



# A CNN-LSTM Stock Prediction Model Based on Genetic Algorithm Optimization

Heon Baek<sup>1</sup>

Accepted: 26 May 2023 / Published online: 17 June 2023

© The Author(s), under exclusive licence to Springer Japan KK, part of Springer Nature 2023

## Abstract

Predicting the stock market remains a difficult field because of its inherent volatility. With the development of artificial intelligence, research using deep learning for stock price prediction is increasing, but the importance of applying a prediction system consisting of preparing verified data and selecting an optimal feature set is lacking. Accordingly, this study proposes a GA optimization-based deep learning technique (CNN-LSTM) that predicts the next day's closing price based on an artificial intelligence model to more accurately predict future stock values. In this study, CNN extracts features related to stock price prediction, and LSTM reflects the long-term history process of input time series data. Basic stock price data and technical indicator data for the last 20 days prepare a data set to predict the next day's closing price, and then a CNN-LSTM hybrid model is set. In order to apply the optimal parameters of this model, GA was used in combination. The Korea Stock Index (KOSPI) data was selected for model evaluation. Experimental results showed that GA-based CNN-LSTM has higher prediction accuracy than single CNN, LSTM models, and CNN-LSTM model. This study helps investors and policy makers who want to use stock price fluctuations as more accurate predictive data using deep learning models.

**Keywords** Stock market prediction · Convolutional neural networks · Long short-term memory · Genetic algorithm

## 1 Introduction

Stock price prediction has emerged as a very important problem in the economic field. However, it is difficult to predict the stock market because stock price prediction is highly uncertain and highly volatile, influenced by many factors, both internal and external, such as the domestic and foreign economic environment, industrial outlook, government policy, and investor sentiment.

---

✉ Heon Baek  
huny01017@hanmail.net

<sup>1</sup> Sogang University, 35 Baekbeom-ro, Mapo-gu, Seoul, Republic of Korea

In general, stock price prediction methods can be largely divided into fundamental analysis and technical analysis. Fundamental Analysis is a method that pays more attention to the intrinsic value of stocks, such as inflation, exchange rate, and interest rate. Related studies suggest investment strategies using stock financial data or present simulation results that show higher profitability than average stock price returns through portfolio analysis (Gu & Jangn 2012; Baresa et al., 2013). On the other hand, Technical Analysis predicts stock price only based on the past flow of stock price and trading volume. It is assumed that stock investor's psychology and market environment are all reflected in this flow, and it analyzes indicators to find a certain pattern or trend and predict stock price. Researches related to this have developed predictive modeling using basic indicators such as stock closing price, market price, and trading volume (Long et al., 2019; Rezaei et al., 2021), and verified the effect of technical analysis by reflecting technical indicators using moving averages (Detry & Gregoire, 2001; de Souza et al., 2018). There are studies that suggest predictive modeling based on technical indicators (Chung & Shin, 2018; Lee, 2022). Although this method of analysis is still commonly used by many organizations and individual investors, in the case of fundamental analysis, it may not be persuasive because the prediction results depend on the professional qualifications of the analyst. In addition, traditional time series analysis methods have limitations in their further application or extension due to the uncertainty and noise of financial time series and the tendency for variable relationships to change dynamically over time. Therefore, the accuracy of using only the traditional analytical model is questioned (Yang & Wang, 2019).

However, recent stock price prediction using artificial intelligence has shown good performance, and related research has been continuously conducted. In order to improve the stock price prediction rate, a combination of simple basic indicators and technical indicators was used as a data set, or the extracted prediction factors were presented as experimental results by increasing the accuracy of the model using various algorithms. This proves the superiority of prediction performance using artificial intelligence compared to traditional time series analysis methods. The recent direction of development is the trend of improving by using various optimal algorithms or combining the advantages of algorithms and applying them to predict trends through hybrid methods using the latest algorithms.

However, the hybrid model has higher model complexity than the single model, and the number of parameters to be considered is large. Since the overall performance may be degraded due to the parameters that are not optimized, the performance of the model may vary depending on how the parameters are applied.

This study proposes a hybrid model of CNN (Convolutional Neural Networks) and LSTM (Long Short-Term Memory) to which a parameter optimization algorithm is applied for stock price prediction. A model was created that combines the advantages of CNN, which shows good performance in effectively extracting features from data, and the advantages of LSTM, which can sequentially arrange data from time series data and find data interdependencies. For parameter optimization of CNN and LSTM, GA (genetic algorithm) technique was applied. GA helps to select parameters scientifically. In other words, this paper proposes a hybrid model that

integrates deep learning algorithms and GA to find a model suitable for predicting the stock market's next-day closing price.

The rest of this study is as follows. Chapter 2 describes related research. Section 3 describes the proposed stock price prediction model in detail. Section 4 describes the experimental results of the model. Finally, Section 5 presents conclusions and future research topics.

## 2 Literature Review

Stock market forecasting is a complex and challenging task due to its high level of uncertainty and non-stationarity (Abu-Mostafa & Atiya, 1996). However, as several empirical studies have proven that stock market prediction is possible to some extent, research and experiments using various analysis methods are still being conducted. In particular, with the advent of artificial intelligence, various techniques used in stock price prediction are being explored using not only machine learning but also deep learning methodology, a type of machine learning. Machine learning technology shows superior performance compared to traditional statistical models because it has an excellent ability to learn hidden relationships between market factors and identify complex patterns in data (Kim & Han, 2000).

