Daily Forecasting Trend Jakarta Composite Index (JCI) Using Multivariate Long Short-Term Memory

Muhammad Mauludin[a], Rodiah[b]

[a] mauludinmhd@gmail.com
[b]rodiah@staff.gunadarma.ac.id
[a]Magister Manajemen Universitas Gunadarma
[b]Departemen Teknik Informatika Universitas Gunadarma

[a,b]Jl. Margonda Raya 100, Depok 16424, Jawa Barat, Indonesia

Abstract

The need to be able to predict stock price movements is one of the problems that are difficult for investors to solve. Forecasting price movements serve as a signal to buy and sell. Forecasting stock price movements can provide a reference for making investment decisions. This study aims to predict the daily trend of JCI. The datasets used in this research are JCI, NASDAQ, and NYSE, ranging from 21 years (01/01/2000 to 31/12/2021). The features used as input are the opening prices of the JCI, NASDAQ, and NYSE. The amount of data used is 5211 lines. The deep learning method used is multivariate LSTM. The optimization of the model used is Adam. There are 4 LSTM models made using loss metrics MSE and MAE, using two epochs of 100 and 300 epochs. The results showed that the 4 LSTM models could predict the daily trend of the JCI. The most optimum model is the model made using MSE with 300 epochs.

Keywords: LSTM, MSE, MAE, JCI, NASDAQ, NYSE

1. Introductions

Jakarta Composite Index (JCI) is one index used to measure the performance of all stocks listed on the Indonesia Stock Exchange (IDX). The up and down of the JCI can signal Indonesia's economic situation. JCI experienced a decline indicating the existence of one or several stocks whose value fell due to a sell-off from investors. The need to predict stock price movements is one of the problems that investors find difficult to solve, forecasting price movements with the function of signals to buy and sell. Forecasting stock price movements can provide us with a reference for making decisions in investing. One of the methods that can be done to maximize stock prices is using LSTM (Long Short-Term Memory).

2. Literature Review

2.1. Related Studies

Research on the forecasting of stock movements has been carried out by various. Research A. Jayanth Balaji, D. S. Harish Ram, and Binoy B. Nair (2018) using deep learning models Long Short Time Memory (LSTM), Gated Recurring Unit (GRU), Convolutional Neural Networks (CNN), and Extreme Learning Machines (ELM). The model used in the study can produce accurate stock price predictions with different levels of accuracy for

each model. Khaled A. Althelaya, El-Sayed M. El-Alfy, and Salahadin Mohammed (2018), with comparison using the LSTM and GRU models, the result LSTM can demonstrate the highest predictive performance for the near and long term. Different models were also studied by Can Yang, Junjie Zhai, and Guihua Tao (2020), using a combination of CNN and LSTM models to produce a model framework that can beat state-of-the-art models in predicting stock price movements.

Another LSTM deep learning program was also carried out by Ahmad Ashril Rizal and Siti Soraya (2018). The results of the three LST models carried out, namely LSTM regression, LSTM with sliding windows, and LSTM with time steps, no model provides optimal training and testing results simultaneously. Soffa Zahara, Sugianto, and M. Bahril Ilmiddafiq (2019), using seven optimization algorithms showed that Nesterov Adam (Adam) has the highest level of accuracy compared to other algorithms. Both research by Ahmad Fauzi (2019) and by Kevin Johan, Julio C. Young, and Seng Hansun (2019) produced accurate models. Still, there were differences in the number of epochs and the number of layers used. Zineb Lanbouri, and Said Achchab (2020), use LSTM to predict a short period with good performance results for the next 1 minute.

