CAUCHY –Jurnal Matematika Murni dan Aplikasi
Volume 8(2) (2023), Pages 16-29
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Long Short Term Memory Using Stochastic Gradient Descent and Adam for Stock Prediction

Muhammad Athanabil Andi Fawazdhia[1], Zani Anjani Rafsanjani[2,*]

[1]Department of Mathematics, Faculty of Applied Science and Technology, Ahmad Dahlan University, Yogyakarta, Indonesia
[2]Department of Mathematics, Faculty of Science and Mathematics, Diponegoro University, Indonesia

Email: zani@lecturer.undip.ac.id

ABSTRACT

The stock market is a place to carry out stock buying and selling transactions, the expected return of course has a profitable difference. Predicting stock prices can be done in various ways, one of which is by using deep learning models. Long Short Term Memory (LSTM) is a method that can be used to predict time series data. This method is a development of the Recurrent Neural Network (RNN), so this method is more complicated and powerful. To conduct training on the LSTM model, optimization is needed to minimize errors. There are lots of optimizations that can be used, but in this research, we use SGD and Adam. Several parameters such as learning rate 0.01, 0.001, 0.0001 and several variations of epochs such as epoch 25, epoch 50, epoch 100, epoch 200, epoch 400, and epoch 1000 were used in this study. The research data used are stock data of BBRI, BBNI, BMRI, and BBTN. This study also tries to predict stock prices on the next day using 5 historical stock price data, the result is that LSTM SGD and LSTM Adam succeeded in predicting the next day.

Keywords: Long Short Term Memory; Stochastic Gradient Descent; Adam; Predict; Stocks

INTRODUCTION

Investors can make decisions to buy and sell their shares to achieve high returns with tools [1]. Many methods can be used to predict, such as arima [2], ordinary differential equations [3], artificial neural network (ANN) [4], and recurrent neural network (RNN) [5] are some examples of methods that can be used for prediction.

Artificial neural network (ANN) is an artificial intelligence method inspired by neuron networks [6]. The ANN architecture has three parts, input layer, output layer, and hidden layer that connects the input layer, and the output layer. The input data on the ANN will be multiplied by the weights and added with the bias, and then the activation function is processed to obtain the output data on the ANN [7]. Recurrent neural network (RNN) is a development of ANN which has been specially designed to be able to process time series data. The data used in RNNs often use time series data [8], [9]. Data that enters the RNN architecture will be processed repeatedly.

There are problems in conducting training on RNN, namely the vanishing gradient

Submitted: October 13, 2022 Reviewed: April 04, 2023 Accepted: November 08, 2023
DOI: http://dx.doi.org/10.18860/ca.v8i2.17789

problem [10]. A vanishing gradient occurs when the gradient value is too small so have no effect when updating the weights in the backpropagation process [11]. The cause of the reduced gradient value is due to the repeated multiplication of the hidden layer RNN when carrying out the backpropagation process [11], [12].

Long Short Term Memory (LSTM) is a method proposed by Sepp Hochreiter and Jurgen Schmidhuber in a paper [10] in 1997. This method is a development of RNN. LSTM was developed to prevent vanishing gradient problems that arise when training RNN so that LSTM can process longer time series data than RNNs. 4 additional components become the architecture of the LSTM, namely forget gate, input gate, output gate, and cell state. LSTM can predict stocks with better performance and smaller errors [13].

Optimization methods are used in the ANN architecture, both traditional and developed architectures such as LSTM. The goals of the optimization method are to make improvements to the weight and bias values by minimizing errors in the architecture used. In this study, SGD and Adam were applied to the LSTM. The concept of SGD is to minimize the error value by calculating the gradient using one or several data to perform parameter updates (weights and biases) [14]. Adam was developed by [15] in 2015 and is now starting to be widely used in various studies.

This study aims to see the performance of LSTM SGD and LSTM Adam using various learning rates and epochs in training. After getting the smallest error value from each method, the last step is to make predictions using historical data.

METHODS

Normalization

In research, data with different scales are often encountered [16]. For example, there is data worth 100, and other data is worth 10000. When using data that is not scaled, it can cause model training to take longer due to large calculations, in some cases, can lead to an ineffective model [17]. This study uses a normalization method that is already quite popular, namely min max scalar.

How min max mcalar works is to use the maximum and minimum data in a data set to generate a new value. Data that has applied by min max scalar will be distributed with a range of 0 to 1 [18]. The equation of min max scalar is below:

$$x_{scaled} = \frac{x_i - x_{min}}{x_{\max} - x_{min}} \tag{1}$$

Normalization only changes the scale of the data but does not change the characteristics of the data. Figure 1 shows that the characteristics of the data do not change, and the distribution of the data changed from 0 to 1 using Min Max Scalar.