Relevant studies developing in this field are reviewed as follows. Chen and Hao (2017) proposed a hybrid framework that can improve predictive ability using a feature weighted support vector machine and a feature weighted K-nearest neighbor through experimental results on two well-known Chinese stock market indices (Chen & Hao, 2017). Sharma and Juneja (2017) propose a tree ensemble in a random forest using LSBoost to predict stock market indicators based on stock price data for the past 10 years. Technical indicators were used as input variables of the prediction model, and the prediction performance of the proposed model was compared with SVR (Support Vector Regression). As a result, it was found that the proposed model showed better predictive performance than SVR (Sharma and Juneja 2017). Zhang and Bai (2019) selected the top 10 stocks most likely to exceed the median return in the CSI300 index to compare the performance of various machine learning algorithms, and machine learning of Random Forest, Artificial Neural Network (ANN) and Logistic Regression. As a result of evaluating, the random forest achieved the highest accuracy. Ye et al. (2021) proposed a stock prediction model based on LightGBM, and the proposed model showed higher returns and superior predictive power than the XGBoost model and Neural Network (Ye et al., 2021). Yang et al. (2018) showed the feasibility and suitability of time series data prediction applied to feature extraction for stock index prediction based on CNN structure (Yang et al., 2018), Wu et al. (2020) compared SVM and traditional neural network with CNN to improve stock trading prediction accuracy, and showed that CNN using multi-filter has the highest stock price prediction efficiency (Wu et al., 2020). Mezghani and Abbas (2022) presented a model that predicts GCC financial stress based on 1D-CNN (One-Dimensional Convolutional Neural Network) (Mezghani & Abbas, 2022). Yan et al. (2021) compared back propagation neural network, recurrent neural network (RNN), and LSTM to model and predict financial transaction

data of Shanghai. It was shown that the stock market can be effectively predicted (Yan et al., 2021). Bhambu (2023) compared deep learning algorithms such as RNN, LSTM, Gated recurrent unit (GRU), and Bi-LSTM (Bi-directional Long Short-Term Memory) models for short-term and long-term time series prediction (Bhambu, 2023).

In addition, until recently, the trend of research on stock price prediction by combining and applying the characteristics of various models has been maintained. Yang et al. (2021) proposed a combined model using XGBoost and LightGBM for stock price prediction, which showed better predictive performance than single model and neural network model (Yang et al., 2021). Shah et al. (2018) applied DNN and LSTM to stock market price prediction, and LSTM, rather than deep neural network (DNN), showed the possibility of finding the underlying trend and making long-term predictions for stock datasets with high volatility (Shah et al., 2018). Chung and Shin (2018) proposed a hybrid method that integrates LSTM and GA, which showed better prediction performance than the existing LSTM model (Chung & Shin, 2018). Lv et al. (2021) showed that the LightGBM and LSTM hybrid model had the best prediction accuracy compared to RNN and GRU (Lv et al., 2021). Yuan et al. (2023) proposed a model combining a temporal convolutional neural network with a multi-scale attention mechanism (MA-TCN) and Multi-View Convolutional-Bidirectional Recurrent Neural Network with Temporal Attention (MVCNN-BiLSTM-Att) to predict the Chinese stock market during COVID-19 (Yuan et al., 2023).

In addition, GA, an optimization algorithm, has been widely applied to machine learning and deep learning algorithms. It is used as an integrated model approach to optimize analysis or train neural networks. Kai and Wenhua (1997) proposed a combined GA and ANN model for stock price prediction, and Yun et al. (2021) proposed a GA-XGBoost hybrid model to improve the performance of the model (Kai & Wenhua, 1997; Yun et al., 2021). This study proposes an improved deep learning prediction system through optimized parameter adjustment and a combined model of CNN and LSTM to define the data combination to be analyzed using GA.

### 3 Methodology

#### 3.1 Data

In this study, a stock price analysis server was built to perform data collection, preprocessing, analysis, and result aggregation in a container-based virtual environment. This study collects data using Yahoo financial version 1.6, a library designed to request/receive data from the API provided by Yahoo Finance. Analysis data is daily stock price data from 2014/9/14 to 2022/10/31. For the indicators collected in this study, the basic indicators of KOSPI, Korea's composite stock index, opening price, highest price, lowest price, closing price and volume, and major technical indicators were used for analysis. Technical indicators are statistical calculations based on variables such as price or other technical indicators. Active traders or technical analysts use technical indicators to analyze short-term or long-term price movements, so these indicators are widely used in the market.

The technical indicators used in this study are Simple Moving Average (SMA), Weighted Moving Average (WMA), Relative Strength Index (RSI), Stochastic K%, Stochastic D%, and Moving Average Convergence Divergence (MACD).

