due to the inherent complexity of the stock market and the chaos caused by many factors, such as economic, political, and company conditions. Recurrent neural networks (RNN) are one of the most common and powerful approaches to sequential data processing. This paper evaluates the performance of those recently proposed recurrent units (LSTM unit and GRU) on sequence modeling. Long-Short term memory (LSTM) is one of the most successful RNN networks which introduce the memory cell and gate structure. This architecture can handle long sequences of input by distinguishing between recent and early inputs and forgetting the memory it considers irrelevant.

On the other hand, we have Gated Recurrent Unit (GRU) proposed by Cho et al. (2014). Like the LSTM, the GRU uses gating units to process data flow, but this architecture does not suffer from cell state, uses the hidden state to transfer information, and uses fewer gates than LSTM. In this study, we apply both LSTM and GRU models to predict the movement of the S&P 500 index. We use 30 years of daily historical data for training and evaluating models. The results show that both models have good precision in predicting market movement, while GRU models show better results than LSTM. The remainder of this paper is organized as follows: A brief background knowledge and related works are provided in Section 2. Section 2 presents the methodology and explains each process in detail. Section 4 represents the results of the experiment. Finally, Section 5 provides some conclusions.

## Literature Review

In an analysis made using sequential data, it is necessary to process related time series at every time step and save the sequence's entire state. RNN networks can handle this task using a recurrent hidden state whose activation at each time is dependent on that of the previous time. Therefore, it solves the problem of forgetting previous inputs. However, Bengio et al. (1994) show that it is difficult to train RNNs to capture long-term dependencies because the gradients tend to either vanish (most of the time) or explode (rarely, but with severe effects). Hochreiter and Schmidhuber (1997) introduced the Long Short-Term Memory (LSTM) model to solve this issue. LSTM uses memory cells instead of neurons, and each cell has three adaptive and multiplicative units called "gates". These gates keep the error flow constant, allowing for weight adjustment and truncation of the gradient when its information is unnecessary. This approach is widely used in forecasting stock market prices and movements (Chen et al. (2015), Heaton et al. (2016), Jia (2016). Gao et al. (2017) evaluated LSTM and SVM prediction accuracy with different lengths of a dataset. LSTM shows better results in all experiments and proves its high performance even in short time series. Shah et al. (2018) compared the performance of LSTM and DNN in the case of stock market forecasting and showed that LSTM outperforms DNN in weekly predictions. Wen and Yuan (2018) merged CNN and LSTM to increase the efficiency of these models. In the proposed model, the raw data are first passed through the CNN layers to extract the data features and become a time series as input for the LSTM. The LSTM network then re-examines this input to extract more features and finally performs the prediction. The proposed network was about 2% more accurate than CNN and about 1% more accurate than LSTM. Baek and Kim (2018) introduced a new model using two LSTM networks together. The first module is designed to prevent network overfitting and uses different data dimensions as input. The second module, which is employed only for market forecasting, uses the output of the previous module as input. GRU networks were introduced in 2014 as a variation of the LSTM network. The main idea of GRU is to simplify things by using fewer gates and also replace cell states with hidden states. These changes help GRU networks train faster because they have fewer parameters than LSTM. Therefore, it has become popular in the past few years. Obaidur Rahman (2019) used the GRU network to forecast the stock market's future price. The results show that the proposed method can predict future prices successfully with good accuracy. Pandey et al. (2020) combined GRU and ARIMA models and built a hybrid model. They claim that their proposed model can outperform any other RNN-ML algorithm present in the

market.

Overall, the articles reviewed in this study show that both LSTM and GRU are very powerful approaches to forecasting time series. The present study aims to answer the question that which of these methods is more suitable and more effective in forecasting the stock market and analyzing its trends and movements.

## Research Methodology

### 1. Long Short-Term Memory Neural Network

LSTM networks use the structure called memory cells. Each memory cell contains four different parts: an input gate, a forget gate, an output gate, and a neuron unit with a self-recurrent connection (connection to itself). These three gates use the activation (transfer) function to compute a value between 0 and 1. Next, based on this value, the memory cell decides which part of data should be read, stored, or forgotten. The input gate control new value flows and decides which information should be stored. Forget gate is responsible for specifying which information is relevant and should be stored in a cell state. Finally, based on input and current cell state, the output gate decides which information must be propagated forward and which one affects other cells. Figure 1 demonstrates the data flow in a memory cell.