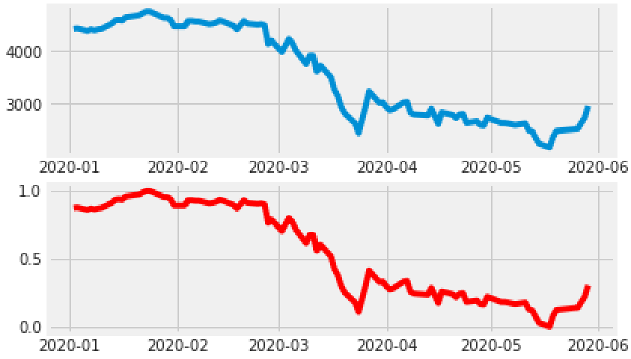
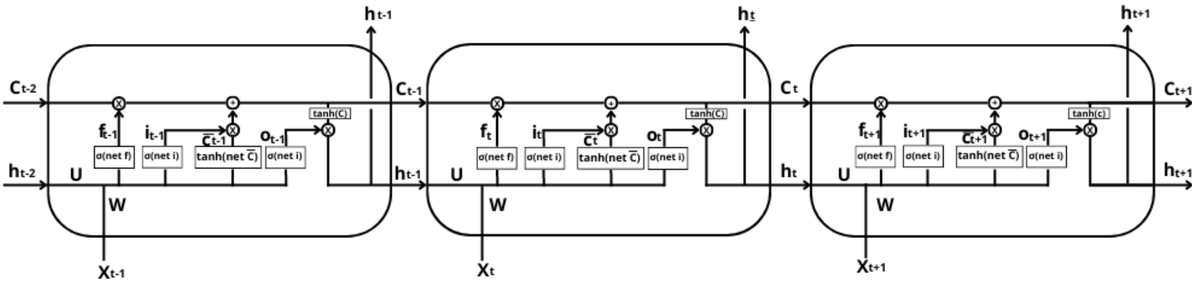


Figure 1. Illustration of Data Before Normalization (blue) and After Normalization (red)

Long Short Term Memory (LSTM)

Sepp Hochreiter and Jurgen Schmidhuber are the figures behind the emergence of the LSTM method was first introduced to the public in 1997. LSTM is a type of RNN that performs better in practice, due to the updating of the architecture and the dynamics of backpropagation in LSTM [19]. The main idea of LSTM is to add cell state and other components [20]. LSTM consists of input gate, forget gate and output gate that can adjust the cell state [21].



Gambar 2. Illustration of LSTM Architecture

The process of each component in LSTM has explained by Nurjaman research in [22], and Lipton research in [23]. The LSTM process starts at the forget gate. The incoming value is converted to range 0 to 1 with a sigmoid activation function [24].

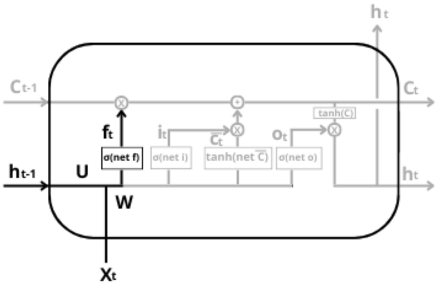


Figure 3. Forget Gate LSTM

the forget gate equation on the LSTM can be seen as follows:

$$f_t = \sigma(W_f X_t + h_{t-1} X_t + b_f) \tag{2}$$

The input gate serves to create a new value that will be passed to the cell state. Look at figure 4, two processes occur at the input gate, the value that passes through the sigmoid activation function, and passes through the tanh activation function which is called a candidate cell ($\tilde{C}$).

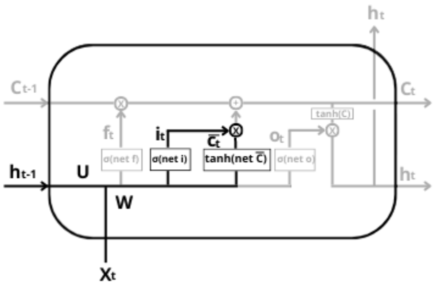


Figure 4. Input Gate LSTM

the equation for the two input gate processes can be seen as follows:

$$i_t = \sigma(W_i X_t + h_{t-1} X_t + b_i) \tag{3}$$

$$\widetilde{C_t} = tanh(W_c X_t + h_{t-1} X_t + b_c) \tag{4}$$

Cell state is the most important component in the LSTM, because it connects each LSTM cell directly [24], [25]. The value of the forget gate and input gate used to update the old cell state to the new one [26].

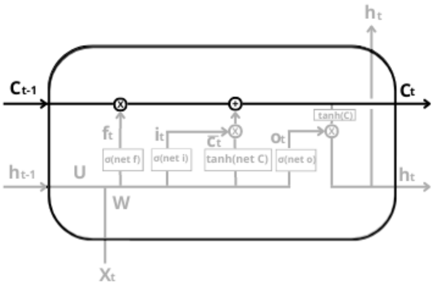


Figure 5. Cell State LSTM

the equation for performing the update is:

$$C_t = f_t \otimes C_{t-1} \oplus i_t \otimes \widetilde{C_t} \tag{5}$$

symbol $\otimes$ indicates that the multiplication process used is the element-wise product.