SMA is a technical indicator that digitizes the direction of a stock price for a specific period based on the closing price, and means a moving average of  $n$  days from a specific day ( $t$ ). WMA assigns more weight to the latest data points because the latest data points contain more relevant information than older data points. The SMA and WMA periods in this study were 5, 10, 20, 60, and 120 days. RSI is an indicator that can measure the relative trend by expressing the strength of the current stock price trend as a percentage (Chen & Hao, 2017). The most commonly used period of the RSI is 14 days, which was also adopted in this study. A stochastic oscillator represents the close position of a stock in relation to a range of high and low prices over a period of time. There are two types of stochastic oscillators: stochastic K% and stochastic D%, and D%, is a 3-day moving average of K% (Chung & Shin, 2018). MACD is an indicator that can identify the trend of stock prices by using the difference between the short-term moving average and the long-term moving average. MACD is a momentum oscillator calculated by subtracting the 26-day EMA (exponential moving average) from the 12-day EMA (Murphy, 1999). Traders use this to determine the trend, direction, momentum, and potential reversal of stock prices (Bhandari et al., 2022). MACD is observed for signal lines. Analysis of the intersection of the signal line and the MACD helps to better understand whether the market is inherently bullish or bearish. Signal line is the exponential moving average of the MACD line itself over a period of time. The most common period for counting signal lines is 9, which averages the MACD line itself, so the signal line is smoother than the MACD line. The histogram is plotted to show the difference between the MACD line and the signal line. It's great component used to identify trends. The histogram is calculated by subtracting the signal line from the MACD line (Table 1).

**Table 1** Technical indicators

| Technical indicators                         | Formula   |
|--|---|
| Simple moving average(SMA)                   | $SMA = \frac{C_t + C_{t-1} + \dots + C_{t-n}}{n}$   |
| Weighted moving average(WMA)                 | $WMA = \frac{((n) \times C_t + (n-1) \times C_{t-1} + \dots + C_n)}{(n + (n-1) + \dots + 1)}$ |
| Relative Strength Index (RSI)                | $RSI = 100 - \frac{100}{1 + (\sum_{i=0}^{n-1} Up_{i-1}/n) / (\sum_{i=0}^{n-1} Dw_{i-1}/n)}$   |
| Stochastic K%                                | $\text{Stochastic K\%} = \frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100$               |
| Stochastic D%                                | $\text{Stochastic D\%} = \frac{\sum_{i=0}^{n-1} K_{t-i}\%}{n}$                                |
| Moving Average Convergence Divergence (MACD) | $MACD = \sum_{i=t-9}^t (EMA(12)_t - EMA(26)_t)$   |

## 3.2 Algorithm

### 3.2.1 CNN (Convolutional Neural Networks)

CNN was first proposed by Fukushima (1988), but was not widely used due to limitations in computational hardware for network training. In the 1990s, LeCun et al. applied a gradient-based learning algorithm to CNN, and since then it shows good performance in image processing and natural language processing, and can be effectively applied to time series prediction (Fukushima, 1988; LeCun et al., 1998). In general, CNNs are composed of layers, and each layer is composed of several layers to form the entire CNN. CNN has a structure that outputs the characteristic result by calculating the input received from the previous layer with a matrix-shaped filter. This has the advantage of being able to extract multiple features from one input through multiple filters, and has been mainly used in image data analysis studies consisting of complex and many values (Szegedy et al., 2015; Krizhevsky et al., 2017). While the application for visual image analysis is operated through a filter in the form of a two-dimensional matrix, a one-dimensional convolutional neural network is used to extract the temporal characteristics of time series data (Liu et al., 2019; Livieris et al., 2020). Conv1D convolves one-dimensional signals to identify basic rules in sequence data, thereby extracting important features from sequence data and reducing the dimensionality of sequences. Equation (1) is a convolution calculation formula. Here,  $F(x)$  means a function for calculating convolution, and  $F(x)$  and  $g(x)$  represent an input and a convolution kernel, respectively.  $x$  is the kernel size and  $\tau$  represents the data index of the sequence.

$$F(x) = (f \times g)(x) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(x - \tau) \quad (1)$$

Dilated Convolutions are mainly used for semantic cutting. Compared to regular convolution, dilated convolution adds a dilation rate parameter to indicate the size of the dilation. Dilated convolution differs from normal convolution in that the size of the convolution kernel is the same, but a larger receptive field is obtained due to the existence of the dilation rate (Sercu & Goel, 2016). The receptive field is the range over which the convolutional kernel can operate on the original information. Equation (2) is the convolution kernel size calculation formula for Dilated Convolutions. Where  $k_{size}$  is the convolution kernel size after expansion,  $rdilation$  is the expansion rate, and  $o_{size}$  is the original size of the defined convolution kernel.

$$k_{size} = rdilation \times (o_{size} - 1) + 1 \quad (2)$$

After the convolution layer, the activation function layer is generally located. As the activation function, relu, sigmoid, elu, etc. are used. The activation function plays a role in increasing nonlinearity in the CNN structure. In a fully connected layer, all weight filters operate on each and every input. Among the CNN layers, the closer to the input data tends to learn general features, and the later layers learn specific data features. In general, each layer has several nodes, and when learning data from an input node to the next node, the signal between the two nodes is calculated

and output as a single value through weight calculation and activation function. Features are extracted through convolution from the image values entered into the normal layer, and as these are accumulated in multiple layers, higher-level features are extracted as the higher layers go. The top fully connected layer and the softmax layer calculate the probability of each label using the high-level features extracted through the lower layers and determine the final recognition result.

### 3.2.2 LSTM (Long Short-Term Memory)

LSTM is a model proposed by Schmidhuber & Hochreiter (1997) and is designed to solve the problem of the long-term dependence of RNN. LSTM has been used in various fields such as machine translation, speech recognition, and text analysis (Shi et al., 2018; Kinghorn et al., 2019; Tiwari et al., 2020). Recently, it has been proven to be effective in the field of stock market prediction by capturing the time series characteristics of input data better than other models. An LSTM memory cell consists of three parts: a forget gate, an input gate, and an output gate.  $t$  in Fig. 1 represents a specific time step and explains the operating mechanism of LSTM through three gates marked inside the cell.

