

## Research Article

# Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction

Harya Widiputra , Adele Mailangkay , and Elliana Gautama 

*Perbanas Institute, Jakarta 12940, Indonesia*

Correspondence should be addressed to Harya Widiputra; [harya@perbanas.id](mailto:harya@perbanas.id)

Received 11 May 2021; Revised 3 July 2021; Accepted 12 August 2021; Published 23 October 2021

Academic Editor: Abdelalim Elsadany

Copyright © 2021 Harya Widiputra et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

At the macroeconomic level, the movement of the stock market index, which is determined by the moves of other stock market indices around the world or in that region, is one of the primary factors in assessing the global economic and financial situation, making it a critical topic to monitor over time. As a result, the potential to reliably forecast the future value of stock market indices by taking trade relationships into account is critical. The aim of the research is to create a time-series data forecasting model that incorporates the best features of many time-series data analysis models. The hybrid ensemble model built in this study is made up of two main components, each with its own set of functions derived from the CNN and LSTM models. For multiple parallel financial time-series estimation, the proposed model is called multivariate CNN-LSTM. The effectiveness of the evolved ensemble model during the COVID-19 pandemic was tested using regular stock market indices from four Asian stock markets: Shanghai, Japan, Singapore, and Indonesia. In contrast to CNN and LSTM, the experimental results show that multivariate CNN-LSTM has the highest statistical accuracy and reliability (smallest RMSE value). This finding supports the use of multivariate CNN-LSTM to forecast the value of different stock market indices and that it is a viable choice for research involving the development of models for the study of financial time-series prediction.

## 1. Introduction

The study of datasets that vary over time is known as time-series data analysis. Time-series datasets keep track of measurements of the same component over time. To measure a company's performance, financial analysts use time-series data such as stock price fluctuations or profits over time [1]. At the macroeconomic level, the movement of the stock market index, which is a financial time-series statistic, is often associated as one of the key indicators in determining a country's economic situation, making it a crucial issue to be examined over time [2]. The stock market index's movement is determined by a variety of internal and external influences, including the domestic and foreign economic climate, the international situation, industrial prospects, and stock market operations, but it is mostly influenced by the stock market index's historical meaning [3, 4].

Previous research has also shown that complex relationships between series can be found in a variety of time-series data related to real-world processes in the economic and financial realms [3]. It has also been known that several time-series travel together over time because of these interrelationships. It is well understood, for example, that the movement of a stock market index in one country is influenced by the movements of other stock market indices around the globe or in that area. These findings are confirmed by the work of [3, 5–7].

However, most of the developed time-series forecasting methods are single-stand-alone algorithm that utilizes univariate time-series analysis, while little attention has been paid to prediction processes that use the dynamics of interactions between the observed series. In addition, projecting the course of movement of time-series values using only a single algorithm has some serious drawbacks, whether it is an econometric time-series forecasting model or a

machine learning model like an artificial neural network. This is attributed to the high noise and volatility of financial time-series and the fact that the relationship between independent and dependent variables is subject to unpredictable shifts over time [8]. The concept also connects with previous research in the field of neural networks, which found that no one model always outperforms the others for all real-world problems [9]. Thus, in the current financial time-series data analysis and modeling phase, integrating the best algorithms to be able to take advantage of the different advantages possessed by each algorithm by creating an ensemble, which combines multiple forecasting models, has become a growth path [8–11].

Based on the outlined conditions above, two key problems in the field of financial time-series data prediction can be reported. The first is that there is no particular methodology that can often forecast the movement of financial time-series data with the greatest precision. Second, despite the fact that interdependencies between variables in financial time-series data are well-known (e.g., the movement of stock market indices is often determined by other markets), most forecasting models still rely on univariate analysis.

In line with this, the study's aim is to develop a time-series data forecasting model that combines the best features of multiple time-series data analysis models. The hybrid ensemble model developed in this study consists of two key components with distinct functions: (1) extracting important features from the observed time-series data and (2) predicting the value of the time-series data using the described features. To accomplish the first function, a Deep Learning model named the Convolutional Neural Network (CNN), which has been proven to have sound performance in feature extraction, is used, while the Long Short-Term Model (LSTM) is put into place to support the time-series forecasting process.

CNN and LSTM are deep learning neural networks that can learn arbitrarily complicated mappings from inputs to outputs and handle many inputs and outputs automatically. These are useful qualities for time-series forecasting, especially for situations with complicated nonlinear relationships, multivalued inputs, and multistep forecasting. These features are the reason why both models were chosen for this investigation. Furthermore, the proposed model applies a multivariate analysis technique to take advantage of the observed time-series data's relationship trend in forecasting their future values. It is anticipated that having such a structure would aid in improving the accuracy of financial time-series data prediction, especially for multiple parallel stock market indices.

This study uses regular stock market indices from four Asian stock markets, namely, Shanghai, Japan, Singapore, and Indonesia, to check the efficacy of the evolved ensemble model during the COVID-19 pandemic, which spans 242 trading days from January 1, 2020, to December 31, 2021. The data is divided into two parts: a training set of 170 trading days and a comparison set of 72 trading days.

The paper is organized in the following way. Section 2 discusses the research works that apply machine learning in

a particular deep learning model in predicting the trajectory of various time-series datasets. The proposed ensemble of multivariate deep learning models that are utilized for multiple time-series prediction is outlined in Section 3, which will be followed by the experimental setting used in this research. Afterward, the results of the conducted trials are given and discussed, and the article ends with a conclusion and future work section.

## 2. Related Work

*2.1. Deep Learning.* Deep learning is a form of an algorithm in the machine learning area that [12] (1) uses a cascade of multiple layers of nonlinear processing units for feature, extraction, and transformation, where each successive layer uses the output of the previous layer as input; (2) learn in a supervised manner (e.g., classification) and/or unsupervised manner (e.g., pattern analysis); (3) capable of modeling different levels of representation according to different levels of abstraction; levels form a hierarchy of concepts.

Most modern deep learning models are based on neural networks, although they can also include propositional formulas or latent variables arranged in layers in deep generative models, such as nodes in deep belief networks and Boltzmann's machine [13]. In deep learning, each layer of the learning structure (in the sense of building a model) converts the input into a slightly more general representation (model). Primarily, the deep learning process learns in depth to be able to learn which features are optimally placed at a particular level by themselves. Of course, this does not eliminate the need for hand-tuning; for example, varying the number of layers and the size of the layers can provide different levels of abstraction [13, 14].

An artificial neural network with several layers between the input and output layers is known as a Deep Neural Network (DNN) [13, 15]. DNN discovered the right mathematical model for transforming linear and nonlinear inputs to outputs. The network traverses the layers, measuring the likelihood of each output. DNNs are capable of modeling nonlinear interactions that are complex. DNN architecture produces a composition model in which objects are expressed as layered primitive compositions. Additional layers allow feature composition of the lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network [13].

*2.2. Convolutional Neural Network (CNN).* A DNN model that is commonly used in computer vision work is the Convolutional Neural Network (CNN) which has also been applied to acoustic modeling for automated speech recognition (ASR) [16]. The basic structure of CNN is given in Figure 1. CNN are inspired by biological processes [18, 19] because the patterns of connectivity between neurons resemble the organization of the visual cortex of animals. CNN is widely used in image and video recognition, recommendation systems, image classification, medical image analysis, and natural language processing. Findings from

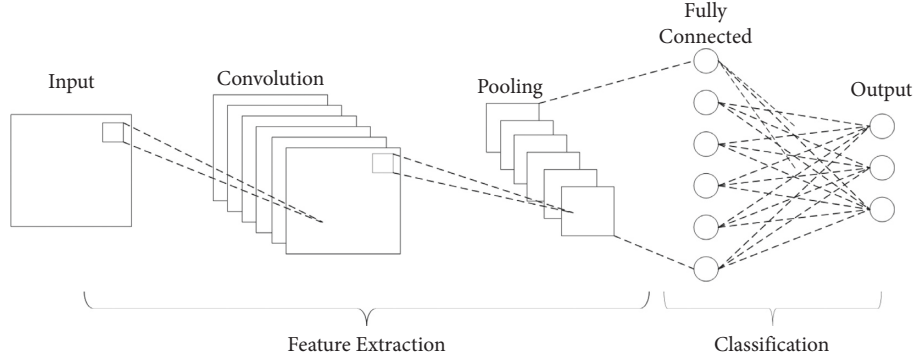


FIGURE 1: Illustration of the feature extraction process from image data to produce the final classification with a basic CNN architecture. The figure is recreated from [17].

previous research confirmed CNN's superiority in processing time-based flowing data [16, 18, 19].