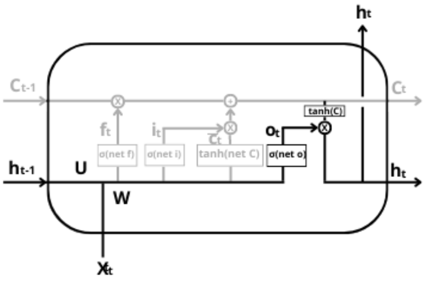


Figure 6. Output Gate LSTM

Figure 6, shows the last step in the LSTM, which is to determine the output value of the LSTM. The output value is determined by the gate output value multiplied by the cell state value that passes through the tanh activation function. The gate output equation is shown as follows:

$$o_t = \sigma(W_o X_t + h_{t-1} X_t + b_o) \tag{6}$$

after getting the cell state and gate output values, the last step is to calculate the hidden value and become the output of the LSTM.

$$h_t = \tanh(C_t) \otimes o_t \tag{7}$$

Activation Function

The activation function is a function to calculate inputs and biases that have been weighted [11]. The activation function is divided into linear activation function and non-linear activation function [27]. This study uses two non-linear activation functions. The sigmoid activation function is one of the non-linear activation functions. The sigmoid activation function will change the incoming input to be in the range of 0 to 1 [11]. The sigmoid activation function equation can be as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

the tanh activation function changes the input into a range of -1 to 1 [11].

$$tanh(x) = \frac{\mathrm{e}^{\mathrm{x}} - e^{-x}}{e^{x} + e^{-x}} \tag{9}$$

Forward Pass LSTM

The forward pass is a process to find predictive data and look for errors. Forward pass on LSTM is more complicated than forward pass RNN because there are several additional components. The input data passes through the forget gate, input gate, cell state, and output gate, and the output of the LSTM is obtained.

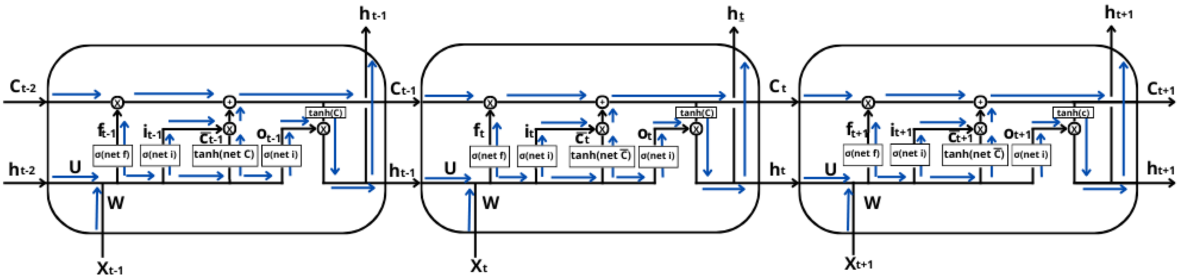


Figure 7. Forward Pass LSTM to Generate Output

The description of forward pass on LSTM are following steps:

1. Calculate the forget gate value using equation (2).
2. Calculate the input gate value by using equation (3) and equation (4).

3. Calculate the cell state value using equation (5).
4. Calculate the output gate value using equation (6).
5. Calculate the hidden value that becomes the output by using the equation (7).
6. Finally, use equation (10) to calculate loss.

$$E = \frac{1}{n}(y - h)^2 \tag{10}$$

Backward Pass LSTM

The backward pass is the process of going backward from the forward pass. The purpose of the backward pass is to minimize errors by updating the weights on the architecture so that the learning process becomes better. The explain of the backward pass in LSTM has been done in Greff [20], and Maohua [28] studies.

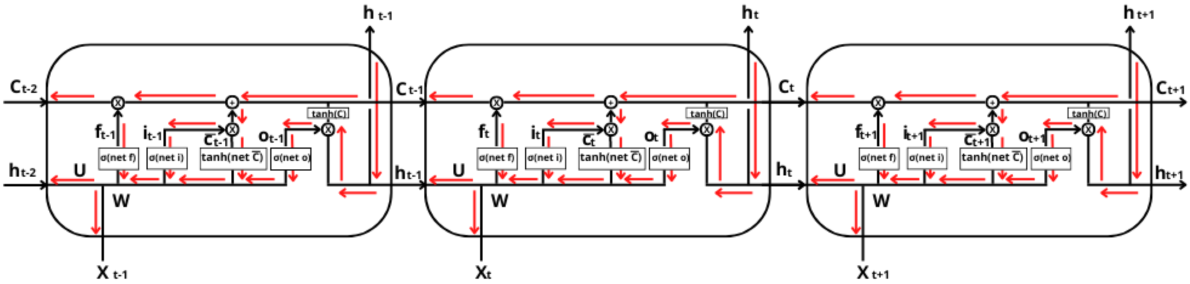


Figure 8. Backward Pass LSTM to Update The Weights

The description of backward pass on LSTM are following steps:

1. Calculating the value of the loss derivative with respect to the input weight of the forget gate, input gate, $\tilde{C}$, and output gate at t = T.

