

Jakarta Composite Index Model Before and During COVID-19 Using CNN-LSTM

Yogi Anggara¹, Epha Diana Supandi^{1,*}

¹Department of Mathematics, UIN Sunan Kalijaga, Yogyakarta, Indonesia

*Corresponding author. Email: epha.supandi@uin-suka.ac.id

ABSTRACT

Deep Learning is a subset of artificial intelligence and machine learning, which is the development of multiple layered neural networks. There are many sectors that deep learning can be applied to such as computer vision, natural languages processing, and even time series data forecasting. One of the deep learning algorithms that have depth in forecasting time series data is CNN-LSTM. CNN-LSTM (Convolutional Neural Network - Long Sort Term Memory) is a deep learning algorithm that uses a convolution layer to automate data extraction and an LSTM layer to learn data patterns by paying attention to the order in the data. In this study, CNN-LSTM was used to model the JCI (Jakarta Composite Index) before the COVID-19 period and during the COVID-19 period. JCI data was taken from December 1, 2018 to June 1, 2021. JCI data was split into data training, data validation, and data testing. Based on the analysis, the MAPE value was 1.4% for the JCI test data before COVID-19 and 0.5% for the JCI test data during COVID-19. From the MAPE value, it can be said that CNN-LSTM has excellent forecasting capabilities for JCI data before and during COVID-19.

Keywords: Deep Learning, CNN-LSTM, JCI, COVID-19.

1. INTRODUCTION

Coronavirus Disease (COVID-19) is a disease caused by Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2). Coronavirus is an RNA virus with a particle size of 120-160 nm. This virus mainly infects animals, including bats and camels. Before the outbreak of COVID-19, there were 6 types of corona viruses that could infect humans, namely Alphacoronavirus 229E, Alphacoronavirus NL63, Betacoronavirus OC43, Betacoronavirus HKU1, Severe Acute Respiratory Disease Coronavirus (SARSCoV), and Middle East Coronavirus Respiratory Syndrome (MERSCoVirus) [1]. Currently, the spread of SARSCoV2 from person to person is the main source of transmission, so the spread is increasingly aggressive. SARSCoV2 is transmitted from symptomatic patients through droplets released by coughing or sneezing [2].

The first case was found in Wuhan in December 2019. COVID-19 cases continued to increase until they finally infected many countries [3]. On March 2, 2020, Indonesia reported its first two cases of COVID-19 in Depok [4]. This case caused the stability of several sectors to be shaken, including the business sector, namely the

movement of stock prices [5]. This can be seen from the graph of the movement of the Jakarta Composite Index (JCI). The JCI was unable to recover after the announcement of the first corona virus case in Indonesia until the end of March 2020. This of course would be detrimental to investors. To overcome similar problems that occur in the future, we need a model that can predict the JCI movement, so that it can be a projection for investors in making decisions in the stock market.

There are many methods that can be used to predict stock prices. Generally, the methods that are widely known are fundamental analysis and technical analysis. Fundamental analysis is the study of the economy, industry, and condition of a company to calculate the value of a company's stock. Fundamental analysis focuses on key data in a company's financial statements to calculate whether stock prices have been rising properly. Fundamental analysis is a stock analysis technique that analyzes data or information related to company performance. In general, financial statements are the main source in this analysis, including the use of stock metrics such as earnings per share or earnings per share (EPS), price-earnings ratio (PER), and others [6]. Technical analysts try to identify price patterns and trends in

financial markets and take advantage of these patterns. Analysts try to find client prototypes, such as reversal patterns known as head and shoulders patterns, and examine various patterns such as price, volume, and moving average prices [7].

In addition to fundamental analysis and technical analysis, stock price forecasting can also use time series models, either with classical statistical models and machine learning. The statistical model commonly used for forecasting is ARIMA (Auto Regressive Integrated Moving Average). Time series models in classical statistics usually require assumptions, such as the assumption of stationarity and linear correlation between historical data. Machine learning models seem to fail to identify and capture the nonlinear and complex behavior of time series [8].

