JOHME: Journal of Holistic Mathematics Education
Vol 9, No 2 Dec 2025 pages: 213 – 224
DOI: https://dx.doi.org/10.19166/johme.v9i2.10220
E-ISSN: 2598-6759

ANALYSIS OF LONG SHORT – TERM MEMORY (LSTM) PARAMETERS IN PREDICTING IHSG

Daud Padut Aritiran R[1], Lina Cahyadi[2], Ferry Vincentius Ferdinand[3], Kie Van Ivanky Saputra[4], Kathleen Teja[5]

[1]PT Siloam International Hospitals Tbk, Tangerang, BANTEN
[2,3,4,5]Universitas Pelita Harapan, Tangerang, BANTEN

Correspondence Email: lina.cahyadi@uph.edu

ABSTRACT

For investors looking to enhance the value of their financial assets, stock investment is a popular choice. A Long Short-Term Memory (LSTM) model will be used to forecast the movement of the Indonesia Composite Index (IHSG) in the domestic capital market. This research focuses on key parameters of the LSTM model, such as sliding window size, the number of epochs, the learning rate, and the type of optimizer. There are four configurations that were tested. First, the sliding window size was varied while keeping other parameters constant. Second, while maintaining the other parameters, the number of epochs was modified. Third, while keeping the remaining parameters unchanged, the learning rate was adjusted. Lastly, while holding the other parameters constant, different optimizers were tested. The dataset is divided into two periods , such as : pre-pandemic and during the pandemic. The dataset is segmented into training and testing sets for every period. During the pre-pandemic period, the best-performing parameters included a sliding window size of 20, training over 40 epochs with a learning rate of 0.001, and the Adam optimizer, resulting in an RMSE of 7.2218. The best results during the pandemic period were obtained with parameters consisting of a sliding window size of 5, 10 epochs, a learning rate of 0.001, and the Adam optimizer, resulting in an RMSE of 1.727. These parameter combinations demonstrated the highest predictive performance for IHSG.

Keywords: IHSG, LSTM, prediction, parameter, stock

ABSTRAK

Untuk para investor yang ingin meningkatkan nilai aset keuangan mereka, investasi saham adalah pilihan populer. Sebuah model *Long Short-Term Memory* (LSTM) akan digunakan untuk memprediksi harga Indeks Harga Saham Gabungan (IHSG) di pasar modal Indonesia. Penelitian ini memfokuskan pada parameter kunci dari model LSTM, seperti ukuran *sliding windows*, jumlah *epoch*, *learning rate*, dan jenis *optimizer*. Ada empat konfigurasi yang diuji. Pertama, ukuran *sliding windows* divariasikan sementara parameter lainnya tetap konstan. Kedua, jumlah *epoch* dimodifikasi dengan tetap mempertahankan parameter lainnya. Ketiga, *learning rate* divariasikan dengan parameter lainnya tetap tidak berubah. Terakhir, berbagai *optimizer* diuji dengan parameter lainnya tetap konstan. Dataset ini dibagi menjadi dua periode, yaitu sebelum pandemi dan selama pandemi. Data dibagi menjadi set pelatihan dan pengujian untuk setiap periode. Parameter optimal untuk periode sebelum

Received: 31/07/2025 Revised: 04/11/2025 Published: 10/12/2025

pandemi adalah ukuran *sliding windows* 20, 40 *epoch*, learning rate 0,001, dan *optimizer Adam*, menghasilkan *Root Mean Squared Error* (RMSE) sebesar 7,2218. Selama pandemi, parameter terbaik adalah ukuran *sliding windows* 5, 10 *epoch*, *learning rate* 0,001, dan *optimizer* Adam, dengan RMSE sebesar 1,727. Kombinasi parameter ini menunjukkan kinerja prediksi tertinggi untuk IHSG.

Kata Kunci: IHSG, LSTM, prediksi, parameter, saham.

INTRODUCTION

The increasing interest in investment among millennials had a significant effect on the Indonesian financial market. The active participation of the millennial generation has significantly increased market liquidity and stability. The involvement of individuals aged 31-40 has led to the development of more innovative investment products tailored to the needs of younger investors. This trend also brings new challenges for the investment industry. To attract and keep millennial investors, securities firms and investment managers must adapt their strategies by focusing on the development of digital technologies and services that meet the high expectations of a digitally connected generation.