$$\frac{\partial E_T}{\partial W_f} = \frac{\partial \mathrm{E_T}}{\partial h_T}\frac{\partial h_T}{\partial C_T}\frac{\partial \mathrm{C_T}}{\partial f_T}\frac{\partial f_T}{\partial W_f} \tag{11}$$

$$\frac{\partial E_T}{\partial W_i} = \frac{\partial \mathrm{E_T}}{\partial h_T}\frac{\partial h_T}{\partial C_T}\frac{\partial \mathrm{C_T}}{\partial i_T}\frac{\partial i_T}{\partial W_i} \tag{12}$$

$$\frac{\partial E_T}{\partial W_c} = \frac{\partial \mathrm{E_T}}{\partial h_T}\frac{\partial h_T}{\partial C_T}\frac{\partial C_T}{\partial \widetilde{C_T}}\frac{\partial \widetilde{C_T}}{\partial W_c} \tag{13}$$

$$\frac{\partial E_T}{\partial W_o} = \frac{\partial \mathrm{E}}{\partial h_T}\frac{\partial h_T}{\partial C_T}\frac{\partial C_T}{\partial o_T}\frac{\partial o_T}{\partial W_o} \tag{14}$$

2. Calculating the value of the loss derivative with respect to the input weight of the forget gate, input gate, $\tilde{C}$, and output gate at t < T

$$\frac{\partial E_t}{\partial W_f} = \frac{\partial \mathrm{E_T}}{\partial h_T}\frac{\partial h_T}{\partial C_T}\left(\prod_{k=2}^{T}\frac{\partial \mathrm{C_k}}{\partial C_{k-1}}\right)\frac{\partial \mathrm{C_t}}{\partial f_t}\frac{\partial f_t}{\partial W_f} + \cdots + \frac{\partial \mathrm{E_t}}{\partial h_t}\frac{\partial h_t}{\partial C_t}\frac{\partial \mathrm{C_t}}{\partial f_t}\frac{\partial f_t}{\partial W_f} \tag{15}$$

$$\frac{\partial E_t}{\partial W_i} = \frac{\partial \mathrm{E_T}}{\partial h_T}\frac{\partial h_T}{\partial C_T}\left(\prod_{k=2}^{T}\frac{\partial \mathrm{C_k}}{\partial C_{k-1}}\right)\frac{\partial \mathrm{C_t}}{\partial i_t}\frac{\partial i_t}{\partial W_i} + \cdots + \frac{\partial \mathrm{E_t}}{\partial h_t}\frac{\partial h_t}{\partial C_t}\frac{\partial \mathrm{C_t}}{\partial i_t}\frac{\partial i_t}{\partial W_i} \tag{16}$$

$$\frac{\partial E_t}{\partial W_c} = \frac{\partial \mathrm{E_T}}{\partial h_T}\frac{\partial h_T}{\partial C_T}\left(\prod_{k=2}^{T}\frac{\partial \mathrm{C_k}}{\partial C_{k-1}}\right)\frac{\partial \mathrm{C_t}}{\partial \widetilde{C_T}}\frac{\partial \widetilde{C_T}}{\partial W_c} + \cdots + \frac{\partial \mathrm{E_t}}{\partial h_t}\frac{\partial h_t}{\partial C_t}\frac{\partial \mathrm{C_t}}{\partial \widetilde{C_T}}\frac{\partial \widetilde{C_T}}{\partial W_c} \tag{17}$$

$$\frac{\partial E_t}{\partial W_o} = \frac{\partial \mathrm{E_T}}{\partial h_T}\frac{\partial h_T}{\partial C_T}\left(\prod_{k=2}^{T}\frac{\partial \mathrm{C_k}}{\partial C_{k-1}}\right)\frac{\partial \mathrm{C_t}}{\partial o_t}\frac{\partial o_t}{\partial W_o} + \cdots + \frac{\partial \mathrm{E_t}}{\partial h_t}\frac{\partial h_t}{\partial C_t}\frac{\partial \mathrm{C_t}}{\partial o_t}\frac{\partial o_t}{\partial W_o} \tag{18}$$

3. Calculating the value of the loss derivatives with respect to the repetition weight and biases of the forget gate, input gate, $\tilde{C}$, and output gate, is the same as equations (11) to (18), but must adjust the variables to be derived.
4. Adding up the loss derivative values at t = T and t < T
5. Updating input weights, repetition weights, and biases using the SGD and Adam equations

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a technique for solving deep learning problems. The problem is to find the optimal weight by minimizing the resulting error. SGD is a development of Gradient Descent (GD), where SGD only uses one data from $\nabla E_t(t = 1, 2, \dots, n)$ and choose it randomly or stochastic [29].

$$W_{t+1} = W_t - \alpha \nabla E_t \tag{19}$$

At the beginning of the algorithm process it starts by chosen random data. Errors from first process are corrected during repeated process using the gradient rule of the function to be minimized.

Adam

Adam is a method that was introduced in 2015 at research [15]. Just like SGD, Adam is used to updating the weights so that learning architecture is better because the error is minimize. The steps in Adam's algorithm are as follows:

1. Initialization m_0 = 0 (first moment).
2. Initialization n_0 = 0 (second moment).

3. Initialization t = 0 (initialize timestep).
4. Updating the first moment with equation:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{20}$$

5. Updating the second moment with equation:

$$n_t = \beta_2 n_{t-1} + (1 - \beta_2) g_t^2 \tag{21}$$

6. Correcting the bias at the first moment with the equation:

$$\widehat{m} = \frac{m_t}{1 - \beta_1^t} \tag{22}$$

7. Correcting the bias at the second moment with the equation:

$$\hat{n} = \frac{n_t}{1 - \beta_2^t} \tag{23}$$

8. Updating the weights with the equation:

$$W_{t+1} = W_t - \alpha \frac{\widehat{m_t}}{\sqrt{\widehat{n}_t + \epsilon}} \tag{24}$$

There are several parameters in Adam, such a g_t is the gradient value, β_1, and β_2 which values 0.9 and 0.99. The last parameter ϵ which is worth 10^{-8} prevents division by 0.