In recent years, deep learning have been successfully applied to several challenging real-world prediction problems, including time series prediction [9]–[12]. They are a suitable methodology to deal with the noisy and chaotic nature of time series prediction problems and produce more accurate predictions. Short-term long-term memory network (LSTM) and convolutional neural network (CNN) are perhaps the most popular, efficient, and widely used deep learning techniques [13]. By using the convolution layer as an automatic feature extraction tool and the LSTM layer to study patterns in sequential data, a time series model will be obtained that can learn sequential data patterns automatically from preprocessing time series data such as differencing to the formation of a model. The model that combines the convolution layer with the LSTM layer is called the CNN-LSTM. As a form of development of the deep learning algorithm, CNN-LSTM has its own advantages when compared to classical models, including being robust against noise and the resulting model is not fixed on a linear model [14]. Previous research found that the CNN-LSTM model has the highest accuracy in predicting stock prices when compared to other deep learning algorithms, such as MLP, RNN, CNN, and LSTM [15].

The purpose of this research was to find out how well CNN-LSTM predicts JCI before and during COVID-19. Then it will also be simulated whether the CNN-LSTM trained using JCI data before COVID-19 still predicts JCI during COVID-19 well. This research is expected to be a

reference for investors or researchers in predicting JCI when unexpected cases such as COVID-19 arise.

2. CNN-LSTM

CNN-LSTM is a deep learning algorithm that uses a convolution layer to automate data extraction and an LSTM layer to study patterns in sequential data. Deep learning itself is a machine learning technique that uses multiple layers of non-linear information processing for the extraction and transformation of supervised or unsupervised features, as well as for pattern analysis and classification [16]. Deep learning is built using artificial neural networks. In the training process, artificial neural networks generally use gradient-based methods to update their weights [17]. CNN, founded by Lecun et al., (1998), was used to automate feature extraction [18]. Generally CNN consists of a convolution layer and a pooling layer. The convolution layer on CNN consists of several convolution kernels of a certain size. The way it works is by doing dot multiplication on sub-data. The kernel is then shifted n steps to apply to other sub-data. The output at the convolution layer is then subjected to an activation function. Figure 1 illustrates data with dimensions of 1x4 which is subject to a 1-dimensional convolution kernel with a kernel length of 2 and a step length of 1.

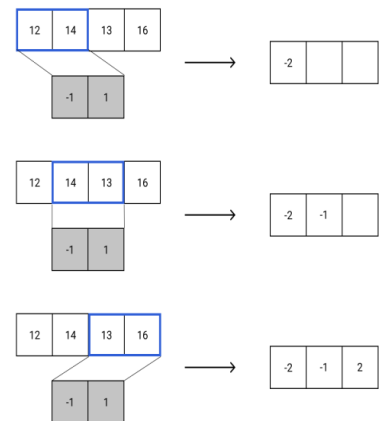


Figure 1. How a 1-dimensional convolution kernel with a kernel length of 2 and a step length of 1 works on data with dimensions of 1x4

In the CNN model, there is also a pooling layer that is used to reduce the feature dimensions that are too large. One layer of pooling that is often used is max pooling. This layer will take the maximum value from each pooling kernel.

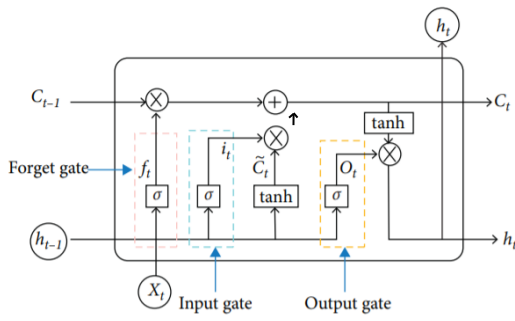


Figure 2. LSTM memory cell architecture [15]