Adhitio Satyo Bayangkari Karno (2020) used high price as input and timestep for seven days, with quite good results. Sidra Mehtab, Jaydip Sen, and Abhishek Dutta (2020) both use univariate and multivariate models with more accurate univariate model results for weekly forecasting. Adil Moghar and Mhamed Hamiche (2020), the model's outcomes were able to trace the evolution of opening prices for GOOGL and NKE with different epochs for each asset. Rahmadi Yotenka and Fazano Fikri El Huda (2020) used three research objects (PT Sawit Sumbermas Sarana Tbk, PT PP London Sumatra Indonesia Tbk, PT Salim Ivomas Pratama Tbk), each using three optimizers, namely Adam, Adamx, and RMSprop, the best results by using a different number of hidden neurons for each optimizer, namely RMSProp with 70 hidden neurons, optimizer Adam with 80 hidden neurons 80, and optimizer Adamax with the 100 of hidden neurons.

Based on the differences in the studies above, the authors use LSTM using a multivariate model with adam optimizer as their research method. Research Dicky Bery, and Saparila Worokinasih (2018) using multiple regression, the results are factors that affect JCI are NASDAQ Composite Index, NYSE Composite Index, SSE Composite Index, TOPIX Composite Index, FTSE All Share Index, this research is the basis for the selection of feature used for the multivariate LSTM model.

In this research, a multivariate time series analysis was carried out to forecasting the trend movement of the composite stock price index using LSTM. The datasets of this study are JCI, NYSE, and NASDAQ for 21 years, optimizers using adam, with the application of deep learning methods. The expected result in this study is that the LSTM multivariate model can forecast the daily trend movement of JCI.

2.2. Long Short-Term Memory (LSTM)

LSTMs are RNNs whose main objective is to overcome the shortcomings of the vanishing gradient and exploding gradient problems. The architecture is built so that they remember data and information for a long period of time. LSTMs were designed to overcome the limitation of the vanishing and exploding gradient problems. LSTM networks are a special kind of RNN that are capable of learning long-term dependencies. They are designed to avoid the long-term dependency problem; being able to remember information for long intervals of time is how they are wired (Matthew Moocarme, Mahla Abdolahnejad, and Ritesh Bhagwat, 2020)

2.3. Mean Absolute Error (MAE) & Mean Squared Error (MSE)

The mean absolute error (MAE) is an evaluation metric for regression models that measures the absolute distance between your predictions and the ground truth. The absolute distance is the distance regardless of the positive or negative sign. The mean squared error (MSE) is computed by taking the squares of the differences between the ground truths and the predictions, summing them, and then dividing by the number of observations (Anthony So et al, 2020).

2.4. Composite Index

1) Jakarta Composite Index (JCI)
   All equities listed on the regular board of the Indonesia Stock Exchange are included in the Jakarta Stock Price Index, which is a modified capitalization-weighted index. The base index value used in the development was 100 as of August 10, 1982.
2) NASDAQ Composite Index
   An index of over 3,700 stocks listed on the Nasdaq stock exchange, the Nasdaq Composite Index is weighted by market capitalization. The Nasdaq Composite Index is a popular measure of the overall health of the financial markets that is significantly skewed toward the vital technology sector.
3) NYSE Composite Index
   The New York Stock Exchange's NYSE Composite Index tracks the performance of all common stocks listed there, including tracking stocks, real estate investment trusts, and American Depositary Receipts issued by overseas corporations. The free-float market capitalization of the index's component companies is used to compute the weights of those companies. The index is calculated according to price and total return, which also takes into account dividends.