Root Mean Squared Error (RMSE)

Model evaluation needs to find out how well the results obtained from the model used [30]. Error prediction is an error that occurs between the predicted data and the actual data. There are many methods to find out the error you get, one of which is the Root Mean Squared Error also known as RMSE. The error value of RMSE will be closer to zero if the error is getting smaller, and it is also a sign that the model is getting better [22].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}} \tag{25}$$

the $\hat{y}$ value is the predicted value obtained from the LSTM forward pass process.

RESULTS AND DISCUSSION

The data is on closing prices for BBRI, BBNI, BMRI, and BBTN shares downloaded from Yahoo Finance! from January 2, 2020, to October 7, 2022.

Table 1. Data that used on this research

BBRI	BBNI	BMRI	BBTN
4410	7775	7750	2130
4420	7800	7725	2150
4370	7625	7600	2130
⋮	⋮	⋮	⋮
4080	6725	7100	1750
4110	6750	7025	1730

The normalization process applied to each shares, then forms a time series data pattern, and finally becomes input from the LSTM SGD and LSTM Adam. The LSTM prediction process starts from the forget gate, input gate, cell state, output gate using equations (2) to (7), this process continues to repeat throughout the time series pattern and LSTM produces the loss value in equation (10). Next, to increase the output value as a predicted value, LSTM uses the Backward process. The aim of this process is to find optimal weights in LSTM. LSTM has four components, so the backward LSTM calculation is based on each component. Because LSTM uses time series data, the backward process starts from the latest data range. The equations used by LSTM at t = T are shown by equations (11) to (14) and the equations used by LSTM at t < T are shown by equations (15) to (18). The process of updating weights and biases uses SGD equation (19), and Adam uses equation (24). Iterations will continue to repeat themselves, and the weights reach optimal when the error difference does not produce a significant difference.

Experimented with various learning rates and epochs, and was found that the average minimum error of the LSTM SGD with a learning rate of 0.01, the average minimum error of the LSTM Adam with a learning rate of 0.001. Table 2 shows the error values of the LSTM SGD:

Table 2. Error LSTM SGD with Learning Rate 0.01

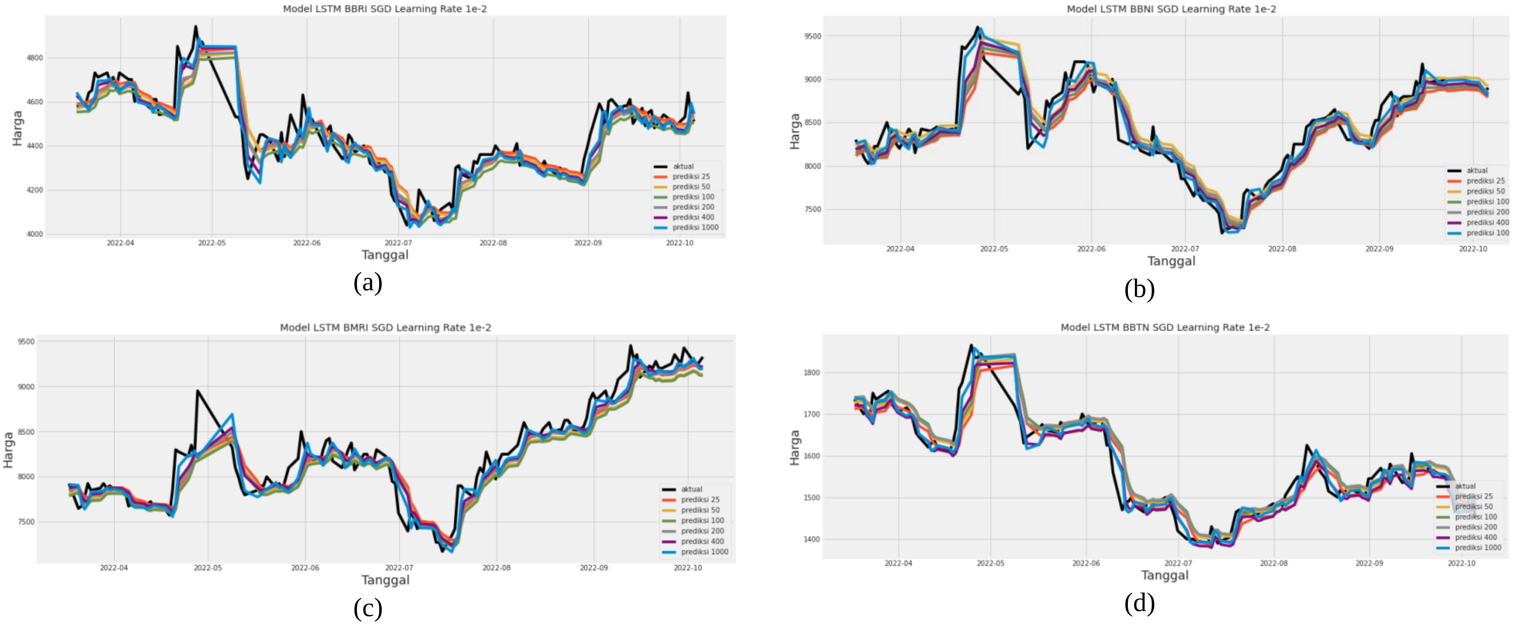
Epoch	BBRI	BBNI	BMRI	BBTN
25	4.20	85.50	54.55	6.20
50	18.43	28.99	98.06	7.79
100	46.30	65.69	120.69	13.17
200	25.23	23.11	62.86	13.77
400	19.17	46.30	46.90	12.61
1000	22.03	16.75	54.33	2.59