CNN's main feature is the ability to process multi-channel input data, so it is ideal for handling different time-series data with multiple inputs and outputs in this study [19–21]. However, there has not been much research into CNN's success in modeling and forecasting the movement of several time-series data values for deep learning models.

One of the main advantages of CNN is the local perception and weight sharing features, which can greatly reduce the number of parameters, thereby increasing the efficiency of the learning process. In terms of structure, CNN mainly consists of two parts, namely, the convolutional layer and the pooling layer. In this case, each convolution layer contains several convolutional kernels. After the convolution operation that occurs at the convolution layer, the important features of the data are extracted, which are accompanied by an increase in the feature dimensions. To solve this problem and reduce the burden on the training process, a layer of integration is added with the main objective of reducing the number of features extracted before finally producing the final result.

**2.3. Long Short-Term Memory (LSTM).** Aside from CNN, the Long Short-Term Memory model, also known as LSTM, is another well-known DNN model. The LSTM is a recurrent neural network subunit (RNN) [22], and the basic architecture is given in Figure 2. The LSTM algorithm is ideal for classifying, sorting, and making predictions from a single time-series dataset. Previous research has also shown that LSTM is capable of forecasting time-series data [23–26].

The computation process that occurs in the LSTM structure to calculate the predictive time-series data begins with the calculation of the output value from the previous time and the input value from the current time becomes an input to the forget gate, and the processing results from the forget gate are obtained through computation using the following formula:

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

where the value range of  $f_t$  is (0, 1),  $W_f$  is the weight of the forget gate,  $b_f$  is the bias value applied to the forget gate,  $x_t$  is

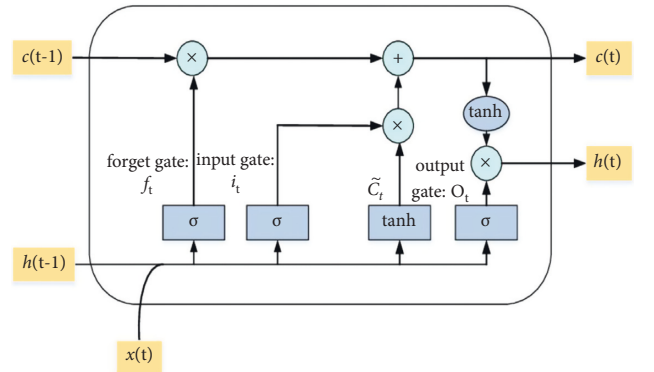


FIGURE 2: Basic architecture of LSTM when being used for time-series prediction in a supervised model. The figure is recreated from [22].

the input value for the current time, and  $h_{t-1}$  is the output value of the previous processing time.

Furthermore, the output value from the previous time and the input value from the current time are also input to the input gate, and the output value and condition of the candidate cell at the input gate are obtained after calculation using 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 value range of  $i_t$  is (0, 1),  $W_i$  is the weight of the gate input,  $b_i$  is the bias value of the gate input,  $W_c$  is the weight of the gate input candidate, and  $b_c$  is the bias value of the gate input candidate. The next stage in the LSTM model is the process of adjusting cell values or model parameters at this time carried out as follows:

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

where the range of values for  $C_t$  is (0, 1). Then, at processing time  $t$ , the output value  $h_{t-1}$  and the input value  $x_t$  become the input for the output gate, and the output from the gate output is calculated using the following formula:

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

where the value range of  $o_t$  is (0,1),  $W_o$  is the weight of the gate output, and  $b_o$  is the bias value of the gate output. Finally, the final output value of the LSTM is generated by the output gate and is the result of the calculation using the following formula:

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

In this case,  $\tanh$  is an activation function that can be tailored to the needs and characteristics of the problem to be resolved. Consequently, LSTM has the ability to process recorded data in a specific time sequence and therefore has been widely used in the process of analyzing and modeling time-series data [26, 27].

Both LSTM and CNN can be used to create deep learning models that can investigate complicated and unknown patterns in large and varied data sets. The idea was then designed to be able to combine different deep learning models, both CNN and LSTM-based, to create an ensemble. Since the findings of previous studies show that each model has different abilities to catch secret trends in the data, it is hoped that the solutions provided will be stronger and more detailed with this ensemble method.

#### 2.4. CNN and LSTM for Financial Time-Series Prediction.

The financial market is currently a noisy, nonparametric competitive environment, and there are two major types of stock price or stock market index forecasting methods: conventional econometric analysis and machine learning. Traditional econometric methods or equations with parameters, on the other hand, are notorious for being unsuitable for analyzing complex, high-dimensional, and noisy financial time-series results [28]. As a result, in recent years, developments in the field of machine learning have emerged as a viable option, especially for neural networks. Since it can derive data features from a vast number of high-frequency raw data without relying on previous information, neural networks have become a hot research path in the field of financial forecasting.

In 2017, Chen and Hao investigated stock market index prediction using a basic hybridized configuration of the feature weighted support vector machine and feature weighted K-nearest neighbor, resulting in enhanced short, medium, and long-term prediction capabilities for the Shanghai Stock Exchange Composite Index and Shenzhen Stock Exchange Component Index [29]. In 2017, Chong et al. released a thorough analysis of the use of deep learning networks for stock market forecasting and prediction [30]. According to the study's empirical results, deep neural networks will derive additional information from the residuals of the autoregressive model and improve the prediction accuracy. Hu et al. experimental findings from 2018 indicate that, while CNN is most widely used for image recognition and feature extraction, it can also be used for time-series prediction since it is a deep learning model, but the forecasting accuracy of CNN alone is relatively poor [31].

In 2019, Hoseinzade and Haratizadeh proposed a CNN-based tool that can be used to extract features from a variety of data sources, including different markets, to predict their

future [17]. In addition, Zhong and Enke proposed that in 2019, hybrid machine learning algorithms can be used, as they have been shown to be effective in predicting the stock market's normal return path [32]. Nabipour et al. compared various time-series prediction techniques on the Tehran stock exchange in 2020 and found that the LSTM produces more accurate results and has the best model fitting ability [33]. Kamalov forecasted the stock prices of four big US public firms using MLP, CNN, and LSTM in 2020. These three approaches outperformed related experiments that predicted the trajectory of price change in terms of experimental findings [34]. In 2020, Liu and Long developed a high-precision short-term forecasting model of financial market time-series focused on LSTM deep neural networks, which they compared with the BP neural network, standard RNN, and enhanced LSTM deep neural networks. The findings revealed that the LSTM deep neural network has a high forecasting precision and can accurately model stock market time-series [23]. Moreover, Lu et al. proposed an ensemble structure of CNN-LSTM and proved that such model is effective when being applied to predict Shanghai Composite Index [31].