Forget gate receives the hidden state at the previous time and the current input,  $h_{t-1}$ ,  $x_t$ , and outputs a value between 0 and 1 through the sigmoid activation function. Through this process, it is determined how much the  $h_{t-1}$  information received from the previous cell will be forgotten. The operation process of the forget gate is as follows (3). Here,  $f_t$  is (0,1),  $h_{t-1}$  is the output value of the last moment,  $x_t$  is the input value of the current time,  $w_f$  is the weight of the forget gate, and  $b_f$  is the bias of the forget gate.

$$f_t = \sigma(w_f \times [h_{t-1}, x_t] + b_f) \tag{3}$$

Next, the left part of the input gate receives  $h_{t-1}$  and  $x_t$  as inputs and outputs new input information to be reflected in the cell state through the tanh function. The

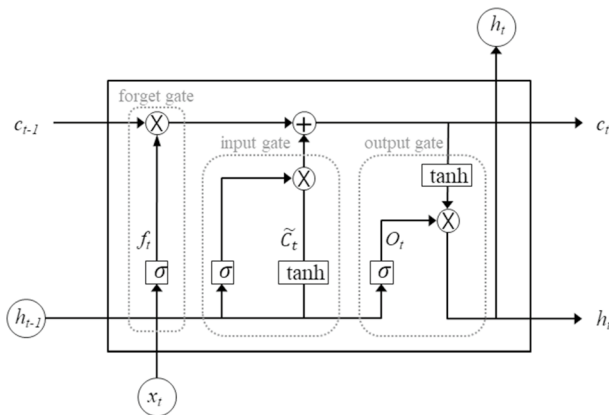


Fig. 1 Structure of LSTM Cell

corresponding process is the same as (4). In the formula,  $w_e$  is the weight of the candidate input gate, and  $b_e$  is the bias of the candidate input gate.

$$\tilde{c}_t = \tanh(w_e \times [h_{t-1}, x_t] + b_e) \quad (4)$$

For the cell state, the value of how much information to forget and remember is multiplied by the cell state at the previous time in the forget gate. Through this, the current state of the cell is created by combining the previous state value reflected with the oblivion information and the information on the current input value. The process is the same as (6). Here,  $c_t$  is the cell state,  $f_t$  is the forget gate,  $c_{t-1}$  is the information of the previous moment, and  $i_t$  is the input gate. The equation of the input gate is as (5). where  $w_i$  is the weight of the input gate, and  $b_i$  is the bias of the input gate.

$$i_t = \sigma(w_i \times [h_{t-1}, x_t] + b_i) \quad (5)$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (6)$$

Finally, the output gate goes through two processes to output the final value. In the first step, the strength of the cell state to be propagated at the next point is determined through the left sigmoid function of the output gate. Here,  $w_o$  is the weight of the output gate, and  $b_o$  is the bias of the output gate.

$$o_t = \sigma(w_o \times [h_{t-1}, x_t] + b_o) \quad (7)$$

In the second process of the output gate, the tanh function is applied to the cell state derived in (6), and then the cell radio wave intensity determined in (7) is multiplied to generate the  $h_t$  value to be propagated at the next time point  $t+1$ . The process of generating the  $h_t$  value is the same as (8).

$$h_t = o_t \times \tanh(c_t) \quad (8)$$

### 3.2.3 Genetic Algorithm (GA)

The Genetic Algorithm proposed by Holland in 1975 is a theory that imitates the principles of survival of the fittest and natural selection, which are the laws of natural evolution. It provides an optimal solution to the optimization problem of large search spaces. It is an algorithm to find through parallel and global search (Jh 1975). The GA process consists of four steps: initialization, fitness calculation, selection, crossover, and mutation. In the initialization stage, N random solutions are set as the population targeting the entire space where the optimal solution to be found exists, and in the next stage, the fitness of the chromosome selected according to the pre-defined fitness function Evaluate After going through a selection process in which a large number of individuals with high fitness values are selected, the crossover in which genetic information is exchanged between the selected individuals and the mutation operation in which random genetic changes are performed are repeated to



evolve the solution group toward the optimal solution. Individuals with good performance are more selected and replicated, while bad individuals are eliminated from the population. The newly formed group by the three operators is evaluated again and the computational process is repeated until an optimal solution is found (Lin et al., 2012). Since GA starts from a random solution group and evolves, it is very unlikely to stochastically fall into a local minima, and in existing studies, it is an experimental study that shows better performance than greedy methodologies in a complex solution space. It was demonstrated in (Oh et al., 2004). In addition, since it is an optimization technique through stochastic iteration, it can be applied to complex optimization problems that are difficult to mathematically model.

### 3.3 CNN-LSTM Hybrid Model Based on GA

The flow chart of the overall GA-based CNN-LSTM integration model proposed in this study is as follows. After performing the preprocessing, the window size was set to 20 through the sliding window method. After constructing the data set, the model is trained by applying GA to optimize the analysis of the model using the training data set. The kernel size and the number of layer filters suitable for CNN networks, the number of layers in LSTM networks, and epochs are searched using GA. It is then applied to the CNN-LSTM model for GA fitness evaluation. Based on the population, the selection and recombination operator starts looking for a good solution, the solution is evaluated with a predefined fitness function and string, and selects the one with good performance for reproduction. In this study, the fitness of chromosomes is calculated and the smallest MSE is selected as the optimal solution for conversion. If the output of the reproduction process satisfies the termination criterion, it is applied to the prediction model, otherwise the entire process of selection/crossover/mutation is repeated again (Fig. 2).