LSTM, which was founded by Hochreiter & Schmidhuber (1997), is a modified form of the recurrent neural network [19]. LSTM is designed to be able to overcome the vanishing gradient problem that occurs in recurrent neural network training [20]. In contrast to recurrent neural networks, in LSTM there are forget gates, input gates, and output gates. The workings of the LSTM are as follows [15]:

1. The output value of the last moment and the input value of the current time are entered into the forget gate, and the output value of the forget gate is obtained after the calculation process, as shown in the following formula:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

where is the range of values of f_t is (0,1), W_f is the weight of the forget gate, b_f is the bias of the forget gate, x_t is the input value of the current time, and h_{t-1} is the output value of the last moment.

2. The last time output value and the current time input value are entered into the input gate, and the output value and the input gate candidate cell status are obtained after calculation, as shown in the following formula:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

where the range of values it is (0,1), W_i is the weight of the input gate, b_i is the bias of the input gate, W_c is the candidate input gate weight, and b_c is the input gate candidate bias.

3. Update the current cell state as follows:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

where the range of C_t is (0,1).

4. The output h_{t-1} and input x_t are accepted as input from the output gate at time t, and the output from the output gate is as follows:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

where the range of o_t is (0,1), W_o is the weight of the output gate, and b_o is bias output gate.

5. The LSTM output value is obtained by calculating the output gate and state of a cell, as shown in the following formula:

$$h_t = o_t * \tanh(C_t) \quad (6)$$

3. MATERIALS AND METHOD

The data used is the daily closing price of the JCI before the COVID-19 pandemic in the period and the closing price of the JCI during the COVID-19. This data is secondary data that comes from the Yahoo Finance website. For the JCI data before the COVID-19 pandemic, the sample taken was 457 days, i.e. in the period 1 December 2018 to 1 March 2020. For the JCI data during the COVID-19 pandemic, the sample taken was also 457 days, i.e. in the period 2 March 2020 to June 1, 2021. The two data are then split into 72% data training, 18% data validation, and 10% data testing.

This research uses the CNN-LSTM model as an alternative in forecasting JCI before and during COVID-19. Then R2, RMSE, MAE, and MAPE were used to evaluate the ability of CNN-LSTM in predicting JCI. Next, the most important lag in the model will be determined by using permutation importance. CNN-LSTM is a black-box model, which means it is difficult to see directly what the model does to get predictions. As a result, it is difficult to know explicitly which features have the greatest impact on the model in predicting. As an alternative, permutation importance is used to determine feature importance on CNN-LSTM. The way permutation importance works are by calculating the score resulting from randomizing the values in a feature column [21]. If the score caused by randomization of values in a feature column is large, it can be said that the feature has a large contribution to the model in predicting. Feature importance weight is obtained by reducing the model score with the average score after the features are randomly shuffled.

The CNN-LSTM for forecasting JCI movements before and during COVID-19 steps in this research are as follows:

1. Split the time series data into 72% data training, 18% data validation, and 10% data testing. Furthermore, the data that has been shared is converted into the form of supervised learning data, namely data consisting of features and targets. In this research, the first 7 lags will be used as features.
2. Determine the arrangement of the CNN-LSTM layers to be used. In this research, the layer arrangement used is a one-dimensional convolution layer with a kernel length of 2 and a step length of 1 as input, an LSTM layer, a full-connected layer, and an output layer. The activation function used for each layer is the ReLU function, as shown in the following:

$$f(x) = \max(0, x) \quad (7)$$

Due to its simple function, ReLU has low complexity when compared to other activation functions. ReLU is also able to overcome the vanishing gradient [22].