Additionally, Mahmud and Mohammed performed a survey on the usage of deep learning algorithms for time-series forecasting in 2021, which found that deep learning techniques like CNN and LSTM give superior prediction outcomes with lower error levels than other artificial neural network models [35]. Furthermore, their literature study discovered that merging many deep learning models greatly improved time-series prediction accuracy. However, Mahmud and Mohammed also conveyed that the performance of CNN and LSTM is not always consistent, with CNN outperforming LSTM at times and vice versa. In general, CNN tends to have superior predictive capacity when dealing with time-series data comprised of a collection of images, but LSTM appears to be superior when dealing with numerical data.

In conclusion, findings from previous studies show that using deep learning models such as CNN and LSTM for financial time-series prediction is successful. However, compared to LSTM, CNN has poorer prediction accuracy when applied to numerical time-series data due to its key characteristics, which include a high point in feature extraction. While at the same time, LSTM also has a weakness related to its capability of extracting the most valuable features from a data set when being compared to CNN. Consequently, constructing a composite or ensemble model that takes advantage of each combination model to overcome its weaknesses to increase time-series prediction accuracy is logical. Furthermore, based on findings from prior studies indicating cointegration between financial time-series data in a real-world environment, it is then reasonable to include a multivariate time-series analysis technique into the ensemble model. Therefore, the following are the major contributions of this work:

- (1) Development of a new ensemble of multivariate deep learning approach named the multivariate CNN-LSTM that utilizes the superior feature of CNN and



LSTM model for simultaneous multiple parallel financial time-series prediction by considering the state of correlation between series into the forecasting process.

- (2) Evaluation of proposed model by conducting experiments using data from real-world setting, i.e., four stock market indices from the Asia region, to confirm that constructing an ensemble model, which uses the core features of each model and incorporates multivariate time-series analysis, offers better forecasting accuracy, and is more suited for multiple parallel financial time-series forecasting by contrasting the evaluation indicators of proposed multivariate CNN-LSTM with stand-alone CNN and LSTM.

### 3. Multivariate CNN-LSTM Model

**3.1. Multivariate Time-Series Analysis.** When dealing with variables from real-world phenomena such as economics, weather, ecology, and so on, the value of one variable is often dependent on the historical values of other variables as well. For example, a household's spending expenses can be influenced by factors such as revenue, interest rates, and investment expenditures. If both of these factors are linked to consumer spending, it makes sense to factor in their circumstances when forecasting consumer spending. In other words, denoting the related variables by  $x_{1,t}, x_{2,t}, \dots, x_{k,t}$ , prediction of  $x_{1,t+h}$  at the end of period  $t$  may be represented by the following form:

$$\hat{x}_{1,t+h} = f_1(x_{1,t}, x_{2,t}, \dots, x_{k,t}, x_{1,t-1}, x_{2,t-1}, \dots, x_{k,t-1}, x_{1,t-2}, \dots). \quad (7)$$

Similarly, a forecast for the second component may be dependent on all system's previous values. This equation can be used to express a projection of the  $k^{th}$  variable more broadly:

$$\hat{x}_{k,t+h} = f_k(x_{1,t}, \dots, x_{k,t}, x_{1,t-1}, \dots, x_{k,t-1}, \dots). \quad (8)$$

There are several time-dependent variables in a multivariate time series. Each variable depends not only on its previous value but also on other variables. A multiple time-series  $x_{k,t}$ ,  $k = 1, \dots, k$ ;  $t = 1, \dots, t$  is a set of time series where  $k$  is the series index and  $t$  is the time-point, and equation (8) expresses the prediction of  $\hat{x}_{k,t+h}$  as a function of a multiple time series [24]. The main goal of multiple time-series analysis, like univariate time-series analysis, is to find appropriate functions  $f_1, \dots, f_q$ , where  $q$  is the number of constructed functions that can be used to forecast the potential values of a variable with good properties. Learning about the interrelationships between a variety of variables is often of concern. For example, in a stock exchange environment with several markets, whether internationally or in a specific area, one may be interested in the possible effect of how these markets interact with one another.

This work explores further multivariate time-series analysis models by incorporating different state-of-the-art learning models originating from the machine learning arena, i.e., the deep learning model, to construct an ensemble model that can predict multiple financial time-series data simultaneously.

**3.2. Multivariate CNN-LSTM General Concept.** CNN and LSTM can be used to build deep learning models that could deeply study complex and hidden patterns in massive and diverse data stacks, including time-series data, especially from the financial sector [36, 37]. Empowering the advantages possessed by the two models to achieve the objectives of this study as presented in the Introduction, that is, to improve the accuracy of forecasting the movement of the stock market index, a time-series data forecasting model is created by combining CNN and LSTM, as well as including a multivariate time-series analysis method into the model to allow for simultaneous forecasting of parallel time-series using series correlation analysis.

In this case, the CNN-LSTM model built has different characteristics compared to most time-series data forecasting techniques, which generally work using a univariate analysis approach, where the CNN-LSTM model will utilize correlation information between series in making predictions. This is in line with the findings of various previous studies that a group of time-series data originating from a similar domain tends to have a relationship and influence each other. Therefore, information about the relationship between series should be used in the process of forecasting future conditions.

The built model is an ensemble of CNN and LSTM, which was referred to as the multivariate CNN-LSTM. In this proposed architecture, the time-series data will be reshaped to fit the input data structure that can be processed by the CNN structure and then the LSTM. The multivariate CNN-LSTM model consists of two main layers: the CNN layer, which has the main function of extracting the main features from the processed time-series data. The LSTM layer, which has the main function of calculating the final prediction result.

The CNN and LSTM ensemble models developed in this analysis are an extension of the framework used by Lu et al. in their study on stock price fluctuations on the Shanghai stock exchange [31]. Adjusting the input data structure to accommodate multiple parallel time-series data, parameter configuration at each layer, alteration of the training method parameters to match the characteristics of multiple parallel financial time-series data, and the inclusion of an LSTM layer to increase the prediction accuracy are all part of the update.

The structural diagram of the CNN-LSTM model is shown in Figure 3, where CNN and LSTM are the main components accompanied by an input layer, a 1-dimensional convolution layer (1D convolutional), a pooling layer, an LSTM (hidden) hidden layer, and layer full connection

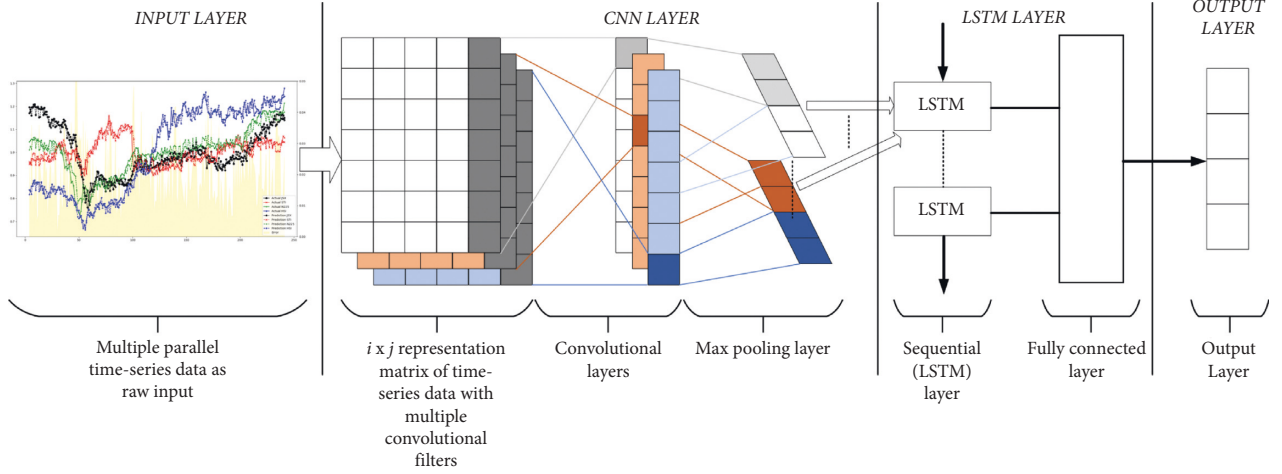


FIGURE 3: Architecture of the proposed multivariate CNN-LSTM for multiple parallel time-series prediction.