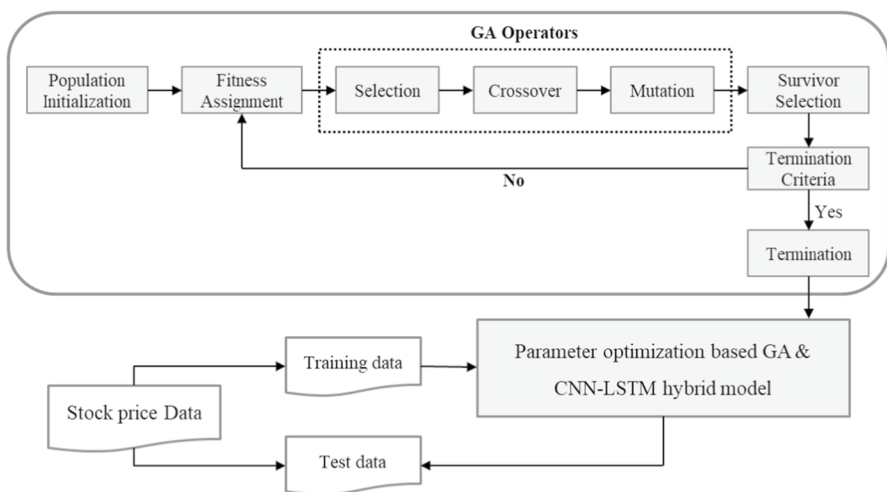


Fig. 2 Framework of GA-based CNN-LSTM hybrid model

### 3.4 Model Performance Metrics

To measure the performance of stock price prediction, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Squared Log Error (MSLE), Mean Absolute Percentage Error (MAPE), Mean Percentage Error (MPE) is used. After reviewing the performance indicators presented in various studies, this study uses RMSE, MAE, and MAPE as evaluation indicators. MSE is used to measure the deviation between predicted and actual values that are sensitive to outliers. MAE is more effective when the error increase is actually proportional to the overall effect. MAPE is also a very useful measure for understanding the performance of predicted values. The formula for calculating each indicator is as follows:

MSE is defined as

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (9)$$

Where  $y_i$  is the desired value,  $\hat{y}_i$  is the predicted output value of the model's  $i$ th observation in the model, and  $n$  is the number of samples.

MAE is defined as follows:

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (10)$$

MAPE is given as

$$\text{MAPE} = \frac{\sum_{i=1}^n |(y_i - \hat{y}_i) / y_i|}{n} \times 100 \quad (11)$$

## 4 Experimental Results and Discussion

The data used in this study were classified as 8:2 between Train and Validation data sets from 2014/9/14 to 2022/10/31. The sliding window method, which is widely used in time series data, was applied, and the data set was reorganized by specifying a window size of 20. The optimization of the CNN-LSTM hybrid model using the GA technique in this study is as follows. First, search parameters were composed of genes to search for genes with optimized values among the population. The search scope used in this paper is as follows. The layer filters of the CNN network were in the range of 4, 64, 128, 256, and 512, respectively, and the kernel size was searched in the range of 3, 5, 7, and 9, respectively. It was fixed using 'relu' as the activation function reflected in the convolution layer. Convolutional layer padding is specified as 'causal'. In this study, the LSTM network was searched in the range of 4 layers of 64, 128, 256 and 512, respectively. And the batch size is set to 32, the optimizer is set to 'adam', and the learning rate is set to 0.0005. To reduce overfitting in the feature extraction process, this study includes dropout, which stochastically excludes

specific nodes from the neural network layer. After initializing the population of the genetic algorithm, a CNN-LSTM model is constructed with the parameters of each gene to proceed with learning. After learning the genes of each of the 10 populations, one of the best genes is maintained and passed on to the next generation, and excellent genes with high suitability are selected for the remaining genes by the roulette wheel selection method. Then, based on the selected gene, single-point crossing is performed excluding the elite gene, and then a mutation is generated for the gene of the offspring generation to form the next generation population. Then, as a termination condition, if the fitness evaluation value is 0 or if the search is up to 3 generations, it ends, and if it is not reached, it repeats. The model with the highest accuracy per generation is selected as the final result of the experiment, and the deep learning model selected in the corresponding generation becomes the fine-tuning result of the experiment.

As mentioned above, in this study, GA was applied for parameter optimization, and the optimized parameters are shown in Table 2. The appropriate kernel size and number of layer filters for the CNN network were found to be 5 and 128, respectively, and the number of hidden layers and epochs of the LSTM network were found to be 512 and 72, respectively, and applied to the model.

The derived results for performance evaluation of the CNN-LSTM hybrid model optimized for GA were measured by calculating MSE, MAE, and MAPE. These performance measures have been widely used and provide a means of determining model efficiency. To test the efficacy of the proposed model, it was compared with CNN, LSTM, and CNN-LSTM models. Table 3 shows the experimental results of the approach proposed in this study.