3. Perform hyperparameter tuning using Hyperband algorithm to obtain optimal parameters. On various kernel-based and deep learning issues, Hyperband can provide more than an order of magnitude faster than popular Bayesian optimization methods [23]. In this research, the parameters to be optimized are the number of kernels in the convolution layer, the number of units in the LSTM layer, the number of units in the full-connected layer, and the learning rate.
4. Apply optimal parameters to CNN-LSTM and train it using training data and validation data. In this study, the optimization function used is Adam. It is a simple and computationally efficient algorithm for gradient-based optimization of the stochastic objective function [24]. Thus, Adam's optimization function is suitable for training deep learning algorithms such as CNN-LSTM which is computationally large.
5. Calculating the MAPE in predicting the data tasting. If MAPE is still too large, go back to step 2.

4. RESULT AND DISCUSSION

4.1. Data Description

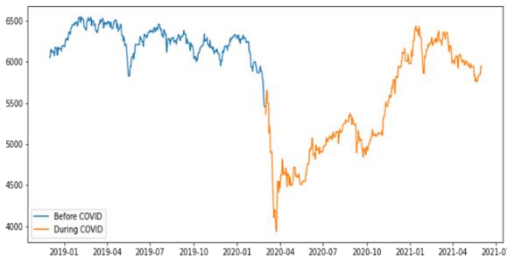


Figure 3. JCI close price plot

Before the COVID-19 pandemic, it was seen that the JCI was still able to rebound. On March 2, 2020, where the first case of COVID-19 in Indonesia was announced, it was seen that the JCI was at its lowest point in the last 3 months. The next day, the JCI rose again and finally continued to fall until March 15, 2021. After that, the JCI trend continued to gradually increase.

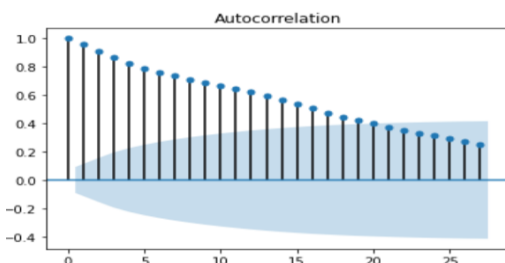


Figure 4. Plot autocorrelation of JCI data before COVID-19

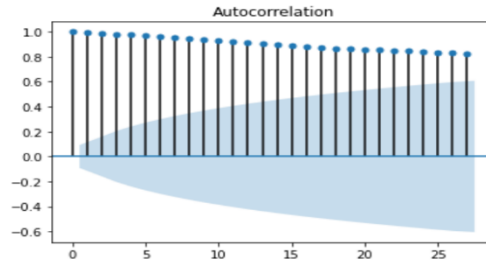


Figure 5. Plot autocorrelation of JCI data during COVID-19

Lag-1 to lag-19 in JCI data before COVID-19 autocorrelate significantly. Then lag-1 to lag-25 on the JCI data during COVID-19 also appears to be autocorrelated significantly. This indicates that the JCI data before and during COVID-19 is non-stationary data. Then, the autocorrelation plot also shows that the JCI data does not have a seasonal pattern.

4.2. CNN-LSTM Model

The software used for training the CNN-LSTM model is Google Collaboratory Notebook and the programming language is Python. In accordance with the research method above, first the data is split into 72% data training, 18% data validation, and 10% data testing. Furthermore, the data that has been shared is converted into supervised learning data, where the first 7 lags are used as features. Then, the training data and validation data are used for the hyperparameter tuning process. Due to computational reasons, the researcher limits the options for the number of convolution layer filters, the number of LSTM units, and the number of full-connected units for hyperparameter tuning from the range 8 to 512. Then for the learning rate options are 0.01, 0.001, and 0.0001. The optimal CNN-LSTM parameters for JCI data before COVID-19 after hyperparameter tuning are 200 filter units for the convolution layer, 456 units for the LSTM layer, 456 units for the full-connected layer, and a learning rate of 0.001. Then, the optimal CNN-LSTM parameters for IHSG data during COVID-19 after hyperparameter tuning are 488 filter units for the convolution layer, 232 units for the LSTM layer, 456 units for the full-connected layer, and a learning rate of 0.0001.