In addressing many challenges across multiple factors, the rapid advancement of technology has become important. The increase in computational power and the accessibility of data have unlocked new potential for predictive analytics. Deep learning algorithms and traditional statistical methods have emerged as effective tools to help investors navigate complex decision-making processes. To utilize predictive analytics tools to evaluate historical stock performance, forecast future trends, and make more informed investment decisions, technological advancements are needed.

Long Short-Term Memory (LSTM) is a form of artificial neural network specifically developed to handle and forecast time-series data. It overcomes the vanishing gradient issue often found in traditional neural networks, enabling the model to recognize historical patterns and deliver more accurate predictions. This study aims to analyze key parameters in an LSTM-based stock price prediction model using Indonesia Composite Index (IHSG) price data from January 2016 to December 2019 and from January 2020 to December 2023. After the LSTM model is established, an error analysis will be conducted to evaluate its performance under various parameter settings. This approach will identify the optimal LSTM configuration for IHSG stock price prediction.

LITERATURE REVIEW

Research conducted by I Ketut Agung Enriko, Fikri Nizar Gustiyana, and Rahmat Hardian Putra, titled *"Comparison of Optimization Results in Stock Price Prediction of PT Telkom Indonesia Using the Long Short-Term Memory Algorithm,"* indicates that the LSTM algorithm achieves strong predictive accuracy, as reflected by the Mean Absolute Percentage Error (MAPE) values and the model outputs obtained from different epoch settings. Using the Adam optimizer, it was observed that increasing the number of epochs resulted in lower loss values, thereby improving the stock price prediction accuracy. The Adam optimization model achieved the highest accuracy, reaching 98.45% Enriko, I. K. A., Gustiyana, F. N., & Putra, R. H.].

Research conducted by Akhmad Yusuf, titled "Prediction of the Indonesia Composite Index (IHSG) Using Long Short-Term Memory," shows that the LSTM parameters used—specifically a batch size of 25 and 50 epochs—produced the best model performance, achieving a lower RMSE of 6.2335, with a predicted value of 6765.5103 compared to the actual value of 6807.50 [Yusuf, A.].

Research conducted by Peter T. Yamak, Li Yujian, and Pius K. Gadosey, titled "A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting," shows that the ARIMA (1,1,0) model achieved the highest accuracy, with a MAPE of 0.0276 and an RMSE of 302.56. In comparison, the GRU model obtained a MAPE of 0.0397 and an RMSE of 381.34, while the LSTM model produced a MAPE of 0.0680 and an RMSE of 603.68 using the following parameters: 400 epochs, a batch size of 170, one dense layer, an input shape of 1, and the Adam optimizer [Yamak, P. T., Yujian, L., & Gadosey, P. K.].

Research conducted by Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin, titled "A Comparison of ARIMA and LSTM in Forecasting Time Series," shows that there is no evidence suggesting that training the network on the same dataset multiple times improves prediction accuracy. In several cases, performance even deteriorated, indicating that the model became overfitted. As a key takeaway, the study concludes that setting the number of epochs to one yields a reasonably accurate prediction model, thereby eliminating the need for additional training on the same data [Siami-Namini, S., Tavakoli, N., & Siami Namin, A.]

The research conducted by Adhitio Satyo Bayangkari Karno, titled "Time Series Data Analysis Using LSTM (Long Short-Term Memory) and Autoregressive Integrated Moving Average (ARIMA) in the Python Programming Language," aims to predict time series data using two approaches: the traditional statistical method, Autoregressive Integrated Moving Average (ARIMA), and the more recent machine learning technique, Long Short-Term Memory (LSTM). Prior to modeling, data cleaning and optimization were performed. The optimization process involved transformations to remove trends and variability, resulting in seven combinations derived from Log, Moving Average (MA), Exponential Weighted Moving Average (EWMA), and Differencing (Diff). These transformations were applied to both the ARIMA(2,1,1) and LSTM models, producing a total of 14 predictions (7 from ARIMA and 7 from LSTM). Among these, the lowest RMSE obtained using ARIMA was 0.02, while the lowest RMSE achieved with LSTM was 0.01. These findings demonstrate that the LSTM model, when applied to Telkom Indonesia (TLKM) stock data, provides higher prediction accuracy compared to the ARIMA model [Karno, A. S. B.].