As shown in Table 3, the CNN-LSTM hybrid model optimized for GA shows superior performance compared to the comparative model in all error measurements. The MSE of CNN, LSTM, and CNN-LSTM were 2324.548, 1735.451, and 1683.706, respectively, while the CNN-LSTM hybrid model was 1604.517, which

**Table 2** Parameter CNN-LSTM based on GA

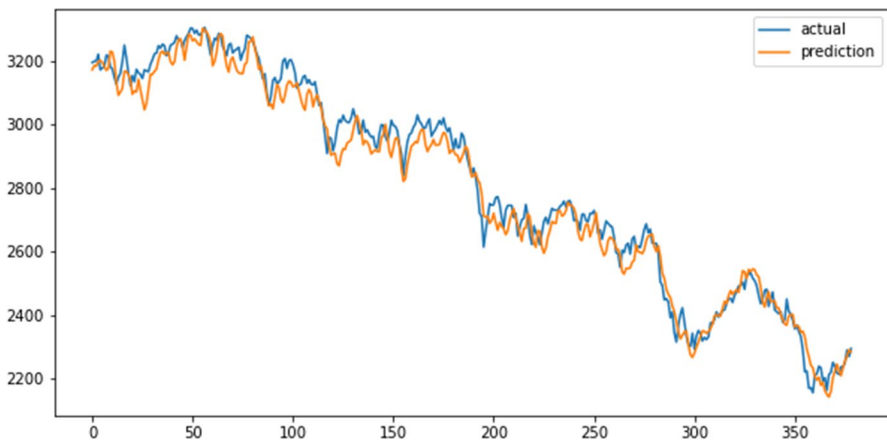
| Parameters                            | Value   |
|---------------------------------------|---------|
| Convolution layer filters             | 128     |
| Convolution layer kernel size         | 5       |
| Convolution layer activation function | Relu    |
| Convolution layer padding             | causal  |
| Number of hidden units in LSTM layer  | 512     |
| LSTM layer activation function        | tanh    |
| Time step                             | 20      |
| Batch size                            | 32      |
| Dropout                               | 0.1     |
| Learning rate                         | 0.00005 |
| Optimizer                             | Adam    |
| Loss function                         | Huber   |
| Epochs                                | 72      |

**Table 3** Prediction performance for KOSPI Index

| Method      | MSE      | MAE    | MAPE  |
|-------------|----------|--------|-------|
| CNN         | 2324.548 | 38.932 | 0.014 |
| LSTM        | 1735.451 | 33.702 | 0.012 |
| CNN-LSTM    | 1683.706 | 32.309 | 0.012 |
| GA-CNN-LSTM | 1604.517 | 31.475 | 0.011 |

was 30.97%, 7.54%, and 4.7% improved compared to the comparative model. The MAE of CNN, LSTM, and CNN-LSTM were 38.932, 33.702, and 32.309, respectively, while the CNN-LSTM hybrid model was 31.475, which was 19.15%, 6.61%, and 2.58% better than the comparative model. The MAPEs for CNN, LSTM, and CNN-LSTM are 0.014, 0.012, and 0.012, respectively, while the CNN-LSTM hybrid model is 0.011. The graphs of the predicted results of the model are presented in Figs. 3, 4, 5 and 6. The blue line represents the actual closing price, and the orange line is the predicted output of the model presented in this study.

In this study, values close to the optimal parameters of CNN-LSTM were searched for using GA, and as a result of comparison with other models to verify the validity of the proposed model approach, MSE, MAE, and MAPE were lower than other models. Which proves that the proposed model is a valid approach with the highest prediction accuracy and the best performance. Experimental results showed that the CNN-LSTM and GA-CNN-LSTM combined models performed better than the single models of CNN and LSTM. In addition, GA-CNN-LSTM to which the parameter optimization technique was applied showed the highest performance compared to other comparison models, proving the superiority of the parameter optimization approach using the GA technique. Parameter tuning is an important factor in achieving satisfactory performance of the model, and finding it is a very difficult task. However, according to the results of this experiment, GA showed the possibility of being an effective tool for determining the optimal model. Through