Table 3 shows the error values of the LSTM Adam with learning rate 0.001:

Table 3. Error LSTM Adam with Learning Rate 0.001

Epoch	BBRI	BBNI	BMRI	BBTN
25	12.33	22.22	118.08	10.63
50	55.51	9.32	65.41	12.21
100	5.43	59.57	22.87	32.97
200	43.89	75.48	99.01	9.60
400	17.21	22.92	88.73	7.78
1000	2.13	38.04	104.49	19.79

Compare table 2 and table 3, which in this study found that in BBRI, BBNI, and BMRI stocks, LSTM Adam produced a better error, while in BBTN stock, LSTM SGD produced a better error. The number of epochs is also highly considered because it will affect the computation time. The next step is to display the actual and predicted plots of the LSTM SGD.



(a)

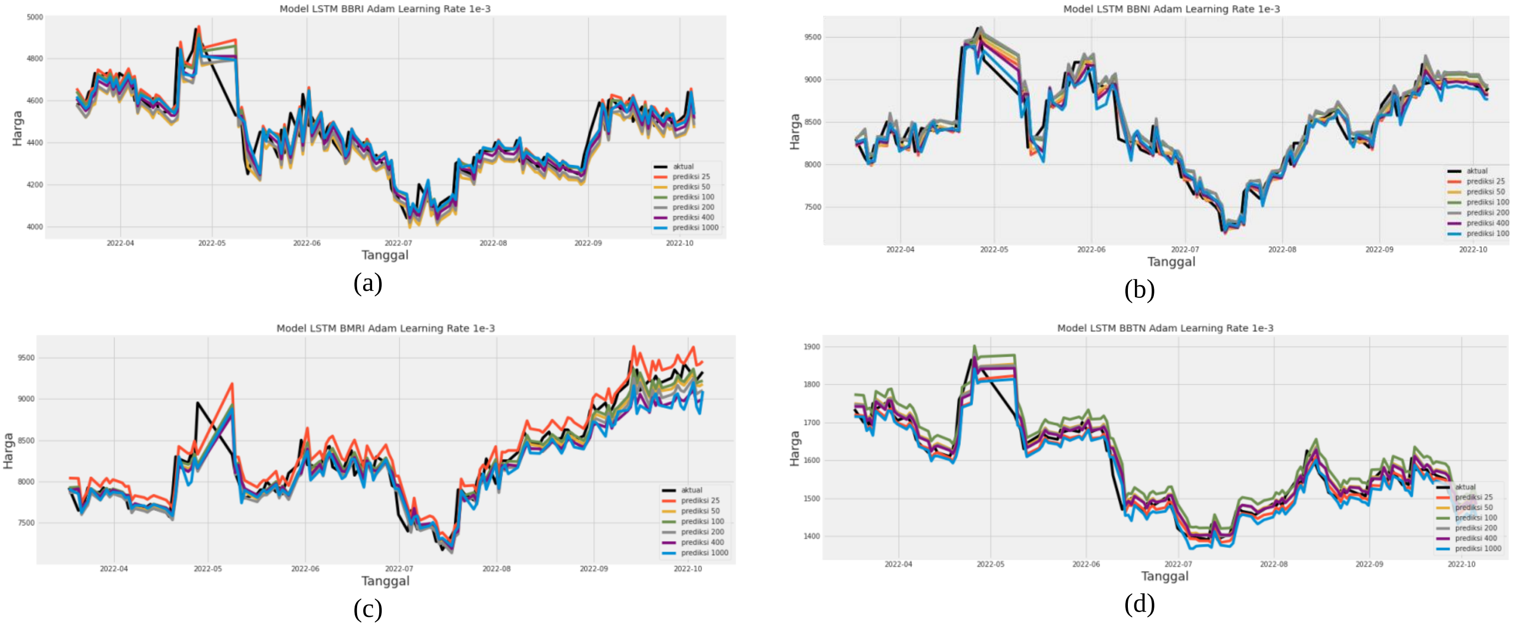
(b)

(c)

(d)

Figure 9. Plot Actual and Predicted Value from LSTM SGD Training, (a) is BBRI, (b) is BBNI, (c) is BMRI, and (d) is BBTN

figure 10 shows the plot actual and predicted value of the LSTM Adam with learning rate 0.001:



(a)

(b)

(c)

(d)

Figure 10. Plot Actual and Predicted Value from LSTM Adam Training, (a) is BBRI, (b) is BBNI, (c) is BMRI, and (d) is BBTN