(full connection) which will issue the final result of the prediction.

**3.3. Multivariate CNN-LSTM Learning Process.** The stages of the training process and CNN-LSTM prediction on the time-series data are given as follows:

- (1) The training stage begins with the training data input process. At this stage, the process of entering the data used for CNN-LSTM training occurs. The next step is to initialize the network parameters to determine the weight and bias value (if any) at the beginning of each CNN-LSTM layer. Then, it is continued with the process at the CNN layer where the input data sequentially passes through the convolution layer and the pooling layer at the CNN layer, followed by the extraction process for the input data feature, and produces an output value that will be the input for the LSTM layer. At the CNN layer, the feature extraction process mainly occurs from the input time-series data.
- (2) Then, the output value from the CNN layer will enter the LSTM layer. In this LSTM layer, the prediction process mainly occurs in the observed time-series value, where the output value from the LSTM layer becomes the input for the full connection layer, which then produces the final predicted value. At this stage, the prediction training process is complete and then it is continued with the evaluation process of the training results where the error of the prediction results is calculated. The output value in the form of a prediction calculated by the output layer is compared with the actual value of the processed data group, and then the error value is calculated.
- (3) The results of the evaluation serve as a reference for determining whether the stopping conditions for training are met. In this case, the training stop condition is that the predetermined number of training cycles (epochs) is reached, and the

predictive error value is lower than a certain pre-determined threshold.

- (4) If based on the evaluation results, it is determined that the stop condition for training has not been achieved, then the calculated error value is propagated back to the previous layer and then adjusted the weights and bias values at each layer (back-propagation error) and return to the first step and repeat the training process. However, if any of the conditions of the stop condition for training are met, the training is completed, and the configuration of the entire CNN-LSTM network is saved.
- (5) The next stage is testing the CNN-LSTM model that has been trained using the test data. This process begins by entering the test data used for prediction or testing data input into the saved CNN-LSTM model and then getting the output value (prediction result) as the final output of the CNN-LSTM training and prediction process.
- (6) Measurement of the level of accuracy will be carried out by applying the calculation of the root mean square error (RMSE) value to see the amount of deviation between the actual value and the resulting predictive value.

## 4. Experiments Setting

**4.1. Financial Time-Series Data.** Multiple parallel financial time-series data were used in this study to represent an integrated structure in the stock exchange sector. The data is regular stock market indices gathered from four Asian exchanges: Shanghai, Japan, Singapore, and Indonesia, which spans 242 trading days from January 1, 2020, to December 31, 2020. Regular indices of the Shanghai, Japan, Singapore, and Indonesia exchanges will be predicted simultaneously using the historical values of the four observed series in this experiment. The data collection is split into two parts: a training set that includes the first 170 trading days and a test set that includes the last 72 trading days.

Since the four stock market indices have a wide range of values, the data collection is transformed, i.e., normalized as follows to help structure the training process and build an improved model:

$$y_i^j = \frac{x_i^j - \bar{x}^j}{s^j}, \quad (9)$$

where  $y_i^j$  is the standardized value of series  $j$ ,  $x_i^j$  is the original input data value of series  $j$ ,  $\bar{x}^j$  is the average of the input data value of series  $j$ , and  $s^j$  is the standard deviation of the input data of series  $j$ . A snapshot of the data set is outlined in Table 1.

**4.2. Model Evaluation.** The root mean square error (RMSE) is used as the estimation criterion of the method to test the forecasting impact of multivariate CNN-LSTM in addition to univariate CNN and LSTM that will perform the prediction in an individual manner:

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

where  $\hat{y}_i$  is the predicted value at a particular time-point  $i$  and  $y_i$  is the actual value. Since the difference between the forecast and the original value is less, a lower RSME value means higher prediction accuracy. The RMSE is measured for each sequence and compared for each model under consideration. During the preparation and testing phase, a comparative review will be carried out.

**4.3. Implementation of Multivariate CNN-LSTM.** Table 2 shows the multivariate CNN-LSTM parameter settings for this experiment. It can deduce that the basic model is designed as follows based on the CNN-LSTM network's parameter settings: the input training set data is a three-dimensional data vector (None, 4, 4), where the first number 4 represents the time step size and the other number 4 is the input dimensions of four properties, which in this case are four stock market indices of Shanghai, Japan, Singapore, and Indonesia.

The data is first fed into a one-dimensional convolution layer, which extracts more features and produces a three-dimensional output vector (None, 4, 64), with 64 being the size of the convolution layer filters. The vector then joins the pooling layer, where it is converted into a three-dimensional output vector (none, 4, 64). The output vector then goes to the first LSTM layer for training; two layers of LSTM are put into place in this proposed structure to improve the accuracy of multiple variable predictions; after training, the output data with shape (None, 100) goes to the second LSTM layer to get the output values; 100 is the number of hidden units in both LSTM layers. Accordingly, Figure 4 depicts the basic structure of the proposed multivariate CNN-LSTM model.

## 5. Results and Discussion

After training CNN, LSTM, and multivariate CNN-LSTM with the processed training set data, the model is used to forecast the test set data, and the actual value is compared with the expected value for both phases, as seen in Figures 5 and 6. For the record, the CNN and LSTM models are trained separately using univariate analysis, while the multivariate CNN-LSTM is trained using multivariate analysis to achieve simultaneous multiple parallel time-series prediction.

Figure 5 displays graphics of the comparison between the predicted index values of the four stock markets observed with the original value at the training stage, while Figure 6 displays comparison charts at the testing stage. In general, from the two figures, it can be observed that the three models tested have the ability to predict the movement of the stock market index with a fairly good level of accuracy. However, more in-depth observations show that the graph of the prediction results generated by the LSTM model is better than that produced by the CNN model, and the graph of the prediction results from the multivariate CNN-LSTM model has the best results among the three. This happened consistently both at the training and testing stages as well as in the four stock markets.

Observing the yellow field on the graph can also be viewed as a guideline that the CNN-LSTM multivariate model does significantly more than the other two versions. The magnitude of the forecast error rate at any point in time is shown by the yellow region of the inn. As a result, it can be inferred that the smaller the field, the lower the cumulative error of prediction. Both Figures 5 and 6 demonstrate that the CNN-LSTM multivariate model's yellow region on the graph is smaller than the CNN and LSTM versions at both the training and testing levels. This supports the finding that of the three models studied, the CNN-LSTM multivariate model, which is an ensemble of the CNN and LSTM models, performs the best. In addition, the graph shows that the average error of the forecast outcomes is within a reasonable range.

In addition, given that the processed time-series data were in the coverage of the COVID-19 pandemic period, where the movements of various economic indicators became more uncertain, the CNN-LSTM multivariate model was also confirmed to have the ability to predict the value of financial time-series data with better performance. The largest difference between the predicted results and the original value occurs in the range of the 55<sup>th</sup> trading day, which, if mapped to a calendar day, falls in early March 2020 when the COVID-19 pandemic begins to affect global economic conditions. However, after that, it appears that the value of the prediction error tends to decrease, which indicates the ability of the CNN-LSTM multivariate model to adapt to changes in the movement pattern of the stock market index value.

Paying more attention to the predictions of the four stock market indices made by the ensemble of the CNN-LSTM model, it is clearly seen that the proposed model can predict the movement of all four stock market indices with a

TABLE 1: Snapshots of daily stock market indices used as multiple financial time-series data.