Table 1. Evaluation metrics for JCI test data before COVID-19

R^2	RMSE	MAE	MAPE
0.650797	115.849016	86.12038	0.014883

After CNN-LSTM was configured using optimal parameters and trained using JCI data before COVID-19, the prediction results are shown in Figure 6. In the graph it can be seen that the model results are quite good at predicting because they can follow trend movements, especially when the trend is stationary. Then by using the test data (Table 1), R^2 is obtained by 65%, which means that 65% of the variation in the data can be explained well by features and the other 35% is influenced by other factors. The RMSE value in representing the magnitude of

the error distribution in the model and MAE can represent the average deviation of predictions from the original data. Because the range of testing data is between 5000 to 6000, the RMSE value of 115,849 and MAE of 86,120 indicate that the predictive ability of the model is quite good. Furthermore, the MAPE of 1.4% is also obtained, which means that CNN-LSTM has very good forecasting capabilities for JCI data before COVID-19.

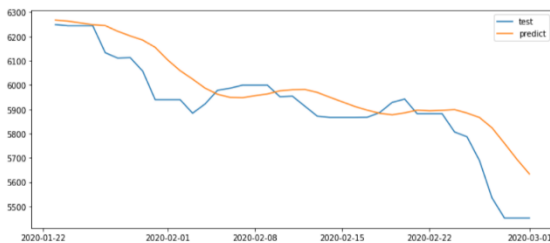


Figure 6. Prediction plot for JCI test data before COVID-19



Figure 7. Prediction JCI during pandemic used model trained with JCI data before COVID

CNN-LSTM still provides a good performance in predicting JCI during COVID-19 even though the data used in the training is JCI before COVID-19. Thus, it can be said that CNN-LSTM has a good ability to generalize forecasts even though it is faced with data with different conditions such as COVID-19.

Table 2. Evaluation metrics for JCI test data during covid-19

R^2	RMSE	MAE	MAPE
0.705633	43.293638	33.53433	0.005678

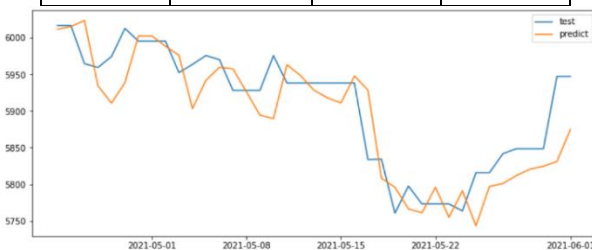


Figure 8. Prediction plot for JCI test data during COVID-19

After CNN-LSTM was configured using optimal parameters and trained using JCI data before COVID-19, the prediction results are shown in Figure 6. In the graph it can be seen that the model results are quite good at

predicting because they can follow trend movements, especially when the trend is stationary. Then by using the test data (Table 1), R^2 is obtained by 65%, which means that 65% of the variation in the data can be explained well by features and the other 35% is influenced by other factors. The RMSE value in representing the magnitude of the error distribution in the model and MAE can represent the average deviation of predictions from the original data. Because the range of testing data is between 5000 to 6000, the RMSE value of 43.294 and MAE of 33.534 indicate that the predictive ability of the model is quite good. Furthermore, the MAPE of 1.4% is also obtained, which means that CNN-LSTM has very good forecasting capabilities for JCI data before COVID-19

After CNN-LSTM was configured using optimal parameters and trained using JCI data after COVID-19, the prediction results shown in Figure 7. In the graph it can be seen that the model results are quite good at predicting because they are able to follow trend movements. Then by using the test data (Table 2), R^2 is obtained by 70%, which means that 70% of the variation in the data can be explained well by features and the other 30% is influenced by other factors. Because the range of testing data is between 5000 to 6000, the RMSE value of 43.294 and MAE of 33.534 indicate that the predictive ability of the model is quite good. Furthermore, the MAPE of 0.5% is also obtained, which means that CNN-LSTM has very good forecasting capabilities for JCI data during COVID-19.