3. Methodology

3.1. Model Framework

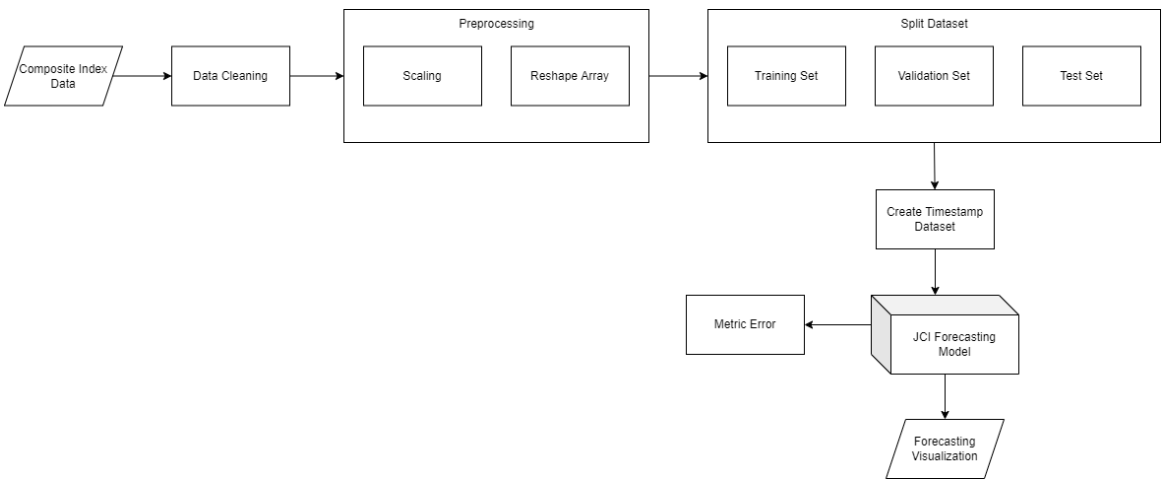


Figure 1: Framework Model Forecasting JCI

The study utilized the following framework design, which follows several steps that are seen in Figure 1.

3.2. Composite Index Data

The Stock Index dataset is the initial stage that is the process of collecting data. The data collected are JCI, NASDAQ, and NYSE, with a span of 21 years from the period 01/01/2000 to 31/12/2021. The number of existing data is 5676-row entries. In each index, there is a difference between the number of non-null data and the total data entries. Non-null data means that the entries are filled with non-empty or null data.