Date	Shanghai		Japan		Singapore		Indonesia	
	Actual	Norm	Actual	Norm	Actual	Norm	Actual	Norm
02/01/2020	252.35	0.8085	23204.86	1.0221	8.49	0.9801	6283.58	1.1971
03/01/2020	258.36	0.8278	23575.72	1.0384	8.30	0.9580	6323.46	1.2047
06/01/2020	260.30	0.8340	23204.76	1.0221	8.25	0.9524	6257.40	1.1921
07/01/2020	261.46	0.8377	23739.87	1.0457	8.31	0.9591	6279.34	1.1963
08/01/2020	259.91	0.8328	23850.57	1.0505	8.22	0.9491	6225.68	1.1861
09/01/2020	266.31	0.8533	24025.17	1.0582	8.43	0.9735	6274.49	1.1954
10/01/2020	266.50	0.8539	23916.58	1.0535	8.42	0.9723	6274.94	1.1955
13/01/2020	271.54	0.8700	23933.13	1.0542	8.36	0.9657	6296.56	1.1996
14/01/2020	269.02	0.8619	24041.26	1.0589	8.36	0.9657	6325.40	1.2051
15/01/2020	271.15	0.8688	24083.51	1.0608	8.35	0.9646	6283.36	1.1971

TABLE 2: Parameters configuration for the proposed multivariate CNN-LSTM model.

Parameters	Values	Layer
Convolutional layer filters	64	CNN
Convolutional layer kernel_size	2	CNN
Convolutional layer activation function	Relu	CNN
Convolutional layer padding	Valid	CNN
Pooling layer pool_size	2	CNN
Number of LSTM layer 1 hidden unit	100	LSTM
Number of LSTM layer 2 hidden unit	100	LSTM
LSTM layer 1 and 2 activation function	Relu	LSTM
Time step	4	LSTM
Optimizer	Adam	Model fitting
Loss function	MAE	Model fitting
Epochs	1000	Model fitting

high degree of accuracy. As the actual values of the four stock market indices is showing an increasing trend towards the end of the year 2020, the predictions made by the proposed model are demonstrating a similar manner as well. Therefore, these predictions can be utilized to help to make an investment decision, which in this case suggests that amid the COVID-19 pandemic period investing in the stock market can be considered worthwhile. This argument is in line with suggestions made by a financial analyst regarding some investing lessons from the pandemic, which states that (1) buy and hold could not have been truer than the past year; (2) the best time to invest is now, i.e., the time when COVID-19 vaccines become available; (3) the market is recovering gradually as the countries around the world are starting to have a grasp on COVID-19 pandemic [38].

Regarding the prediction's quality assessment, as previously stated, in this study, the performance of the three time-series data prediction models was also evaluated by calculating the RMSE value. RMSE calculations are carried out both at the training stage and at the testing stage as well. Additionally, to prove that the accuracy of financial time-series data predictions can be improved by building a CNN-LSTM ensemble structure that utilizes multivariate analysis techniques, the trial of the index prediction of the four stock markets using the CNN model and the LSTM model is carried out individually and based on analytical techniques univariate.

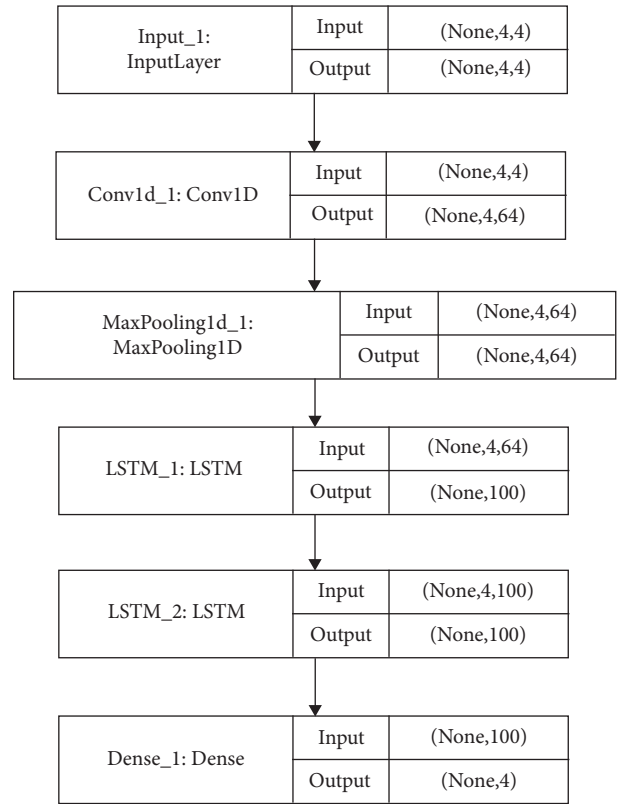


FIGURE 4: Basic structure of the proposed multivariate CNN-LSTM model for multiple time-series prediction with 4 features and 4-time steps for the prediction process.

The comparison of RMSE values for each model and for each stage is shown in Figures 7 and 8 and details are outlined in Table 3. From the two graphs, it can be learned that the RMSE value for the CNN-LSTM multivariate model is the smallest in all indices, both at the training and testing stages. Thus, this fact confirms the theory, which states that combining various features and advantages possessed by different models or algorithms into an ensemble and using multivariate analysis techniques can provide better results in solving a problem, which in this case is to predict—index values of four stock markets in the Asian region.

Additionally, evaluation of predictions quality made by the proposed multivariate CNN-LSTM is also conducted by