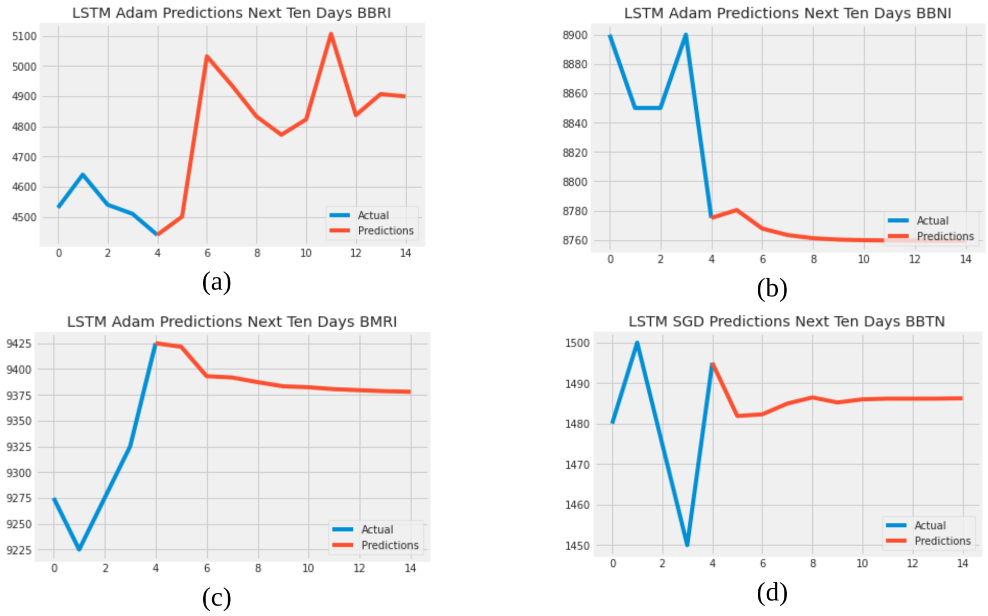
Figure 9 and Figure 10 show the actual comparison with predictions from LSTM SGD and LSTM Adam training, result is that each stock has a different epoch value to produce a prediction value that is close to the actual data and produce the minimum error. LSTM Adam requires 1000 epochs to the minimum error of 2.13 on BBRI shares. LSTM Adam produces a minimum error on BBNI with a score of 9.32 and requires 50 epochs. LSTM Adam with 100 epochs produces a minimum error of 22.87 on BMRI shares, and LSTM SGD produces a minimum error on 1000 epochs with a score of 2.59 on BBTN shares.

LSTM SGD and LSTM Adam have successfully trained. The next step is to try to make predictions for the next ten days using five historical data and using a model that produces the minimumt error such as LSTM Adam on BBRI, BBNI, and BMRI shares, and LSTM SGD on BBTN shares.:

Table 4. Predict The Next Ten Days with LSTM SGD and LSTM Adam

Date	BBRI	BBNI	BMRI	BBTN
03 October 2022 (Actual)	4530	8900	9275	1480
04 October 2022 (Actual)	4640	8850	9225	1500
05 October 2022 (Actual)	4540	8850	9275	1475
06 October 2022 (Actual)	4510	8900	9325	1450
07 October 2022 (Actual)	4440	8775	9425	1495
10 October 2022 (Prediction)	4500.43	8780.38	9421.59	1481.89
11 October 2022 (Prediction)	5032.38	8767.79	9393.06	1482.29
12 October 2022 (Prediction)	4936.38	8763.32	9391.74	1484.95
13 October 2022 (Prediction)	4832.66	8761.20	9387.25	1486.46
14 October 2022 (Prediction)	4771.99	8760.22	9383.24	1485.22
17 October 2022 (Prediction)	4823.25	8759.82	9382.31	1485.99
18 October 2022 (Prediction)	5106.91	8759.65	9380.49	1486.16
19 October 2022 (Prediction)	4837.14	8759.57	9379.46	1486.15
20 October 2022 (Prediction)	4907.18	8759.54	9378.51	1486.16
21 October 2022 (Prediction)	4899.19	8759.53	9377.94	1486.24

figure 11, shows a prediction plot for the next ten days.



(a)

(b)

(c)

(d)

Figure 11. Plot Prediction from, (a) BBRI by LSTM Adam, (b) BBNI by LSTM Adam, (c) BMRI by LSTM Adam, and (d) BBTN by LSTM SGD

plot 11(a) shows stock predictions in an uptrend, plot 11(b) and 11(c) shows stock predictions in an downtrend and stagnant, and plot 11(d) shows stock predictions in an uptrend for the next ten days.

CONCLUSIONS

Based on the results of the study, can be concluded that LSTM SGD and LSTM Adam can be trained well. LSTM Adam with a learning rate of 0.001 produces a smaller error value in training, also succeeded in making predictions for the next ten days using the LSTM SGD and LSTM Adam. The result is that both methods can predict stock prices in the next ten days using five historical data.

Suggestions for further research is to focus on testing various time series data patterns, aims to find out how long the time series data can be used in LSTM to prevent vanishing gradient problems and try to experiment with variants of the LSTM architecture, such as Stacked-LSTM.