Table 1: Dataframe Info

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5676 entries, 2000-01-03 to 2021-12-30
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   (Adj Close, ^IXIC)  5535 non-null   float64
 1   (Adj Close, ^JKSE)  5352 non-null   float64
 2   (Adj Close, ^NYA)   5535 non-null   float64
 3   (Close, ^IXIC)      5535 non-null   float64
 4   (Close, ^JKSE)      5352 non-null   float64
 5   (Close, ^NYA)       5535 non-null   float64
 6   (High, ^IXIC)       5535 non-null   float64
 7   (High, ^JKSE)       5352 non-null   float64
 8   (High, ^NYA)        5535 non-null   float64
 9   (Low, ^IXIC)        5535 non-null   float64
 10  (Low, ^JKSE)        5352 non-null   float64
 11  (Low, ^NYA)         5535 non-null   float64
 12  (Open, ^IXIC)       5535 non-null   float64
 13  (Open, ^JKSE)       5352 non-null   float64
 14  (Open, ^NYA)        5535 non-null   float64
 15  (Volume, ^IXIC)     5535 non-null   float64
 16  (Volume, ^JKSE)     5352 non-null   float64
 17  (Volume, ^NYA)      5535 non-null   float64
dtypes: float64(18)
memory usage: 842.5 KB
```

As seen in Table 1, the stock index data taken there are 18 columns, which can be grouped into three stock indices, namely ^IXIC (NASDAQ), ^JKSE (JCI), and ^NYA (NYSE). Each of these indices has six data categories: adj close, close, high, low, open, and volume. The explanation of the six categories is as follows:

1) Adj Close is the daily closing price of stock indices adjusted for corporate actions such as stock split, reverse stock, and right issue.
2) Close is the daily closing price of a stock index.
3) High is the daily highest price of a stock index.
4) Low is the daily lowest price of a stock index.
5) Open is the daily opening price of a stock index.
6) Volume is the volume of the number of daily transactions of stock indices in a matter of shares.

3.3. Data Cleaning

The data to be used is the Open Price data. JCI has more missing data, JCI has 324 entries of null data rows, NASDAQ and NYSE have 141 entries of null data rows. This difference is due to JCI (Indonesia), NASDAQ (USA), and NYSE (USA) having differences in the time zones and holidays of the two countries. A large number of JCI null data compared to other indices indicates more holidays in Indonesia, causing stock market transactions to close. The cleaning process removes incomplete data on the date rows and columns. This needs to be done so that the model creation process will not occur bias or errors due to null data. After the data cleaning process, which was initially 5676-row entries, it became 5211-row entries, and there was no more null data. Only complete data will later be used for the following process in creating the LSTM model.

3.4. Preprocessing

In the preprocessing stage, there are two processes, namely scaling data and reshaping data into arrays. Scaling using MinMaxScaler from sklearn.preprocessing library will convert previously large data into a range of values of at least 0 to a maximum of 1. Reshape arrays convert data that was once in the form of a table into an array. Shape data after scaling is (5211, 3)

3.5. Split Dataset

Dataset is split into three parts, training set, validation set, and test set. The data need to be split into three parts because we need to see if our LSTM model has underfitting or overfitting. The ratio of dataset division is training set (70%), validation set (15%), and test set (15%)

1) Training set/train_data, 3647 data with shape (3647, 3).
2) Validation set/val_data, 782 data with shapes (782, 3).
3) Test set/test_data, 782 data with shapes (782, 3).

3.6. Create Timestamp Data

Creating timestamp data using a 60-day timestamp creates a 3-dimensional array by using the previous 60 days of data as the third dimension as input and label it as dataset x. Create an output dataset with JCI value minus the first 60 data for each dataset and label it as dataset y. Each dataset shape after creating data x and y:

1) x_train data with shape (3587, 60, 3)
2) y_train data with shape (3587, 1)
3) x_val data with shape (722, 60, 3)
4) y_val data with shape (722, 1)
5) x_test data with shape (722, 60, 3)
6) y_test data with shape (722, 1)

3.7. JCI Forecasting Model

Table 2: LSTM Model

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 60, 60)            15360

 lstm_1 (LSTM)               (None, 60, 60)            29040

 lstm_2 (LSTM)               (None, 60, 60)            29040

 lstm_3 (LSTM)               (None, 60, 60)            29040

 lstm_4 (LSTM)               (None, 60)                29040

 dense (Dense)               (None, 1)                 61

=================================================================
Total params: 131,581
Trainable params: 131,581
Non-trainable params: 0
_________________________________________________________________
```

As seen in Table 2, LSTM models are formed using six layers, 5 LSTM layers, and 1 Dense layer. Output the first to fourth shape layers of 3 dimensions (None, 60, 60), which means:

1) The first dimension (None) signifies the batch size. None means that the batch size is not yet known or in the input.
2) The second dimension (60) signifies LSTM units.
3) The third dimension (60) signifies the input shape of the x_train data.

The process is compiling a model using adam's optimizer. Loss using MAE and MSE. The number of epochs used is 2, namely 100 and 300 epochs with a batch size of 64. Train model with train_data, x_train as input, and y_train as output. The validation model uses val_data, x_val as input, and y_val as output. The selection of adam optimizer refers to previous research where the use of adam is most optimal than other optimizers. Batch size 64 means that 64 data will be processed in each step train model.

3.8. Metric Error

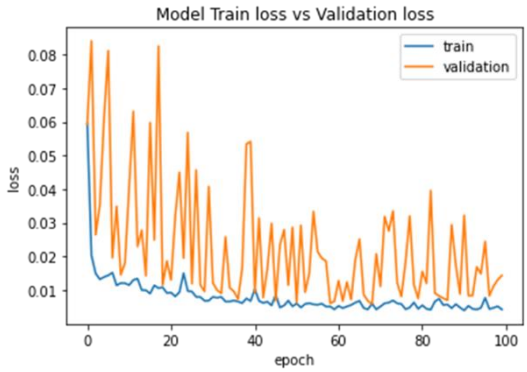
Two metric errors are used, Mean Absolute Error (MAE) and Mean Squared Error (MSE). It aims to compare and determine which model has the lowest loss value. Each metric error will be trained using two numbers of epochs to find the optimum model or the lowest loss.