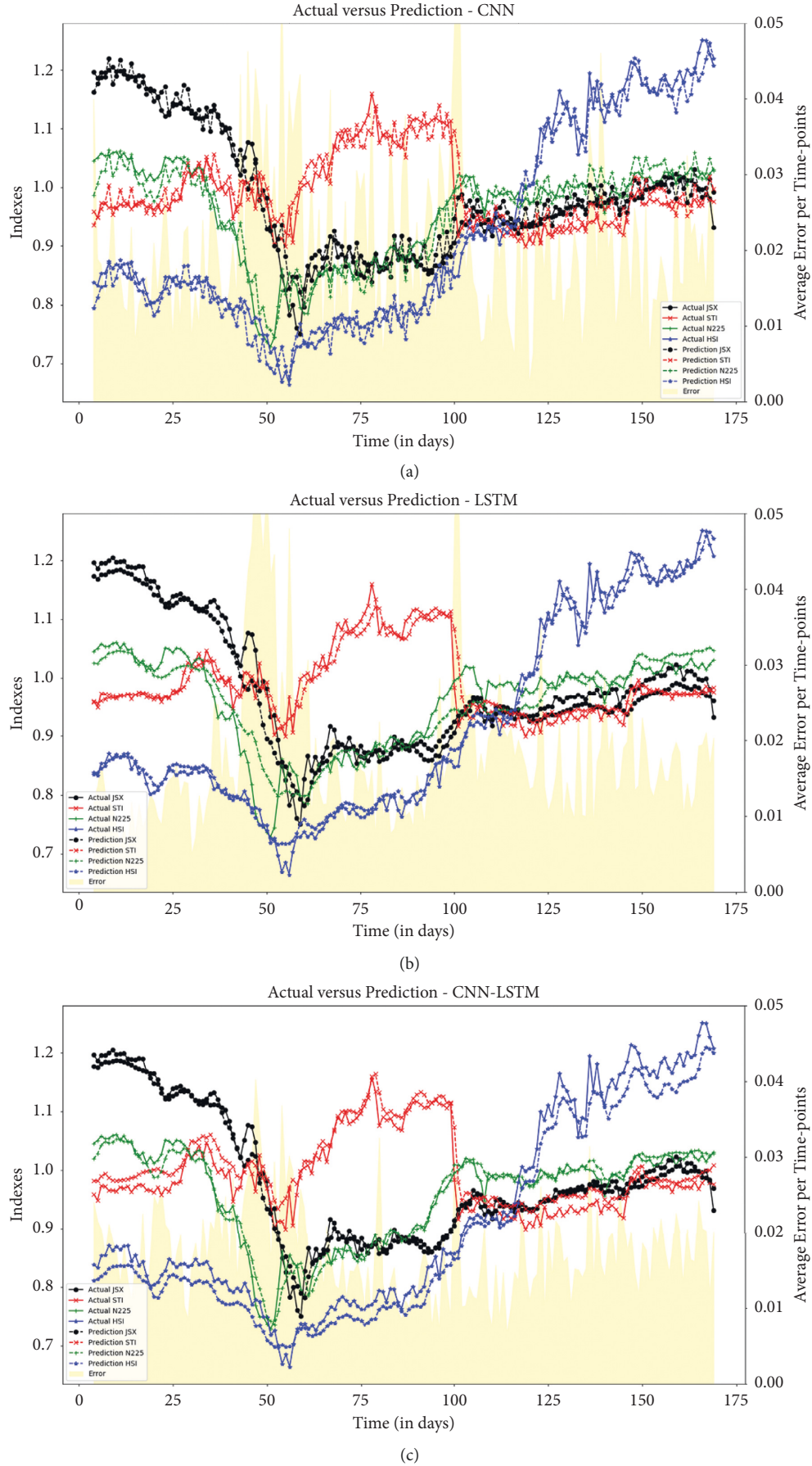


FIGURE 5: Comparison of the actual (solid line) and predicted values (dashed line) in the training phase (first 170 trading days of the year 2020): (a) by CNN Model, (b) by LSTM model, and (c) by the ensemble of CNN-LSTM.

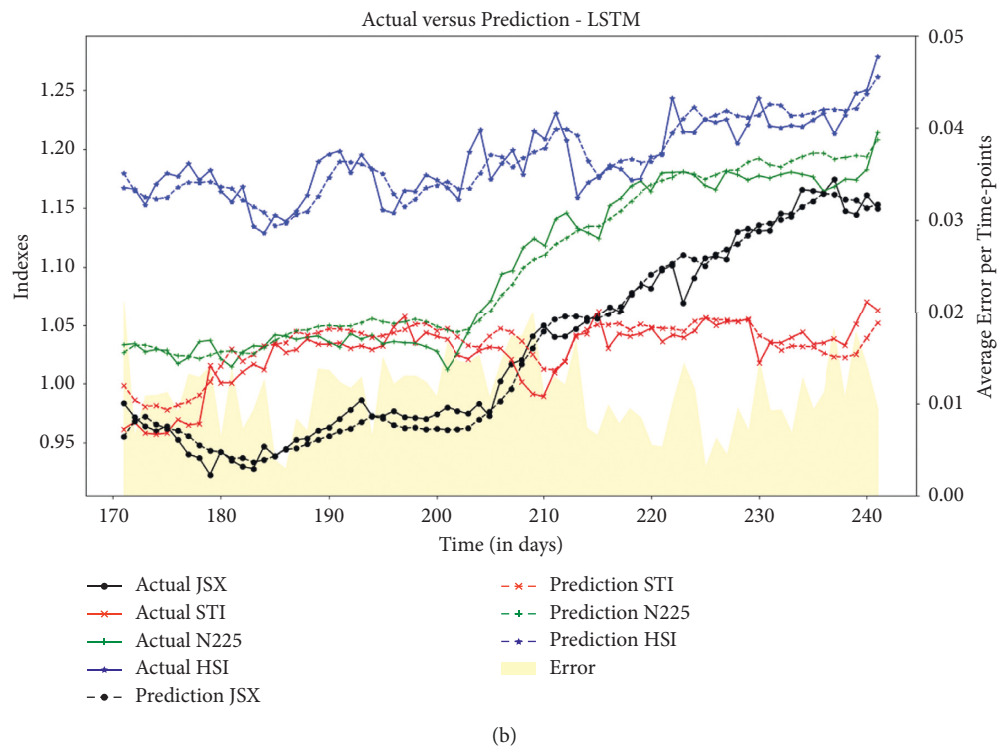
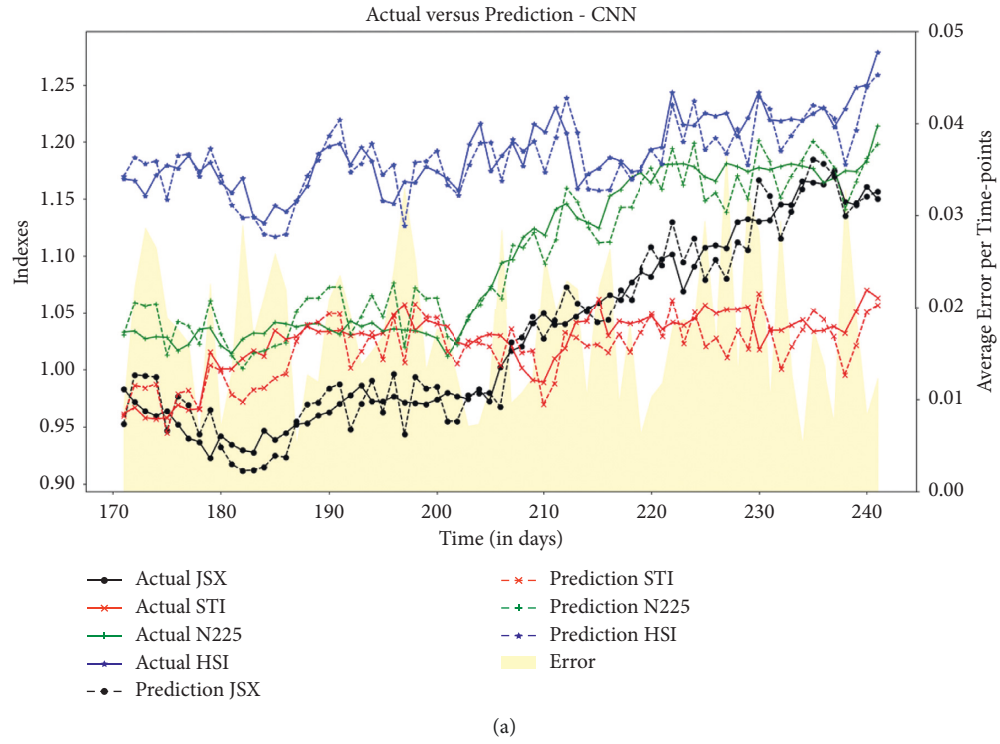


FIGURE 6: Continued.