**Fig. 3** Prediction outputs of CNN model

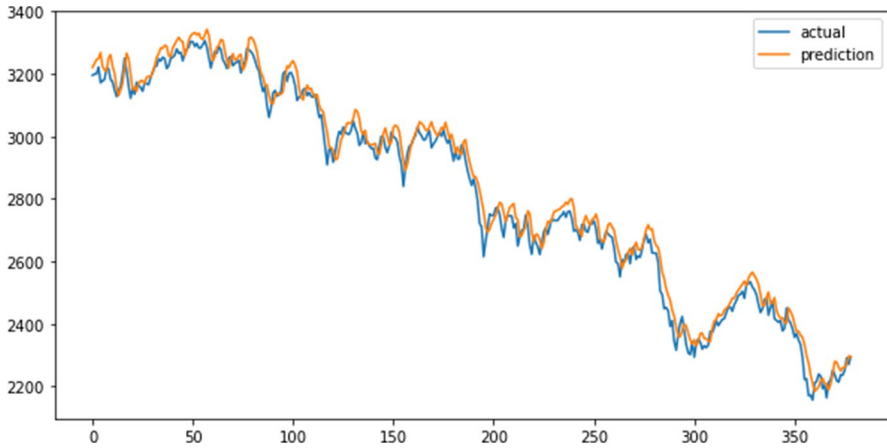


Fig. 4 Prediction outputs of LSTM model

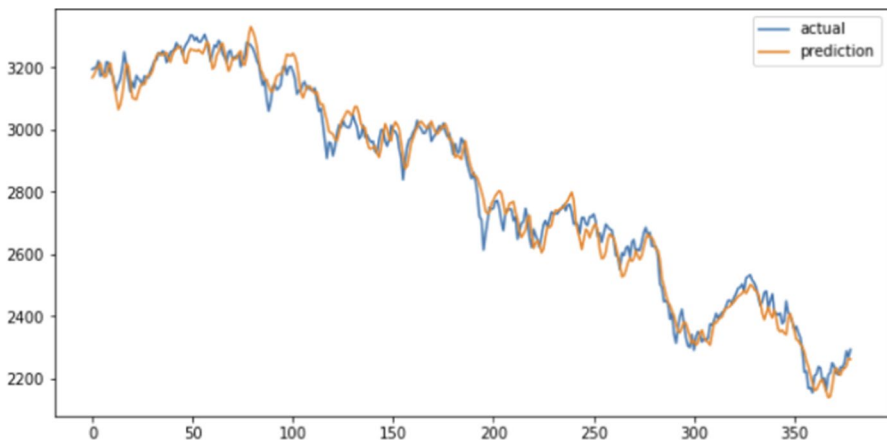
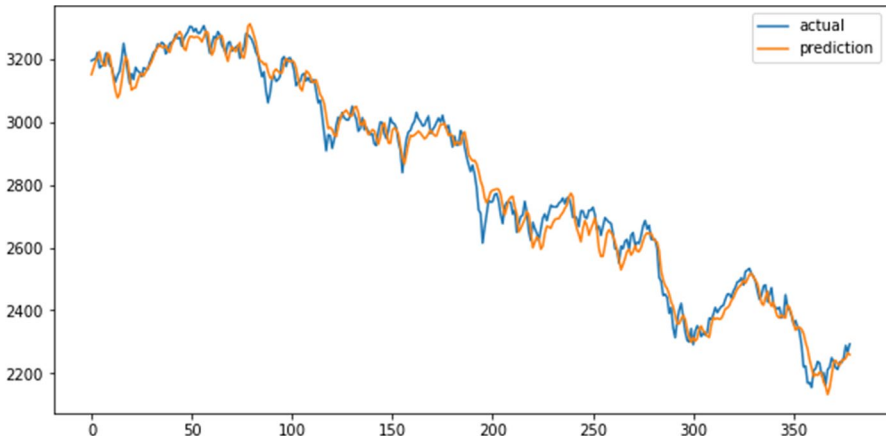


Fig. 5 Prediction outputs of CNN-LSTM model

this, when the number of parameters to be considered is large, the performance may vary depending on whether optimization is applied, which shows the need to apply the optimization process for models with high complexity.

## 5 Conclusion

Stock price prediction is a major field of interest for stock traders and individual investors, and it is important to increase the predictability of the model because it is actually linked to financial loss or gain. In order to predict the complex and highly volatile stock market, there have been many studies to model and predict financial time series using deep learning algorithms, a type of machine learning, from



**Fig. 6** Prediction outputs of GA-CNN-LSTM model

traditional statistical analysis methods. In particular, deep learning techniques are being actively applied with excellent results.

In this study, a stock price prediction model was built by adding parameter optimization using GA to a hybrid model combining CNN and LSTM, one of the representative algorithms of deep learning. To prove the superiority of the proposed model, the performance of the single model and the model without optimization was compared.

Through this study, the following implications were derived. First, according to the results of this experiment, the GA-based CNN-LSTM proposed in this paper was superior to the comparative model and predicted the closing price of the next day well. The proposed model is shown to achieve better performance with higher efficiency, which is useful because it provides a GA-based solution rather than a simple rule of thumb. In other words, in the case of a model built with more than two algorithms, if the parameters are not optimized, the performance of the model may deteriorate. As the efficiency of feature extraction of the CNN algorithm and the prediction accuracy of the time series data of the LSTM algorithm are highlighted, an approach to build a model with optimal parameters is needed. This hybrid deep learning method can make better predictions and successfully handles complex high-dimensional data by providing optimal models for financial forecasting. The proposed model will help explain theoretical and practical tools to investors and policy makers who want to utilize stock price fluctuations as more accurate predictive data using deep learning models.

Although the proposed model has good predictive performance, there are still limitations. First, in terms of data, it is necessary to develop a model suitable for predicting volatility by additionally discovering various macroeconomic factors and applying economic indicators to the model. In the stock market, macroeconomic indicators such as interest rates and price index are also presented as important factors, but this study did not consider them. Second, this study created a short-term forecasting model focusing on the prediction of the next day's closing price, but it is

judged that future research will need to extend the period to predict the mid- to long-term stock price.