3.9. Forecasting Visualization

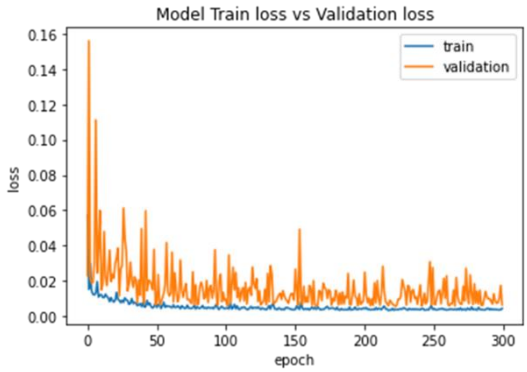
The last process is the visualization of forecasting, which is the process of using the finished LSTM model to forecast JCI trends by using x_test data as input. Visualization is using graphs by comparing the JCI true value or actual value with the forecasted JCI value. This benchmarking aims to see whether the forecasting results of the LSTM model can forecast/follow the JCI trend.

4. Results and Analysis

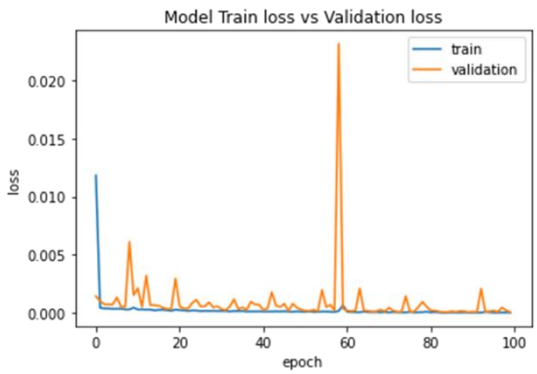
4.1. Train Loss and Validation Loss



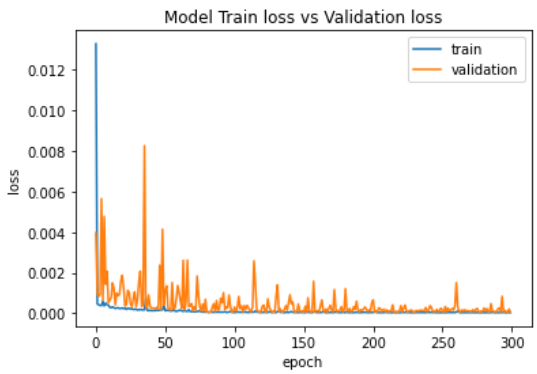
Mean Absolute Error With 100 Epochs (Model 1)



Mean Absolute Error With 300 Epochs (Model 2)



Mean Squared Error With 100 Epochs (Model 3)



Mean Squared Error With 300 Epochs (Model 4)

Figure 2: Train Loss vs. Validation Loss

As seen in Figure 2, the results of the trained model using MAE and MSE from epochs 100 and 300, the existing loss is getting lower. The value of loss and val loss is getting lower. From the chart above, it can be seen that the current model does not experience overfitting or underfitting. The difference between loss and val loss is minimal.