RESEARCH METHODOLOGY

The procedure for comparing the parameters of the Long Short-Term Memory (LSTM) model is illustrated in Figure 1. The research steps shown in Figure 1 will be explained in more detail, starting with part A Processing and Data Collection until part D. LSTM Model Evaluation.

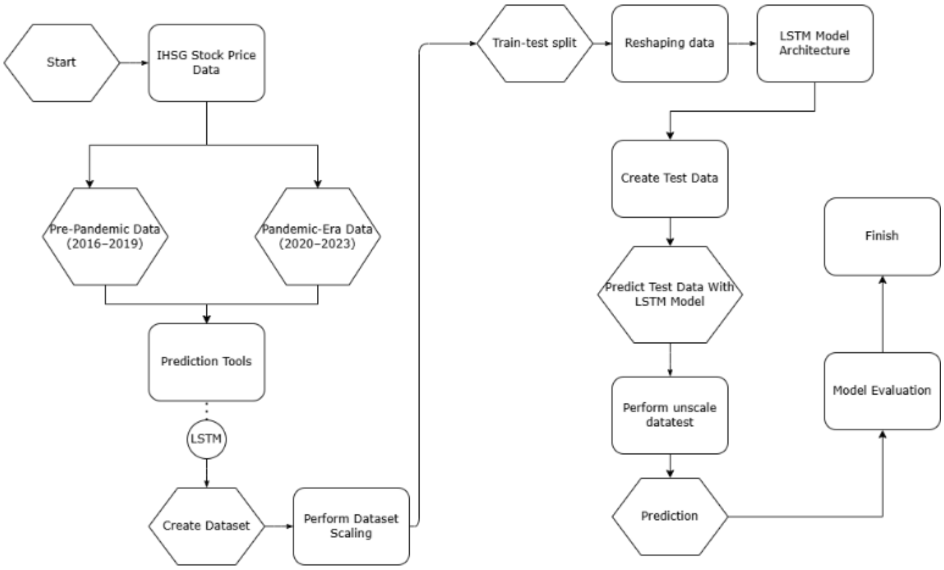

Figure 1. Flowchart work steps.

Processing and data collection

This study will use a dataset obtained from finance.yahoo.com. The data include stock price projections for two periods: January 4, 2016, to December 30, 2019, and January 2, 2020, to December 29, 2023. These data are used to enable a comparison of the parameters obtained during the periods before and after the COVID-19 pandemic. The dataset contains several columns, such as Date, Open, Close, High, Low, Adj Close, and Volume. However, this study only uses the Date and Close columns. The Date column provides dates at daily intervals, while the Close column contains the stock closing prices.

Overall, this dataset contains 1,019 and 974 entries for the two different time periods, respectively. The data is partitioned into training (80%) and testing (20%) sets. Thirty random divisions of the train and test datasets will be generated using Python, the performance of the study regarding accuracy and reliability is examined. The code used in this study can be accessed at https://github.com/ArtDz12/StockPricePredLSTM. The plots of the dataset are presented in Figures 2 and 3.

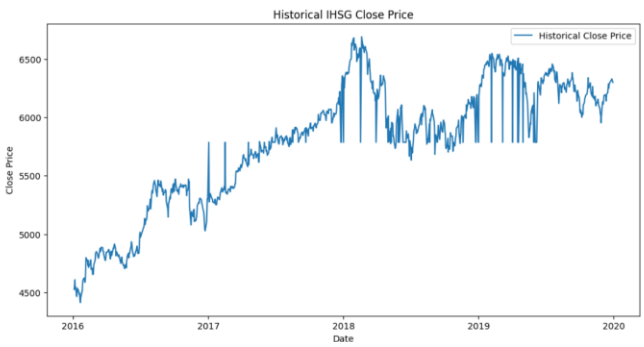

Figure 2. IHSG Stock price on year 2016-2019

Figure 3. IHSG stock price on 2020-2023

LSTM Model

The Long Short-Term Memory (LSTM) model is a specialized form of Recurrent Neural Network (RNN) created to address the shortcomings of conventional RNNs, especially the vanishing gradient issue. LSTM networks, by capturing long-term dependencies in sequences, are highly effective for applications such as time-series forecasting, natural language processing (NLP), and other tasks involving sequential data. A key limitation of standard recurrent neural networks (RNNs) is the vanishing gradient problem, which restricts their capacity to maintain long-term dependencies in sequential data. To overcome this issue, long short-term memory (LSTM) networks were developed with specialized memory cells that allow the network to retain information over longer periods. As a result, LSTMs are able to effectively learn from previous time steps, making them well-suited for forecasting tasks that depend on historical patterns. [Huang. Y and Yan. E].