4.3. Feature Importance

Based on Table 3 and Table 4, lag-1 and lag-2 on CNN-LSTM for JCI data before and during COVID-19 have a big influence on the model in predicting. Then there is a difference in the third to sixth rank of feature importance. Furthermore, lag-7 is the feature importance with the lowest ranking for both the JCI data before and during COVID-19. Intuitively, it makes sense that the initial lag has a big impact on the forecasting process.

Table 3. Permutation importance CNN-LSTM for JCI before covid-19 using R^2 score

Weight	Feature
0.0979 ± 0.0143	lag-1
0.0415 ± 0.0146	lag-2
0.0253 ± 0.0125	lag-3
0.0216 ± 0.0058	lag-4
0.0192 ± 0.0090	lag-6
0.0142 ± 0.0072	lag-5
-0.0018 ± 0.0029	lag-7

Table 4. Permutation importance CNN-LSTM for JCI during covid-19 using R^2 Score

Weight	Feature
0.0792 ± 0.0054	lag-1
0.0596 ± 0.0119	lag-2
0.0456 ± 0.0038	lag-5
0.0353 ± 0.0041	lag-6

0.0258 ± 0.0064	lag-4
0.0208 ± 0.0024	lag-3
0.0052 ± 0.0010	lag-7

In the first model (Table 3), interesting things appear in lag-5 and lag-6. Although the autocorrelation of lag 5 is greater than lag-6, it does not guarantee that lag-5 is considered more important than lag-6 by the model. This also occurs in the second model (Table 4), although the autocorrelation at lag-3 is greater than the autocorrelation at lag-4 to lag-6, it does not guarantee that lag-3 is considered more important than lag-4 to lag-6 by the model.

5. CONCLUSIONS AND SUGGESTIONS

CNN-LSTM is a deep learning algorithm that uses a convolution layer to automate data extraction and an LSTM layer to learn data patterns by paying attention to the order in the data. As one of the developments of artificial neural network algorithms, CNN-LSTM has the capabilities of artificial neural networks, including robust against noise and the resulting model is not fixed on a linear model. Based on the results of the analysis above, the MAPE value was 1.4% for the JCI test data before COVID-19 and 0.5% for the JCI test data during COVID-19. From the MAPE value, it can be said that CNN-LSTM has excellent forecasting capabilities for JCI data before and during COVID-19. In the experiment above, it can also be seen that CNN-LSTM still has a good ability to predict JCI during COVID-19 even though the data used for the training process is JCI data before COVID-19. Thus, it can be said that CNN-LSTM is still able to predict well even though it is faced with data with different conditions such as COVID-19. From these various advantages, CNN-LSTM can be an option in predicting time series data such as JCI. Then, there are similarities between the CNN-LSTM model trained with JCI data before and during COVID-19, namely lag-1 and lag-2 ranked in the top 2 feature importance and lag-7 ranked last for feature importance. Tables 3 and 4 also show that the large autocorrelation of the lag does not guarantee that the lag is considered an important feature by the model.

The CNN-LSTM model in this research is still very simple, only consisting of a convolution layer, a convolution layer, and a full-connected layer. The features used only come from past data from JCI closing prices. Therefore, the suggestion for further research is to improve the CNN-LSTM layer which is suitable for JCI data and other features that can support the forecasting process.