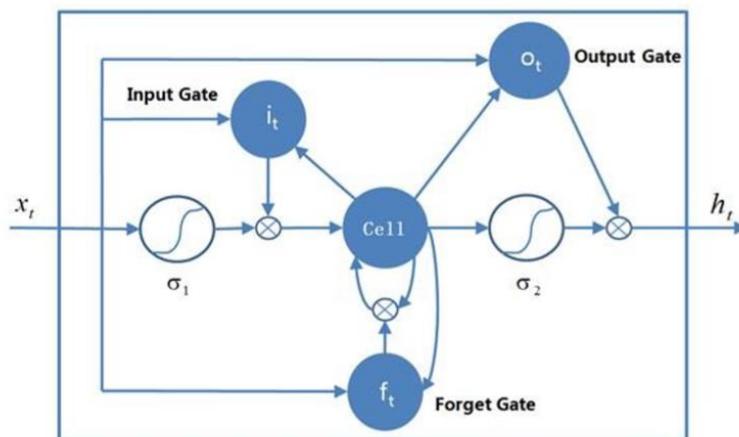


Figure 1. Memory cell structure

At each time step, the memory cell calculates the state of each gate, input candidate, and cell state using the following equations:

$$i_t = \sigma_1(W_i x_t + U_i c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma_1(W_f x_t + U_f c_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma_1(W_o x_t + U_o c_{t-1} + b_o) \quad (3)$$

$$c_t = f_t * c_{t-1} + i_t * \sigma_1(W_c x_t + b_c) \quad (4)$$

$$h_t = o_t * \sigma_2(c_t) \quad (5)$$

Where:

- $W_i$ ,  $W_f$ ,  $W_o$ ,  $W_c$ ,  $U_i$ ,  $U_f$ , and  $U_o$  are weight matrices.
- $b_i$ ,  $b_f$ ,  $b_o$  and  $b_c$  are bias vectors.
- $\sigma_1$  and  $\sigma_2$  are activation functions.
- $x_t$  is our input vector in time step t.
- $i_t$ ,  $f_t$ , and  $o_t$  are gates vectors.
- $c_t$  is the cell state in time t.
- $h_t$  is the output vector of the memory cell.

## 2. Gated Recurrent Unit

GRU uses two gates, namely the update gate and resets gate. Same as LSTM, these gates use activation functions. The input data for time step t and the information from previous time steps are added together and sent to the Update gate. This gate works like the combination of input gate and forget gate in the LSTM network and decides how much of this data flow needs to be passed along to the future. The reset gate is responsible for deciding how much of the previously computed state should be forgotten and storing the relevant information. Finally, to calculate the output of the current unit, the update gate decides what to collect from the current memory content and previous steps.

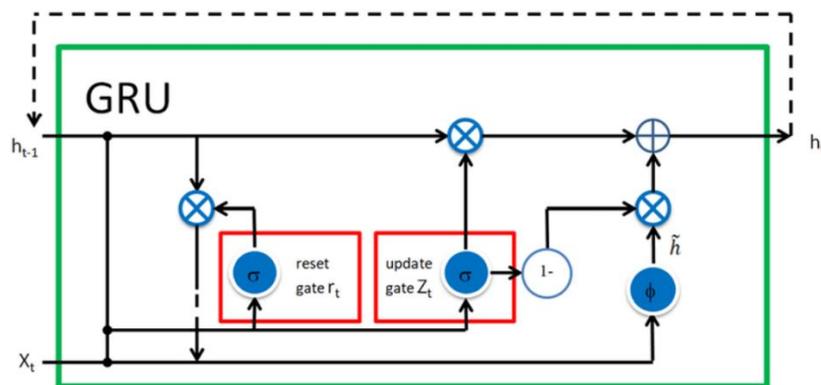


Figure 2. GRU cell structure

Eqs. (6) to (9) are used to calculate these respective gates:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (6)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (7)$$

$$H_t = \tanh(W_{x_t} + r_t \odot U_H h_{t-1}) \quad (8)$$

$$h_t = (1 - z_t)h_{t-1} + z_t H_t \quad (9)$$

Where:

- $x_t$  is our input vector in time step  $t$ .
- $W_z, W_r, W_{x_t}, U_z, U_r,$  and  $U_H$  are weight matrices.
- $\odot$  is an element-wise multiplication.
- $r_t$  and  $z_t$  are gates vectors.
- $H_t$  is current memory content.
- $h_t$  is the output vector of the current GRU cell.

It is easy to see the similarities between the LSTM unit and the GRU unit, but there are some differences as well. For example, in LSTM, the output gate controls the amount of memory content used by other units in the network, while in GRU, we have full content with no control. Another difference is in the location of the input gate in LSTM and the reset gate in GRU. The input gate's position enables LSTM to control the amount of the new memory content added to the memory cell independently from the forget gate. On the other hand, the GRU cannot control the amount of the candidate content independently because it uses an update gate to control it. Thus, based on these similarities and differences, we cannot easily choose between these gating units. This issue motivated us to compare LSTM and GRU in the financial area.

### 3. Evaluation

In this study, Mean Square Error (MSE) and Mean Absolute Error (MAE) were used to evaluate our model's efficiency. MSE is calculated using the following formula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (10)$$

On the other hand, MAE is given by:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (11)$$

where  $n$  is the total number of sample data,  $y_i$  is the real value of time  $i$ , and  $\hat{y}_i$  is the predicted value at the time  $i$  obtained from a particular model.

#### 4 .Obtaining dataset and preprocessing

In the models studied in this work, we use historical index data instead of historical stock data because of their less noise. Therefore, it helps models to be more accurate. The dataset consists of the daily closing price of the S&P 500 index from 14/5/1991 to 14/5/2021. Since we use recurrent networks, the input dataset needs to be changed into sequence data. Hence, a sliding window is applied to the entire dataset. Initially, 10 days window length was used for both models, and then different window sizes were tested for each of them. Best results are obtained when the sliding window length is set to 5 days, rolling window to 1 day, and predicting term to 1 day. In other words, the model predicts the S&P 500 of the current day by looking at the previous 5 days. This process is described in Figure 3.

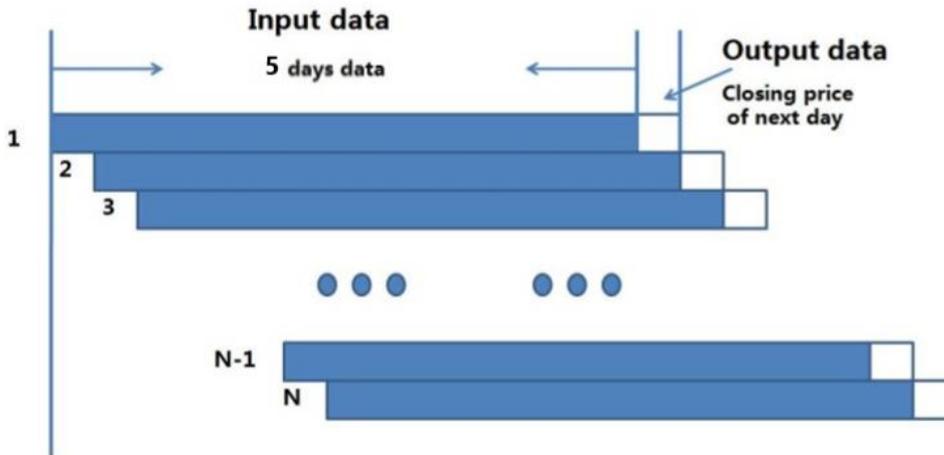


Figure 3. Diagram of building up the sequence dataset

In the next step, the dataset is divided into training (80%), validation (10%), and testing datasets (10%). Finally, we used Eq. (10) to normalize our data. Data normalization converts their value to a specific range, which helps improve the model's performance.

$$\hat{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (12)$$

Where:

- $\hat{x}_i$  is  $i^{\text{th}}$  normalized value.
- $x_i$  is  $i^{\text{th}}$  closing value.
- $x_{\max}$  and  $x_{\min}$  are maximum and minimum closing values, respectively.