The LSTM cell is mathematically structured around three crucial gates: forget, input, and output, which manage the transmission of information through the cell state. Each gate performs a specific function, controlling how information is updated, retained, or discarded over time.

The operations of an LSTM cell can be described by the following equations [Staudemeyer, R. C., & Morris, E. R.]:

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{C}_t &= \tanh(W_C x_t + U_C h_{t-1} + b_C) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
h_t &= o_t \odot \tanh(C_t)
\end{aligned}
$$

Where:

- x_t represents the input at time step t,
- h_{t-1} is the previous hidden state,
- C_{t-1} is the previous cell state,
- W and U denote the weight matrices for the input and hidden state, respectively,
- b represents the bias vector,
- σ is the sigmoid activation function,
- $\tanh$ is the hyperbolic tangent activation function, and
- $\odot$ denotes element-wise multiplication.

Through the forget gate f_t, the network determines which portions of the previous cell state to eliminate. The input gate $i_t$and candidate cell state $\tilde{C}_t$control how much new information is added to the cell state. The cell state update combines the retained and new information to form the updated memory C_t. Finally, the output gate $o_t$determines the portion of the memory that will influence the hidden state h_t, which serves as the output of the LSTM cell for the current time step.

Through these mechanisms, the LSTM model can effectively learn temporal dependencies and patterns in data over long time horizons. Therefore, it finds broad application in areas such as time-series prediction, speech recognition, and analysis of trends before and after major events such as the COVID-19 pandemic.

The first step in implementing the LSTM model is normalizing the dataset. Normalization scales all input values to a common range, typically between 0 and 1, ensuring that the model trains efficiently and that each feature contributes proportionally. This process is usually performed using the Min–Max normalization formula, which improves convergence speed and model accuracy during training. The data is then divided into 80% training data and 20% testing data. The LSTM model will be trained using the hyperparameters listed in Table 1.

This architecture is a recurrent model consisting of three LSTM (Long Short-Term Memory) layers designed for processing sequential data, such as time series. Each LSTM layer contains 50 units and uses return_sequences=True in the first layer to provide sequential output for each time step. A dropout layer with a rate of 0.2 is added after each LSTM layer to reduce overfitting. The model concludes with a Dense layer containing a single unit for prediction. With this structure, the model is capable of making predictions based on the sequential information learned during training.

The data will be divided into train and test datasets, as shown in Table 2. In the development of the LSTM model, four networks of analysis will be conducted, each involving a different combination of treatments for every test. Each network of analysis focuses on evaluating the impact of a single key parameter on model performance. First, the analysis will be performed on the Sliding Window parameter, as it is a crucial factor in the model's performance.

Table 1. LSTM Model Architecture

Parameter	Value
Epoch	10,20,30,40, and 50
Batch size	20
Learning rate	0.001, 0.01, and 0.1
Dense layer	1
Dropout	0.2
Optimizer	ADAM, AdaGrad, SGD, and RMSProp
Dataset Composition	80 : 20
Sliding Window	5,10.15, and 20

Table 2. Training and Testing Data Sum

Period	Comparison	Training data sum	Testing data sum
Jan 4, 2016 – Dec 30, 2019	80 : 20	815	204
Jan 2, 2020 - Dec 29, 2023	80 : 20	779	195

Table 3. Fragment of the Series of Analysis of Windows Sliding Parameters, Epoch, Learning Rate and Optimizer

Analysis 1	Analysis 2	Analysis 3	…	Analysis 240
Sliding window = 5	Sliding window = 10	Sliding window = 15		Sliding window = 20
Epoch = 10	Epoch = 10	Epoch = 10		Epoch = 50
Learning rate = 0.001	Learning rate = 0.001	Learning rate = 0.001		Learning rate = 0.1
Optimizer = AdaGrad	Optimizer = AdaGrad	Optimizer = AdaGrad		Optimizer = SGD

First, the analysis of the Sliding Window parameter will be conducted, varying the Sliding Window while keeping all other parameters constant. Next, the Epoch parameter will be varied, with the remaining parameters held constant. Then, the Learning Rate will be adjusted, while other parameters remain unchanged. Finally, the Optimizer will be varied, keeping all other parameters the same. The results of these variations are presented in Table 3, comprising a total of 240 combinations. The main objective of this analysis is to understand how each parameter influences the LSTM model's ability to predict sequential data, enabling the selection of an optimal combination of parameters for accurate prediction.