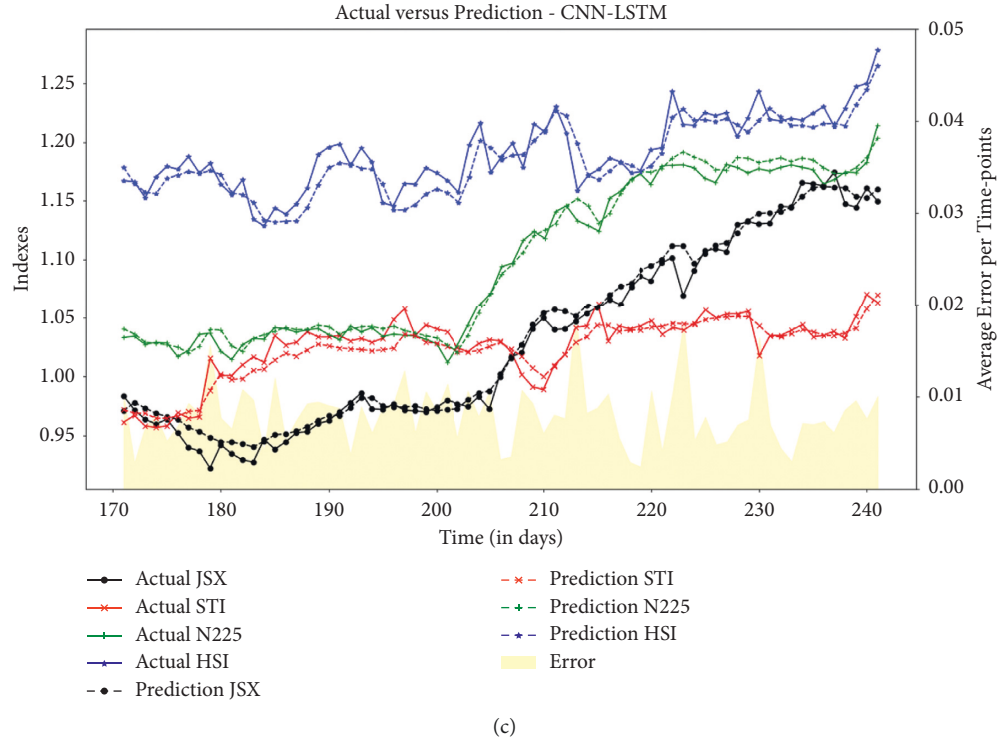


FIGURE 6: Comparison of the actual (solid line) and predicted values (dashed line) in the testing phase (last 72 trading days of the year 2020): (a) by CNN Model, (b) by LSTM model, and (c) by the ensemble of CNN-LSTM.

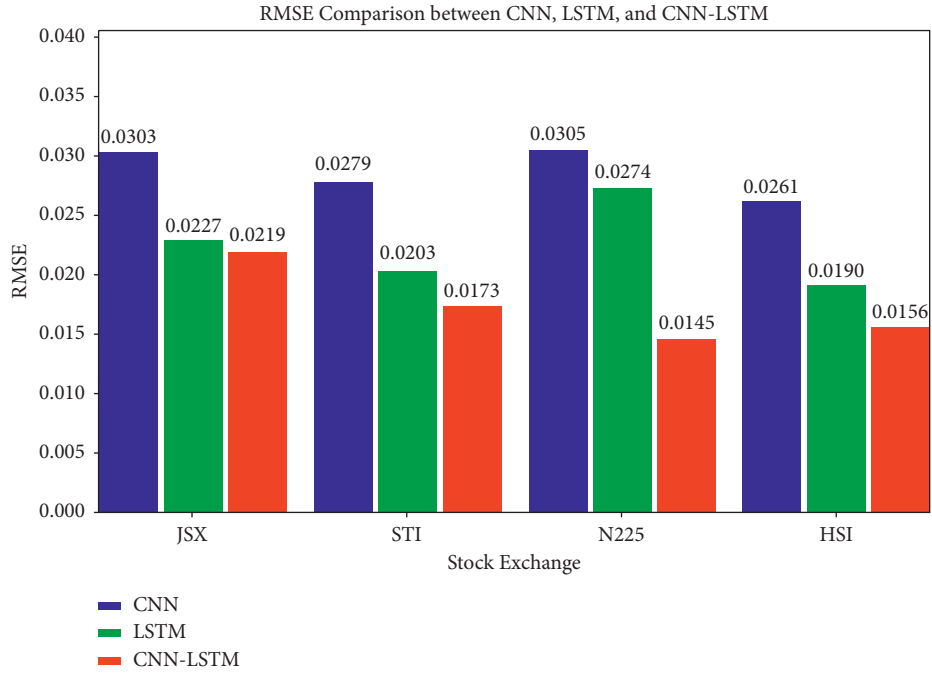


FIGURE 7: RMSE comparison between CNN, LSTM, and multivariate CNN-LSTM when predicting data in the training phase, i.e., the first 170 trading days of the year 2020.

comparing their descriptive statistics values with the actual values. The basic premise of such a comparison is that similar descriptive statistics values between series suggest

that the data set is comparable as their basic nature is analogous. Descriptive statistics between actual and predicted values in the testing phase are given in Table 4.

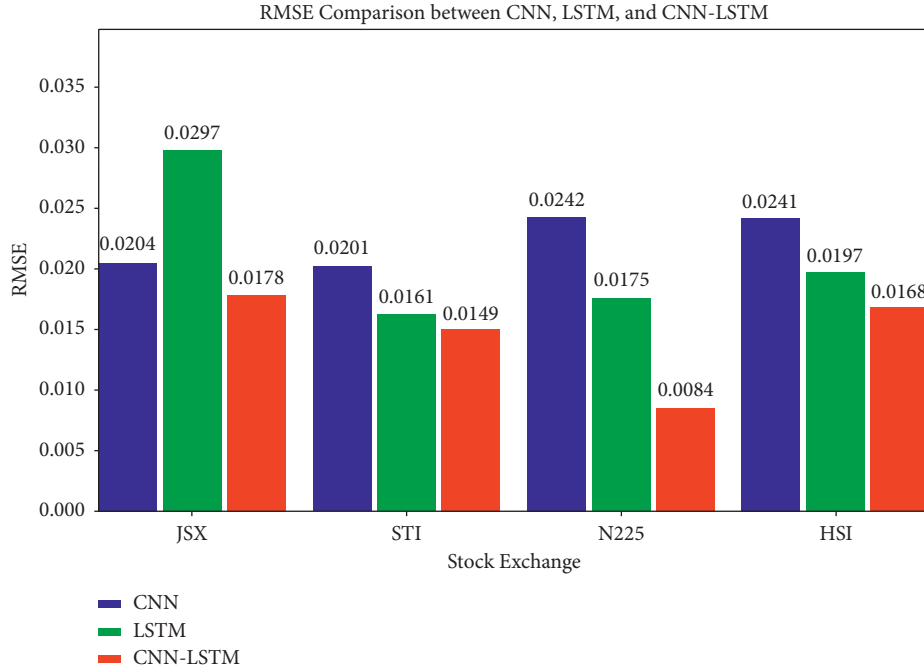


FIGURE 8: RMSE comparison between CNN, LSTM, and multivariate CNN-LSTM when predicting data in the testing phase, i.e., the last 72 trading days of the year 2020.

TABLE 3: Comparison of RMSE values between CNN, LSTM, and multivariate CNN-LSTM during training and testing phase.

Model	Shanghai (HSI)		Japan (N225)		Singapore (STI)		Indonesia (JSX)	
	Train	Test	Train	Test	Train	Test	Train	Test
CNN	0.0261	0.0241	0.0305	0.0242	0.0279	0.0201	0.0303	0.0204
LSTM	0.0190	0.0197	0.0274	0.0175	0.0203	0.0161	0.0227	0.0297
CNN-LSTM	0.0156	0.0168	0.0145	0.0084	0.0173	0.0149	0.0219	0.0178

TABLE 4: Comparison of descriptive statistics between actual and predicted time-series data produced by the proposed multivariate CNN-LSTM model in testing period, i.e., last 72 trading days of 2020. Note: Act = actual and Pred = predictions.

Descriptive statistics	Hang Seng (HSI)		Japan (N225)		Singapore (STI)		Indonesia (JSX)	
	Act	Pred	Act	Pred	Act	Pred	Act	Pred
Mean	1.1917	1.1802	1.0997	1.0853	1.0260	1.0221	1.0315	1.0421
Median	1.1872	1.1783	1.0956	1.0847	1.0336	1.0401	1.0093	1.0102
Standard deviation	0.0320	0.0296	0.0677	0.0643	0.0277	0.0291	0.0787	0.0766
Sample variance	0.0010	0.0012	0.0046	0.0043	0.0008	0.0071	0.0062	0.0058
Kurtosis	0.3295	0.3187	1.7725	1.7523	0.8161	0.8021	1.3399	1.3187
Skewness	0.3376	0.3152	0.1284	0.1179	1.1910	1.1876	0.3793	0.3685
Range	0.1497	0.1502	0.2022	0.2013	0.1132	0.1045	0.2520	0.2471
Minimum	1.1288	1.1256	1.0121	1.0097	0.9570	0.9613	0.9226	0.9197
Maximum	1.2786	1.2691	1.2143	1.2095	1.0702	1.0273	1.1746	1.1637
Confidence level (95.0%)	0.0075	0.0086	0.0159	0.0137	0.0065	0.0061	0.0185	0.0157

Comparative analysis between the descriptive statistics of actual and predicted values of four observed stock market indices as outlined in Table 4 indicates that both data set retain considerably similar values. This result confirms that both actual and predicted data set have similar basic nature and therefore can be concluded that the proposed multivariate CNN-LSTM model is capable of forecasting future values of financial time-series not only with good accuracy, in terms of relatively small RMSE values, but also with truthful basic nature.