## Result

First, we need to determine some parameters to reach the best results for both models. We perform various simulations using different parameters, presented in Table 1.

Table 1. model's parameters

Parameter	Number
batch-size	500
neurons	25-50-100
Window size	3-5-7-10-15
LSTM/GRU layer	1-2-3

We have the best results for both models when neurons are set to 50, and the window size is 5. Neither of the models showed improvement after adding the third layer, suggesting that two layers are enough to handle the complexity of our dataset. We also use dropout layers with a rate of 0.2 to prevent overfitting.

In the following, the obtained results are compared in the form of some graphs:

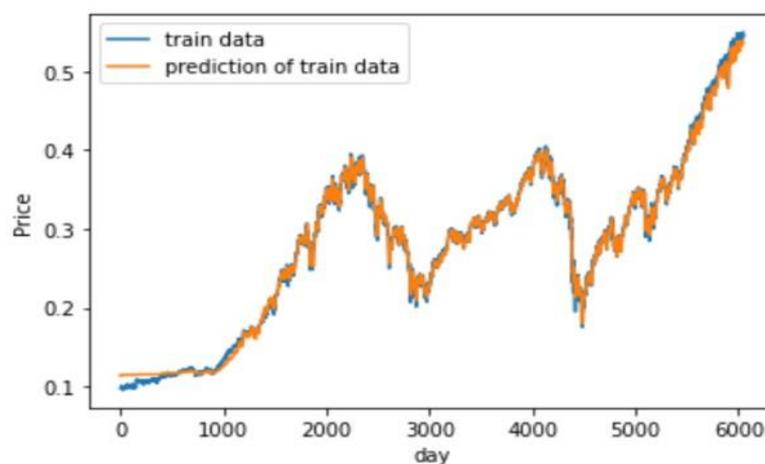


Figure 4. LSTM predictions using the train dataset

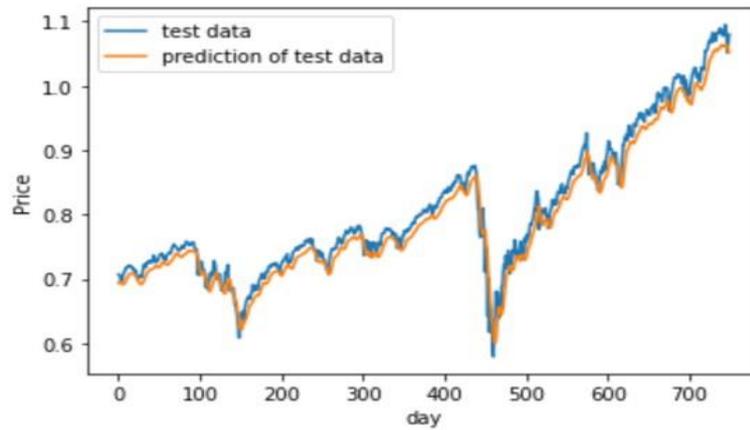


Figure 5. LSTM predictions using the test dataset

Table 2. Results of the LSTM model

Parameters	train	test
MSE	3.64e-05	0.00041
MAE	0.00459	0.01741

In Figure 5, there is a little space between predicted values and actual values, and they do not exactly match.