Dataset Denormalization

After the prediction step on the testing data, the predicted values, which are still in the normalized scale, will be converted back to the original scale to allow comparison with the actual stock prices. This denormalization is performed using scaler.inverse_transform(). Before evaluating the prediction results, the dataset will be denormalized to restore the values that were scaled during the initial normalization step. The denormalization formula is presented in Formula 2 [Wiranda & Sadikin].

$$D = D' \, (\text{maximum} - \text{minimum}) + \text{minimum}$$

Where $D$is the denormalized prediction result, $D'$is the normalized value, maximum is the maximum value of the actual stock data, and minimum is the minimum value of the actual stock data.

LSTM Model Evaluation

In the LSTM model evaluation step, the Root Mean Square Error (RMSE) is used as the primary metric. Stock price predictions are prone to outliers or values that deviate significantly from the mean. RMSE is particularly suitable in this context because it penalizes larger errors more heavily, providing a more accurate representation of the model's predictive performance. Furthermore, RMSE is expressed in the same units as the original data, which makes it easier to interpret the magnitude of prediction errors in the context of real stock prices [L. Zhang et al.].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y(i) - y'(i))^2}. \tag{3}$$

with y(i) representing actual value on t, y ' (i) is the prediction value for the same period, and n is prediction number.

Based on evaluation results LSTM model can be used to get conclusion for this study. Eventually, can be choose LSTM model with the best combination parameter which is better to predict stock price based on evaluation value RMSE which is smaller.

RESULT AND DISCUSSION

The first step in processing the stock data is to divide the datasets, which contain 1,019 and 974 entries, into two parts: train and test datasets, using an 80:20 ratio, as shown in Table 3. Next, the training data is used to develop the model based on the parameters listed in Table 1. Using the model trained on the training data, the testing data is then employed to evaluate the model's performance, which is measured using RMSE. The evaluation results for the Sliding Window, Epoch, Learning Rate, and Optimizer parameters were used to select the optimal configuration for the IHSG stock for the periods 2016–2019 and 2020–2023, as presented in Table 4.

Table 4. Best result evaluation from parameters which in 2 periods

Period	Sliding Window	Epoch	Learning Rate	Optimizer	RMSE
4 Jan 2016 - 30 Dec 2019	20	40	0.001	ADAM	7.221810
2 Jan 2020 - 29 Dec 2023	5	10	0.001	ADAM	1.727176

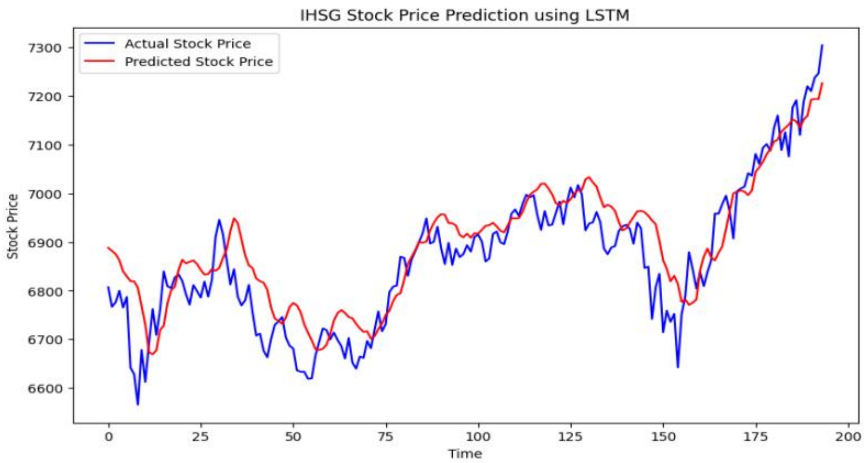

Figure 4. Evaluation result plot from the combination parameter IHSG stock year 2016-2019