REFERENCES

[1] Muchlas and T. Sutikno, “Prediksi harga saham berbasis web dengan sistem inferensi fuzi tsukamoto,” 2007.

[2] S. T. Wahyudi, “The ARIMA model for the Indonesia stock price,” *Int. J. Econ. Manag.*, vol. 11, no. SpecialIssue1, pp. 223–236, 2017.

[3] Z. A. R. HSM, “Analisa Laju Perubahan Harga Saham Lq45 Menggunakan Persamaan Diferensial,” *J. Ris. Akunt. Politala*, vol. 3, no. 2, pp. 60–66, 2020, doi: 10.34128/jra.v3i2.68.

[4] G. L. Shelley, A. Traian, and W. J. Trainor, “Stock market ‘prediction’ models,” *Econ. Bull.*, vol. 40, no. 2, pp. 1548–1556, 2020.

[5] A. Nilsen, “Perbandingan Model RNN , Model LSTM , dan Model GRU dalam Memprediksi Harga Saham-Saham LQ45,” vol. 6, no. 1, pp. 137–147, 2022.

[6] A. Olawoyin and Y. Chen, “Predicting the future with artificial neural network,” *Procedia Comput. Sci.*, vol. 140, pp. 383–392, 2018, doi: 10.1016/j.procs.2018.10.300.

[7] Z. E. Mohamed, “Using the artificial neural networks for prediction and validating solar radiation,” *J. Egypt. Math. Soc.*, vol. 27, no. 1, 2019, doi: 10.1186/s42787-019-0043-8.

[8] M. Abdul Dwiyanto Suyudi, E. C. Djamal, A. Maspupah Jurusan Informatika, and F. Sains dan Informatika Universitas Jenderal Achmad Yani Cimahi, “Prediksi Harga Saham menggunakan Metode Recurrent Neural Network,” *Semin. Nas. Apl. Teknol. Inf.*, pp. 1907–5022, 2019.

[9] W. Wahyu *et al.*, *Recurrent neural network-long short term memory*. 2021.

[10] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, 1997, doi: 10.1162/neco.1997.9.8.1735.

[11] J. Dawani, *Hands-On Mathematics for Deep Learning*. Mumbai.

[12] O. Calin, *Deep Learning Architectures : A Mathematical Approach*. 2020.

[13] J. K. Lubis and I. Kharisudin, “Metode Long Short Term Memory dan Generalized Autoregressive Conditional Heteroscedasticity untuk Pemodelan Data Saham,” *Prism. Pros. Semin. Nas*, vol. 4, pp. 652–658, 2021,

[14] M. Hardt, B. Recht, and Y. Singer, “Train faster, generalize better: Stability of stochastic gradient descent,” in *33rd International Conference on Machine Learning, ICML 2016*, 2016, vol. 3.

[15] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[16] E. Bisong, *Training a Neural Network*. 2019.

[17] J. Brownlee, “Long Short-Term Memory Networks With Python,” *Mach. Learn. Mastery With Python*, vol. 1, no. 1, p. 228, 2017.

[18] T. Jayalakshmi and A. Santhakumaran, “Statistical Normalization and Back Propagationfor Classification,” *Int. J. Comput. Theory Eng.*, vol. 3, no. 1, pp. 89–93, 2011, doi: 10.7763/ijcte.2011.v3.288.

[19] A. Karpathy, J. Johnson, and L. Fei-Fei, “Visualizing and Understanding Recurrent Networks,” pp. 1–12, 2015,

[20] K. Greff, R. K. Srivastava, J. Koutn, and B. R. Steunebrink, “LSTM : A Search Space Odyssey,” pp. 1–12, 2016, doi: 10.1109/TNNLS.2016.2582924.

[21] K. Johan, J. C. Young, and S. Hansun, “LSTM-RNN automotive stock price prediction,” *Int. J. Sci. Technol. Res.*, vol. 8, no. 9, pp. 173–176, 2019.

[22] A. Nurjaman, A. H.-P. 2021, “Long Short-Term Memory (LSTM) untuk Prediksi Harga Saham Pfizer Inc,” *Prosiding.Statistics.Unpad.Ac.Id*, 2021.

[23] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A Critical Review of Recurrent Neural Networks for Sequence Learning,” pp. 1–38, 2015.

[24] C. A. Goodfellow lan, Bengio Yoshua, “Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville - Google Books,” *MIT Press*. 2016.

[25] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. 2008.

[26] I. Sutskever, “Training Recurrent Neural Networks,” 2013.

[27] M. Hagan, H. Demuth, M. Beale, and O. De Jesus, “Neural Network Design, Boston,” *PWS Pub. Co. USA*, 1996.

[28] Z. Maohua, X. Yuan, R. Minsoo, C. Jason, and W. K. Stephen, “Training Long Short-Term Memory With Spar-Sified Stochastic Gradient Descent,” *Iclr*, no. 2, pp. 1–10, 2017.

[29] X. S. Yang, *Optimization techniques and applications with examples*. 2018.

[30] D. Wallach and B. Goffinet, “Mean squared error of prediction as a criterion for evaluating and comparing system models,” *Ecol. Modell.*, vol. 44, no. 3–4, pp. 299–306, 1989, doi: 10.1016/0304-3800(89)90035-5.