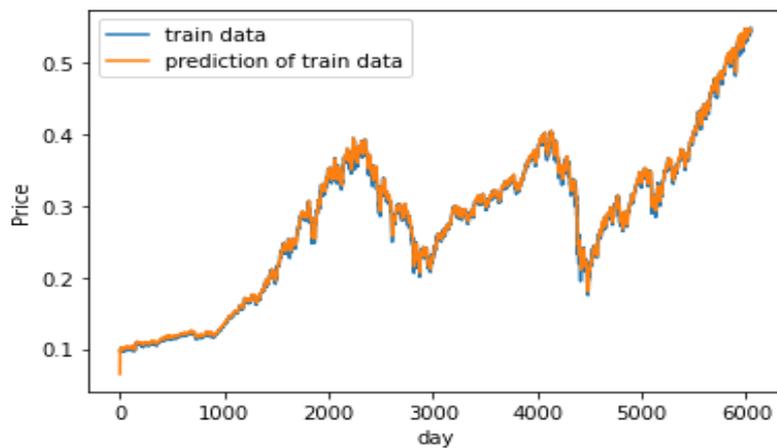


Figure 6. GRU predictions using the train dataset

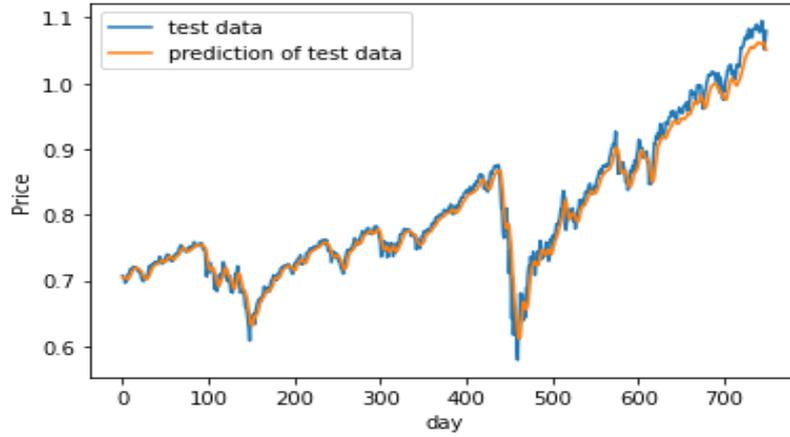


Figure 7. GRU predictions using the train dataset

Table 3. Results of the GRU model

parameters	train	test
MSE	2.2463e-05	0.000267
MAE	0.003380	0.0121326

In Figure 7, unlike Figure 5, actual values and predicted values at the end of the graph are different.

To make sure our networks are not overfitted, we use loss function graphs, Figures 8 and 9 were plotted based on normalized values. As can be seen, the values converge to 0 in both cases, proving that the networks provide a good fit to the data and are not overtrained.

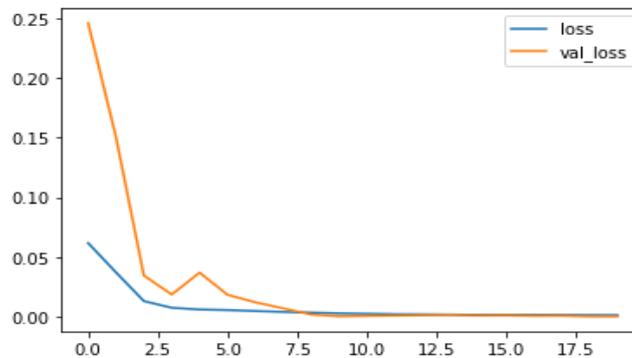


Figure 8. Training and validation losses of LSTM

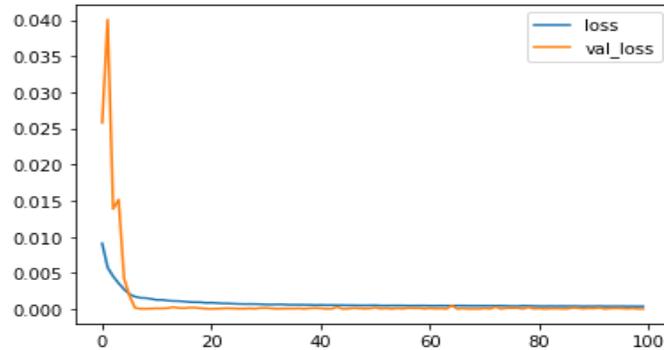


Figure 9. Training and validation losses of GRU

These results demonstrate that both models can analyze time series and make accurate predictions. Based on Tables 2 and 3, and graphs of test datasets, the GRU model outperforms the LSTM model in forecasting stock price movements. This difference could be due to GRU's simple structure, which has fewer gates with fewer parameters and thus its faster training. Hence, the GRU network shows better results and faster analyzes time series. Based on this case study, we can introduce GRU networks as the most suitable tool for financial time series analysis.