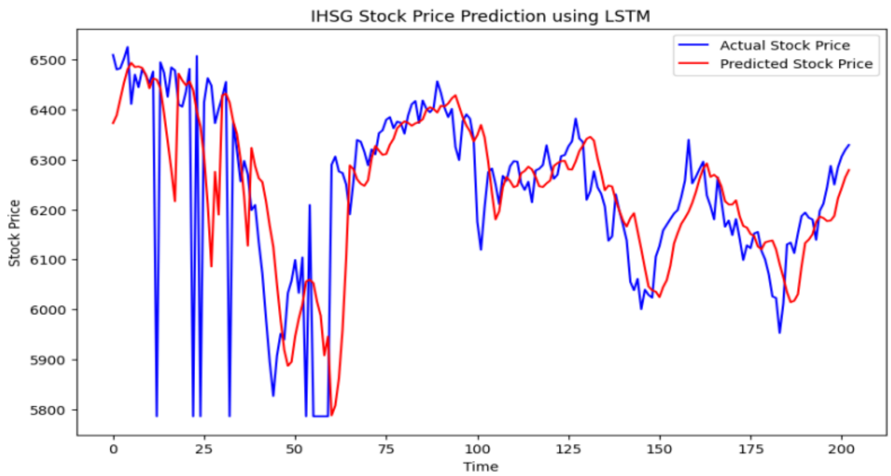

Figure 5. Evaluation result plot from the combination parameter IHSG stock year 2020-2023

The best results from Table 4 are illustrated in Figures 4 and 5. The graphs show that the IHSG predictions for 2016–2019 (before the pandemic) and 2020–2023 (during the pandemic), using the optimal parameter combination with the lowest RMSE, closely match the actual stock prices.

CONCLUSION

The results of the evaluation suggest that the performance of the LSTM model in predicting the IHSG is strongly affected by the Sliding Window, Epoch, Learning Rate, and Optimizer parameters. For the period 2016–2019 (before the pandemic), the best configuration was a Sliding Window of 5, Epoch of 10, Learning Rate of 0.001, and the Adam optimizer, which resulted in an RMSE value of 7.2218. For the period 2020–2023 (during the pandemic), the optimal configuration was a Sliding Window of 20, Epoch of 40, Learning Rate of 0.001, and the Adam optimizer, producing a lower RMSE value of 1.7272, indicating better prediction accuracy during this period.

REFERENCES

Aggarwal, C. (2018). *Neural networks and deep learning.* Springer. https://doi.org/10.1007/978-3-031-29642-0_13

Camuñas-Mesa, L. A., Linares-Barranco, B., & Serrano-Gotarredona, T. (2019). Neuromorphic spiking neural networks and their memristor-CMOS hardware implementations. *Materials, 12*(17), 1-28. https://doi.org/10.3390/ma12172745

Chiang, T. C. (2020). US policy uncertainty and stock returns: Evidence in the US and its spillovers to the European Union, China, and Japan. *The Journal of Risk Finance, 21*(5), 621–657. https://doi.org/10.1108/jrf-10-2019-0190

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research, 12*(7), 2121–2159. Retrieved from https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf

Enriko, I. K. A., Gustiyana, F. N., & Putra, R. H. (2023). Komparasi hasil optimasi pada prediksi harga saham PT Telkom Indonesia menggunakan algoritma long short term memory. *Jurnal Media Informatika Budidarma, 7*(2), 659–667. https://doi.org/10.30865/mib.v7i2.5822

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press.

Hua, Y., Zhao, Z., Liu, Z., Chen, X., Li, R., & Zhang, H. (2018). Traffic prediction based on random connectivity in deep learning with long short-term memory. *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 1–6. https://doi.org/10.1109/vtcfall.2018.8690851

Huang, Y., & Yan, E. (2023) Economic recession forecasts using machine learning models based on the evidence from the covid-19 pandemic. *Modern Economy*, *14*, 899-922. https://doi.org/10.4236/me.2023.147049

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice.* Retrieved from https://robjhyndman.com/uwafiles/fpp-notes.pdf

Hull, J. C. (2009). *Options, futures and other derivatives.* Upper Saddle River, NJ: Prentice Hall.