## 6. Conclusions

This research proposes a multivariate CNN-LSTM model to forecast the value of multiple parallel financial time-series one stage in time based on the characteristics of the stock market index regular value time-series results (the next day). The technique used is multivariate time-series data forecasting, in which several time-series are predicted simultaneously by considering the condition of all observable



series. CNN extracts features from the input data in the model, while LSTM studies the derived function data and performs the final step of estimating the performance of the stock market index the following day. This research uses applicable data from four Asian stock exchanges as training and test data to validate the experimental findings, namely, Shanghai, Japan, Singapore, and Indonesia. When compared with individual CNN and LSTM models, the experimental findings reveal that multivariate CNN-LSTM has the highest predictive precision and better efficiency (smallest RMSE value). This finding supports the assumption that incorporating relationships between variables into a prediction model will help with the multiple time-series problems of forecasting parallel movements of a set of time-sensitive variables that are related. As a result, multivariate CNN-LSTM can be used to predict the value of various stock market indices and can serve as a useful tool for investors when making investment decisions. Aside from that, multivariate CNN-LSTM is a viable option for research involving the construction of models for financial time-series data analysis. However, the existing model has a few flaws, including the fact that different data relating to external variables such as public opinion and national policy were not considered during the prediction period. In this regard, the future study work plan will focus on using more factors, both quantitative and qualitative in nature, as input into the prediction model and constructing a fully working investment-trading system based on the proposed model as well.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was funded by the Ministry of Research and Technology, Republic of Indonesia, under the grant for the Basic Research scheme year 2021.

## References

- [1] R. Vanaga and B. Sloka, "Financial and capital market commission financing: aspects and challenges," *Journal of Logistics, Informatics and Service Science*, vol. 7, no. 1, pp. 17–30, 2020.
- [2] L. Zhang and H. Kim, "The influence of financial service characteristics on use intention through customer satisfaction with mobile fintech," *Journal of System and Management Sciences*, vol. 10, no. 2, pp. 82–94, 2020.
- [3] L. Badea, V. Ionescu, and A.-A. Guzun, "What is the causal relationship between stoxx europe 600 sectors? but between large firms and small firms?" *Economic Computation and Economic Cybernetics Studies and Research*, vol. 53, no. 3, pp. 5–20, 2019.
- [4] W. T. D. Sousa Junior, J. A. B. Montevecchi, R. D. C. Miranda, F. Rocha, and F. F. Vilela, "Economic lot-size using machine learning, parallelism, metaheuristic and simulation," *International Journal of Simulation Modelling*, vol. 18, no. 2, pp. 205–216, 2019.
- [5] A. Coser, M. M. Maer-Matei, and C. Albu, "Predictive models for loan default risk assessment," *Economic Computation and Economic Cybernetics Studies and Research*, vol. 53, no. 2, pp. 149–165, 2019.
- [6] R. Qiao, "Stock prediction model based on neural network," *Operations Research and Management Science*, vol. 28, no. 10, pp. 132–140, 2019.
- [7] C. Jung and R. Boyd, "Forecasting UK stock prices," *Applied Financial Economics*, vol. 6, no. 3, pp. 279–286, 1996.
- [8] B. S. Kim and T. G. Kim, "Cooperation of simulation and data model for performance analysis of complex systems," *International Journal of Simulation Modelling*, vol. 18, no. 4, pp. 608–619, 2019.
- [9] Y. Sun, Y. Liang, and W. Zhang, "Optimal partition algorithm of the RBF neural network and its application to financial time-series forecasting," *Neural Computing and Applications*, vol. 14, pp. 1441–1449, 2005.
- [10] Q. Yang and C. Wang, "A study on forecast of global stock indices based on deep LSTM neural network," *Statistical Research*, vol. 36, no. 6, pp. 65–77, 2019.
- [11] K.-S. Moon and H. Kim, "Performance of deep learning in prediction of stock market volatility," *Economic Computation and Economic Cybernetics Studies and Research*, vol. 53, no. 2, pp. 77–92, 2019.
- [12] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 1–199, 2014.
- [13] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [14] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [16] T. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8614–8618, Vancouver, Canada, May 2013.
- [17] E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019.
- [18] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3642–3649, Providence, RI, USA, June 2012.
- [19] M. L. Brocardo, I. Traore, I. Wonggang, and M. S. Obaidat, "Authorship verification using deep belief network systems," *International Journal of Communication Systems*, vol. 30, no. 12, p. e3259, 2017.
- [20] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [21] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," *Proceedings of the 51<sup>st</sup> Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 455–465, 2013.

- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] H. Liu and Z. Long, "An improved deep learning model for predicting stock market price time-series," *Digital Signal Processing*, vol. 102, p. 102741, 2020.
- [24] S. Y. Shih, F. K. Sun, and H. Y. Lee, "Temporal pattern attention for multivariate time-series forecasting," *Machine Learning*, vol. 108, no. 8, pp. 1421–1441, 2019.
- [25] X. Yuan, L. Li, and Y. Wang, "Nonlinear dynamic soft sensor modeling with supervised long short-term memory network," *IEEE transactions on industrial informatics*, vol. 16, no. 5, pp. 3168–3176, 2019.
- [26] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market," *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020.
- [27] A. H. Marblestone, G. Wayne, and K. P. Kording, "Toward an integration of deep learning and neuroscience," *Frontiers in Computational Neuroscience*, vol. 10, p. 94, 2016.
- [28] J. H. Stock and M. W. Watson, "Twenty years of time series econometrics in ten pictures," *The Journal of Economic Perspectives*, vol. 31, no. 2, pp. 59–86, 2017.
- [29] Y. Chen and Y. Hao, "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction," *Expert Systems with Applications*, vol. 80, pp. 340–355, 2017.
- [30] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187–205, 2017.
- [31] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," *Complexity*, vol. 2020, Article ID 6622927, 2020.
- [32] X. Zhong and D. Enke, "Predicting the daily return direction of the stock market using hybrid machine learning algorithms," *Financial Innovation*, vol. 5, no. 24, 2019.
- [33] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. Shahab, "Deep learning for stock market prediction," *Entropy*, vol. 22, no. 8, p. 840, 2020.
- [34] F. Kamalov, "Forecasting significant stock price changes using neural networks," *Neural Computing and Applications*, vol. 32, no. 2, pp. 17655–17667, 2020.
- [35] A. Mahmoud and A. Mohammed, "A survey on deep learning for time-series forecasting," *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*, Springer, Cham, Switzerland, pp. 365–392, 2021.
- [36] N. C. Petersen, F. Rodrigues, F. C. Pereira, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Systems with Applications*, vol. 120, pp. 426–435, 2019.
- [37] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.
- [38] D. Tashanova, A. Sekerbay, D. Chen et al., "Investment opportunities and strategies in an era of coronavirus pandemic," *SSRN Electronic Journal*, p. 3567445, 2020.

Copyright © 2021 Harya Widiputra et al. This is an open access article distributed under the Creative Commons Attribution License (the “License”), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. <https://creativecommons.org/licenses/by/4.0/>