REFERENCES

- [1] S. Riedel, S. A. Morse, T. A. Mietzner, and S. Miller, *Jawetz Melnick & Adelbergs Medical Microbiology* 28 E, McGraw Hill Professional, 2019.
- [2] Y. Han and H. Yang, The transmission and diagnosis of 2019 novel coronavirus infection disease (COVID-19): a Chinese perspective, *Journal of Medical Virology*, 92(6):639-644, 2020, doi: 10.1002/JMV.25749.
- [3] C. Huang et al, Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China, *Lancet*, 395(10223):497-506, 2020, doi: 10.1016/S0140-6736(20)30183-5
- [4] A. Susilo et al, Coronavirus disease 2019: Tinjauan literatur terkini. *Jurnal Penyakit Dalam Indonesia*, 7(1):45-67, 2020, doi: 10.7454/jpdi.v7i1.415.
- [5] D. Rakhmawati, R. Wahyudi, and C. G. Yuliawan. *Pemodelan harga saham IHSG selama pandemi COVID-19 menggunakan arima non musiman*. *Probisnis*, 13(2):39-48, 2020.
- [6] Fakhruddin and M. Hendy. *Teori Portofolio Dan Analisis Investasi*. Yogyakarta: BPFE, 2008.
- [7] D. R. Artha, *Analisis Fundamental, Teknikal Dan Makroekonomi Harga Saham Sektor Pertanian*. *Jurnal Manajemen dan Kewirausahaan (Journal of Management and Entrepreneurship)*, 16(2):175-184, 2014, doi: 10.9744/jmk.16.2.175.
- [8] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN-LSTM model for gold price time-series forecasting," *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17351-17360, 2020, doi: 10.1007/s00521-020-04867-x.
- [9] J. Li, Q. Dai, and R. Ye, "A novel double incremental learning algorithm for time series prediction," *Neural Comput. Appl.* 2018 3110, vol. 31, no. 10, pp. 6055-6077, Mar. 2018, doi: 10.1007/S00521-018-3434-0.
- [10] Y. Ai et al., "A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system," *Neural Comput. Appl.* 2018 315, vol. 31, no. 5, pp. 1665-1677, Apr. 2018, doi: 10.1007/S00521-018-3470-9.
- [11] J. Zheng, X. Fu, and G. Zhang, "Research on exchange rate forecasting based on deep belief network," *Neural Comput. Appl.* 2017 311, vol. 31, no. 1, pp. 573-582, May 2017, doi: 10.1007/S00521-017-3039-Z.
- [12] W. Zou and Y. Xia, "Back propagation bidirectional extreme learning machine for traffic flow time series prediction," *Neural Comput. Appl.* 2018 3111, vol. 31, no. 11, pp. 7401-7414, Jun. 2018, doi: 10.1007/S00521-018-3578-Y.
- [13] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Min. Knowl. Discov.*, vol. 33, no. 4, pp. 917-963, Sep. 2018, doi: 10.1007/s10618-019-00619-1.
- [14] J. Brownlee, *Deep Learning for Time Series*, 1.4. (n.p.): (n.p.), 2018.
- [15] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," *Complexity*, vol. 2020, 2020, doi: 10.1007/S00521-018-3434-0.

- 10.1155/2020/6622927.
- [16] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2013, doi: 10.1561/20000000039.
- [17] Christopher M. Bishop, *Pattern Recognition and Machine Learning*. 2016.
- [18] Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, "GradientBased Learning Applied to Document Recognition," *Proc. IEEE*, no. November, pp. 1–46, 1998.
- [19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [20] V.-D. Ta, C.-M. Liu, and D. A. Tadesse, "Portfolio Optimization-Based Stock Prediction Using Long-Short Term Memory Network in Quantitative Trading," *Appl. Sci.* 2020, Vol. 10, Page 437, vol. 10, no. 2, p. 437, Jan. 2020, doi: 10.3390/APP10020437.
- [21] L. Breiman, "Random Forests," *Mach. Learn.* 2001 451, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [22] A. M. Javid, S. Das, M. Skoglund, and S. Chatterjee, "A ReLU Dense Layer to Improve the Performance of Neural Networks," pp. 2810–2814, May 2021, doi: 10.1109/ICASSP39728.2021.9414269.
- [23] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, pp. 1–52, 2018.
- [24] 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.