Ismailsyah, S. E. (2020). Analisis pengaruh free float saham-saham first liner, second liner, dan third liner terhadap likuiditas saham. *Jurnal Ilmiah Mahasiswa FEB, 9*(1), 1-28. Retrieved from https://jimfeb.ub.ac.id/index.php/jimfeb/article/view/7067

Karno, A. S. B. (2020). Analisis data time series menggunakan LSTM (Long short term memory) dan ARIMA (autocorrelation integrated moving average) dalam bahasa Python. *Ultima InfoSys: Jurnal Ilmu Sistem Informasi, 11*(1), 1–7. https://doi.org/10.31937/si.v9i1.1223

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *ICLR 2015*, 1-15. Retrieved from https://arxiv.org/pdf/1412.6980

Li, A. W., & Bastos, G. S. (2020). Stock market forecasting using deep learning and technical analysis: A systematic review. *IEEE Access, 8*, 185232–185242. https://doi.org/10.1109/access.2020.3030226

Mustapha, A., Mohamed, L., & Ali, K. (2021). Comparative study of optimization techniques in deep learning: Application in the ophthalmology field. *Journal of Physics: Conference Series, 1743*, 1-13. https://doi.org/10.1088/1742-6596/1743/1/012002

Noviando, E. S., Ervianto, E., & Yasri, I. (2016). Study on the application of ANN (Artificial neural network) to eliminate harmonics in the main computer center building [Doctoral dissertation]. Retrieved from https://media.neliti.com/media/publications/184148-ID-studi-penerapan-ann-artificial-neural-ne.pdf

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), 1-15. https://doi.org/10.1371/journal.pone.0227222

Rasamoelina, A. D., Adjalila, F., & Sincák, P. (2020). A review of activation function for artificial neural network. *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 281–286. https://doi.org/10.1109/sami48414.2020.9108717

Setiawan, D. (2018). The impact of information and communication technology development on culture. *Jurnal Simbolika: Research and Learning in Communication Study (E-Journal), 4*(1), 62–72. https://doi.org/10.31289/simbollika.v4i1.1474

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929–1958. Retrieved from https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

Staudemeyer, R. C., & Morris, E. R. (2019). *Understanding LSTM – A tutorial into long short-term memory recurrent neural networks.* Retrieved from https://arxiv.org/abs/1909.09586

Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications, 40*(14), 5501–5506. https://doi.org/10.1016/j.eswa.2013.04.013

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review, 53*(8), 5929–5955. https://link.springer.com/article/10.1007/s10462-020-09838-1

Vanstone, B., & Finnie, G. (2009). An empirical methodology for developing stock market trading systems using artificial neural networks. *Expert Systems with Applications, 36*(3), 6668–6680. https://doi.org/10.1016/j.eswa.2008.08.019

Wiranda, L., & Sadikin, M. (2019). Application of long short-term memory on time series data to predict product sales of PT Metiska Farma. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI, 8*(3), 184–196. Retrieved from https://www.neliti.com/id/publications/407800/penerapan-long-short-term-memory-pada-data-time-series-untuk-memprediksi-penjual

Yamak, P. T., Yujian, L., & Gadosey, P. K. (2019). A comparison between ARIMA, LSTM, and GRU for time series forecasting. *Proceedings of the 2019 2nd International Conference* *on Algorithms, Computing and Artificial Intelligence*, 49–55. https://doi.org/10.1145/3377713.3377722

Yusuf, A. (2022). Prediksi indeks harga saham gabungan (IHSG) menggunakan long short-term memory. EPSILON: Jurnal Matematika Murni dan Terapan, 15(2), 124-132. https://doi.org/10.20527/epsilon.v15i2.5026

Zamanlooy, B., & Mirhassani, M. (2014). Efficient VLSI implementation of neural networks with hyperbolic tangent activation function. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 22*(1), 39–48. https://doi.org/10.1109/tvlsi.2012.2232321

Zhang, L., Wen, J., Li, Y., Chen, J., Ye, Y., Fu, Y., & Livingood, W. (2021). A review of machine learning in building load prediction. *Applied Energy, 285*, 1-22. Retrieved from https://www.sciencedirect.com/science/article/pii/S0306261921000209