4.2. Forecasting Results

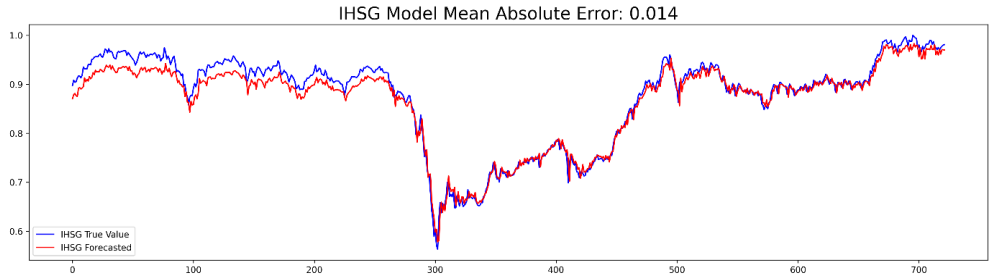


Figure 3: Forecasting Results Model 1

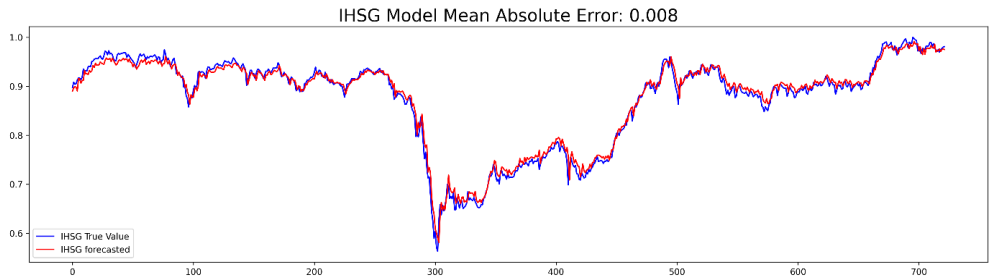


Figure 4: Forecasting Results Model 2

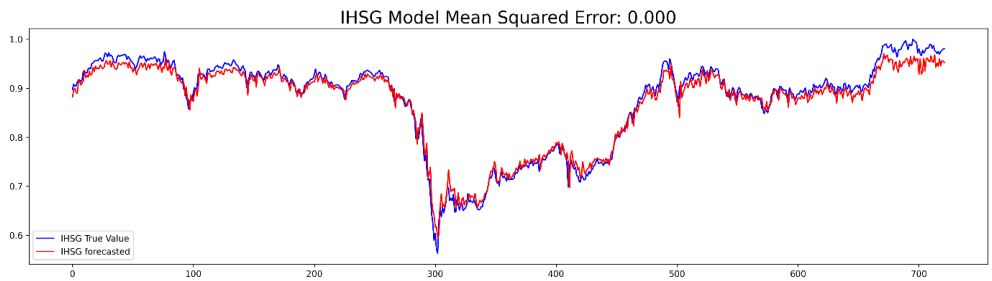


Figure 5: Forecasting Results Model 3

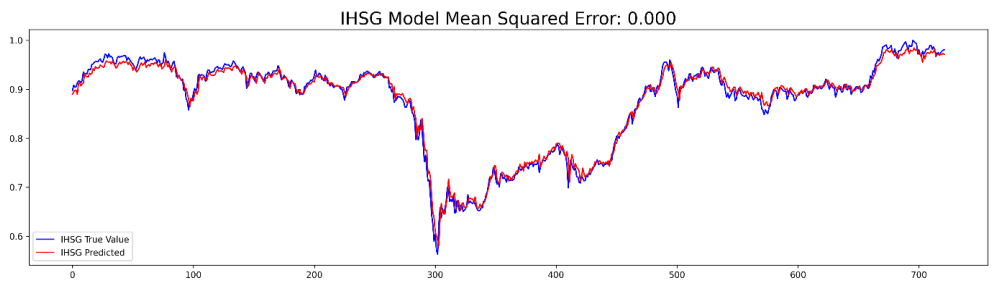


Figure 6: Forecasting Results Model 4

As seen in the figure above, all models can forecast and follow the movement of JCI trend direction. The difference in each model is how close the forecasting JCI value to the actual JCI value. The more closer the value forecasted to the actual value is better.

4.3. Most Optimum Model

The most optimal model is the one that has the lowest MAE and MSE loss values forecasting the test data. For comparison can be seen in Table 3.

Table 3: Loss Forecasting

Loss	Model 1	Model 2	Model 3	Model 4
MAE	0.014314	0.008808	0.012405	**0.008487**
MSE	0.000316	0.000136	0.000250	**0.000128**

Based on Table 3, all models have a low loss value and can forecast the movement of the daily JCI trend. The lowest MAE and MSE value are found in model 4, which is a model made using MSE with 300 epochs, then model 4 is the most optimal model.