Figure 10 provides the structure of the GRU model presented in this study:

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 1, 50)	8550
dropout (Dropout)	(None, 1, 50)	0
gru_1 (GRU)	(None, 1, 50)	15300
dropout_1 (Dropout)	(None, 1, 50)	0
gru_2 (GRU)	(None, 50)	15300
dropout_2 (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51
Total params: 39,201		
Trainable params: 39,201		
Non-trainable params: 0		

Figure 10. The structure of the proposed GRU model

Our proposed model includes three dropout layers and three GRU layers, leading to 39201 parameters to learn.

## **Conclusion**

With the recent developments in deep learning approaches, these techniques have become highly popular among researchers. This paper developed LSTM and GRU models to compare these newly introduced models for forecasting stock market movements. Firstly, we collected 30 years daily close price dataset of the S&P 500 and converted it to a sequential dataset. Then, we used LSTM and GRU models to capture hidden dynamics in the historical data. The results demonstrate that both models effectively forecast stock market movements. However, in this study, GRU models outperformed the LSTM models. The good accuracy of these models helps investors, researchers, and any person interested in the stock market by providing them with valuable information about the stock market's future situation.

## **Declaration of Conflicting Interests**

The authors declared no potential conflicts of interest concerning the research, authorship and, or publication of this article.

## **Funding**

The authors received no financial support for the research, authorship and, or publication of this article.

## References

- Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert systems With Applications*, 457-480.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157-166.
- Chen, K., Zhou, Y., & Dai, F. (2015). A lstm-based method for stock returns prediction: A case study of china stock market. *2015 IEEE International Conference*.
- Cho, K., Merriënboer, B. v., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv*, 1259-1409.
- Gao, T., Chai, Y., & Liu, Y. (2017). Applying long short-term memory neural networks for predicting stock closing price. *8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. Beijing, China.
- Heaton, J., Polson, N., & Witte, J. (2016). Deep learning in finance. *arXiv preprint arXiv*.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- Jia, H. (2016). Investigation into the effectiveness of long short-term memory networks for stock price prediction. *arXiv preprint arXiv*.
- Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLoS ONE*, 14(2).
- Nandakumar, R., Uttamraj, K. R., Vishal, R., & Lokeswari, Y. V. (2018). Stock price prediction using long short-term memory. *International research journal of engineering and technology*, 5(3), 3342-3348.
- Nelson, D. Q., Pereira, A., & Oliveira, R. (2017). Stock market's price movement prediction with LSTM neural networks. *IEEE*, 1419-1426.
- Pandey, V., Mangalampalli, R., Khetre, P., & Malviya, V. (2020). Stock Prediction using Hybrid ARIMA and GRU Models. *International Journal of Engineering Research & Technology*, 9(5), 737-743.
- Rahman, M. O., Junaid, T., Hossain, S., Forhad, S. A., & Hossen, M. K. (2019). Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks. *International Journal of Computer Science and Network Security*, 19(1), 213-222.

- Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock price using LSTM. *International Journal of Science and Research*, 6(4), 1754-1756.
- Selvin, S., Gopalakrishnan, E. A., Vinayakumar, R., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. *IEEE*, 16431647.
- Shah, D., Campbell, W., & Zulkernine, F. H. (2018). A Comparative Study of LSTM and DNN for Stock Market Forecasting. *IEEE International Conference on Big Data*.
- Wen, Y., & Yuan, B. (2018). Use the CNN-LSTM network to analyze secondary market data. *2nd International Conference on Innovation in Artificial Intelligence*.
- Yao, S., Luo, L., & Peng, H. (2018). High-frequency stock trend forecast using LSTM model. *The 13th International Conference on Computer Science & Education*. Colombo, Sri Lanka.

---

**Bibliographic information of this paper for citing:**

Ghadimpour, Mohammadreza & Ebrahimi, Seyed Babak (2022). Forecasting Financial Time Series Using Deep Learning Networks: Evidence from Long-Short Term Memory and Gated Recurrent Unit. *Iranian Journal of Finance*, 6(4), 81-94.

---