## Reference

- Abu-Mostafa, Y. S., & Atiya, A. F. (1996). Introduction to financial forecasting. *Applied intelligence*, 6, 205–213.
- Baresa, S., Bogdan, S., et al. (2013). Strategy of stock valuation by fundamental analysis. *UTMS Journal of Economics*, 4(1), 45–51.
- Bhambu, A. (2023). Stock Market Prediction Using Deep Learning Techniques for Short and Long Horizon. In *Soft Computing for Problem Solving: Proceedings of the SocProS 2022*, Springer, 121–135.
- Bhandari, H. N., Rimal, B., et al. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, 9, 100320.
- Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, 80, 340–355.
- Chung, H., & Shin, K.-S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, 10(10), 3765.
- de Souza, M. J. S., Ramos, D. G. F., et al. (2018). Examination of the profitability of technical analysis based on moving average strategies in BRICS. *Financial Innovation*, 4(1), 1–18.
- Detry, P. and P. Gregoire (2001). Other evidences of the predictive power of technical analysis: the moving averages rules on European indexes. Available at SSRN 269802.
- Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2), 119–130.
- Gu, S.-H., & Jang, S.-Y. (2012). A simulation study of the investment strategy in stocks on fundamental analysis. *Korean Management Science Review*, 29(2), 53–64.
- Jh, H. (1975). *Adaptation in natural and artificial systems*.
- Kai, F. & Wenhua, X. (1997). Training neural network with genetic algorithms for forecasting the stock price index. In 1997 *IEEE International Conference on Intelligent Processing Systems (Cat. No. 97TH8335)*, IEEE.
- Kim, K.-J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, 19(2), 125–132.
- Kinghorn, P., Zhang, L., et al. (2019). A hierarchical and regional deep learning architecture for image description generation. *Pattern Recognition Letters*, 119, 77–85.
- Krizhevsky, A., Sutskever, I., et al. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
- LeCun, Y., Bottou, L., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- Lee, M.-C. (2022). Research on the feasibility of applying GRU and attention mechanism combined with technical indicators in stock trading strategies. *Applied Sciences*, 12(3), 1007.
- Lin, T.-Y., Hsieh, K.-C., et al. (2012). Applying genetic algorithms for multiradio wireless mesh network planning. *IEEE Transactions on Vehicular Technology*, 61(5), 2256–2270.
- Liu, S., Ji, H., et al. (2019). Nonpooling convolutional neural network forecasting for seasonal time series with trends. *IEEE transactions on neural networks and learning systems*, 31(8), 2879–2888.
- Livieris, I. E., Pintelas, E., et al. (2020). A CNN-LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32, 17351–17360.
- Long, W., Lu, Z., et al. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, 164, 163–173.
- Lv, J., Wang, C., et al. (2021). An economic forecasting method based on the LightGBM-optimized LSTM and time-series model. *Computational Intelligence and Neuroscience*, 2021, 1–10.
- Mezghani, T., & Abbes, M. B. (2022). Forecast the Role of GCC Financial Stress on Oil Market and GCC Financial Markets Using Convolutional Neural Networks. In *Asia-Pacific Financial Markets*, pp.1–26.
- Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin.

- Oh, I.-S., Lee, J.-S., et al. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11), 1424–1437.
- Rezaei, H., Faaljou, H., et al. (2021). Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications*, 169, 114332.
- Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. *Neural Computation* 9(8), 1735–1780.
- Sercu, T., & Goel, V. (2016). Dense prediction on sequences with time-dilated convolutions for speech recognition. arXiv preprint [arXiv:1611.09288](https://arxiv.org/abs/1611.09288).
- Shah, D., Campbell, W., et al. (2018). A comparative study of LSTM and DNN for stock market forecasting. In *2018 IEEE international conference on big data (big data)*, IEEE.
- Sharma, N., & Juneja, A. (2017). Combining of random forest estimates using LSboost for stock market index prediction. In *2017 2nd International conference for convergence in technology (I2CT)*, IEEE.
- Shi, L., Teng, Z., et al. (2018). DeepClue: visual interpretation of text-based deep stock prediction. *IEEE Transactions on Knowledge and Data Engineering*, 31(6), 1094–1108.
- Szegedy, C., Liu, W. et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Tiwari, G., Sharma, A., et al. (2020). English-Hindi neural machine translation-LSTM seq2seq and Conv2S. In *2020 International Conference on Communication and Signal Processing (ICCSPP)*, IEEE.
- Wu, J. M. T., Li, Z., et al. (2020). A CNN-based stock price trend prediction with futures and historical price. In *2020 International Conference on Pervasive Artificial Intelligence (ICPAI)*, IEEE.
- Yan, X., Weihan, W., et al. (2021). Research on financial assets transaction prediction model based on LSTM neural network. *Neural Computing and Applications*, 33, 257–270.
- Yang, H., Zhu, Y., et al. (2018). A multi-indicator feature selection for CNN-driven stock index prediction. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part V 25*, Springer.
- Yang, Y., Wu, Y., et al. (2021). Stock price prediction based on xgboost and lightgbm. In *E3S Web of Conferences*, EDP Sciences.
- Yang, Q., & Wang, C. (2019). A study on forecast of global stock indices based on deep LSTM neural network. *Statistical Research*, 36(6), 65–77.
- Ye, F., Wang, J., et al. (2021). Jane Street Stock prediction model based on LightGBM. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, IEEE.
- Yuan, C., Ma, X., et al. (2023). COVID19-MLSF: A multi-task learning-based stock market forecasting framework during the COVID-19 pandemic. *Expert Systems with Applications* 119549.
- Yun, K. K., Yoon, S. W., et al. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Systems with Applications*, 186, 115716.
- Zhang, C., & Bai, Y. (2019). Chinese a share stock ranking with machine learning approach. In *2019 6th International Conference on Information Science and Control Engineering (ICISCE)*, IEEE, 195–199.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.