5. Conclusion

Some conclusions that can be drawn from this study include:

- All model LSTM that is created can forecast the daily JCI trend movement.
- The most optimal model is the MSE model with 300 epochs (model 4)

References

A. Jayanth Balaji, D. S. Harish Ram, dan Binoy B. Nair. 2018. Applicability of Deep Learning Models for Stock Price Forecasting An Empirical Study on BANKEX Data. Elsevier B.V. Procedia Computer Science 143 (2018) 947–953.

Adhitio Satyo Bayangkari Karno. 2020. Prediksi Data Time Series Saham Bank BRI Dengan Mesin Belajar LSTM (Long Short Term Memory). Journal of Information and Information Security (JIFORTY) Vol. 1, No. 1, Juni 2020, 1 – 8.

Adil Moghar, dan Mhamed Hamiche. 2020. Stock Market Prediction Using LSTM Recurrent Neural Network. Elsevier B.V. Procedia Computer Science 170 (2020) 1168–1173.

Ahmad Ashril Rizal, dan Siti Soraya. 2018. Multi Time Steps Prediction Dengan Recurrent Neural Network Long Short Term Memory. Jurnal MATRIK Vol.18 No.1 (Nopember) 2018, Hal 115-124.

Ahmad Fauzi. 2019. Forecasting Saham Syariah Dengan Menggunakan LSTM. Al Masraf: Jurnal Lembaga Keuangan dan Perbankan-Volume 4, Nomor 1, Januari-Juni 2019.

Anthony So, Thomas V. Joseph, Robert Thas John, Andrew Worsley, and Dr. Samuel Asare. 2020. The Data Science Workshop. Packt Publishing Ltd.

Can Yang, Junjie Zhai, dan Guihua Tao. 2020. Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory. Hindawi. Mathematical Problems in Engineering Volume 2020, Article ID 2746845.

Dicky Bery, dan Saparila Worokinasih. 2018. Pengaruh Indeks Harga Saham Global Terhadap Indeks Harga Saham Gabungan (IHSG) (Studi Pada Bursa Efek Indonesia Periode 2014 2017). Jurnal Administrasi Bisnis (JAB)|Vol. 64 No. 1 November 2018.

Kevin Johan, Julio C. Young, dan Seng Hansun. 2019. LSTM-RNN Automotive Stock Price Prediction. International Journal Of Scientific & Technology Research Volume 8, Issue 09, September 2019.

Khaled A. Althelaya, El-Sayed M. El-Alfy, Salahadin Mohammed. 2018. Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU). Department of Information and Computer Science, College of Computer Sciences and Engineering King Fahd University of Petroleum and Minerals.

Matthew Moocarme, Mahla Abdolahnejad, and Ritesh Bhagwat. 2020. The Deep Learning with Keras Workshop Second Edition. Packt Publishing Ltd.

Rahmadi Yotenka, dan Fazano Fikri El Huda. 2020. Implementasi Long Short-Term Memory Pada Harga Saham Perusahaan Perkebunan Di Indonesia. Jurnal UJMC, Volume 6, Nomor 1, Hal. 9 – 18.

Sidra Mehtab, Jaydip Sen, dan Abhishek Dutta. 2020. Stock Price Prediction Using Machine Learning and LSTM Based Deep Learning Models. Department of Data Science and Artificial Intelligence, Praxis Business School.

Soffa Zahara, Sugianto, M. Bahril Ilmiddafiq. 2019. Prediksi Indeks Harga Konsumen Menggunakan Metode Long Short Term Memory (LSTM) Berbasis Cloud Computing. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi) Vol. 3 No. 3 (2019) 357 – 363.

Zineb Lanbouri, dan Said Achchab. 2020. Stock Market prediction on High frequency data using Long Short Term Memory. Elsevier B.V. Procedia Computer Science 175 (2020) 603–608.