

Research Article

A Novel Synergetic LSTM-GA Stock Trading Suggestion System in Internet of Things

Jimmy Ming-Tai Wu ¹, Lingyun Sun, ¹ Gautam Srivastava ^{2,3} and Jerry Chun-Wei Lin ⁴

¹College of Computer Science and Engineering, Shandong University of Science and Technology, Qindao, Shandong 266590, China

²Department of Mathematics and Computer Science, Brandon University, Brandon, MB R7A 6A9, Canada

³Research Centre for Interneural Computing, China Medical University, Taichung 406040, Taiwan

⁴Department of Computer Science Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Inndalsveien 28, 5063 Bergen, Norway

Correspondence should be addressed to Jerry Chun-Wei Lin; jerrylin@ieee.org

Received 31 May 2021; Accepted 13 July 2021; Published 26 July 2021

Academic Editor: Fazlullah Khan

Copyright © 2021 Jimmy Ming-Tai Wu 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.

The Internet of Things (IoT) play an important role in the financial sector in recent decades since several stock prediction models can be performed accurately according to IoT-based services. In real-time applications, the accuracy of the stock price fluctuation forecast is very important to investors, and it helps investors better manage their funds when formulating trading strategies. It has always been a goal and difficult problem for financial researchers to use predictive tools to obtain predicted values closer to actual values from a given financial data set. Leading indicators such as futures and options can reflect changes in many markets, such as the industry's prosperity. Adding the data set of leading indicators can predict the trend of stock prices well. In this research, a trading strategy for finding stock trading signals is proposed that combines long short-term memory neural networks with genetic algorithms. This new framework is called long short-term memory neural network with leading index, or LSTM-LI for short. We thus take the stock markets of the United States and Taiwan as the research objects and use historical data, futures, and options as data sets to predict the stock prices of these two markets. After that, we use genetic algorithms to find trading signals for the designed stock trading system. The experimental results show that the stock trading system proposed in this research can help investors obtain certain returns.

1. Introduction

The Internet of Things (IoT) model plays an important role in recent years since many applications and services rely on IoT-based services. In the financial sector, the IoT-based services [1] include banking, insurance, and investments. Forecasting or prediction [2] is the most interesting thing in the financial market, and many studies [3, 4] deploy IoT-based services to achieve higher accuracy. Li et al. [5] studied the development of IoTs and used deep learning to predict stock price trends. Yang et al. [6] proposed a new data mining algorithm to study the multimedia Internet of Things and the stability of stocks. The financial market is a

mechanism for determining the price of financial funds and trading financial assets. It is a market that enables the financing of securities and the trading of securities. The capital market is also called the "long-term financial market," which mainly includes the stock market, fund market, and bond market. Its volatility can reflect the degree of risk of assets. The fluctuation of stock prices plays a considerable role in the appropriate timing of buying and selling stocks [7]. For investors, the true meaning of investing in the stock market is to obtain extraordinary returns by buying low and selling high, so the prediction of stock price fluctuations has become a special focus of private investors and investment companies [8]. Due to a large amount of data in stocks, the

low data correlation, and many factors that affect stock prices, financial markets are full of uncertainty, which also makes predicting stock price fluctuations a significant problem for stock researchers. Initially, people had predicted the fluctuations of stock prices several times [9, 10], but the results were unsatisfactory.

Investors are not completely rational. For example, people may have positive or negative emotions at certain moments [11–15]. Therefore, after this hypothesis was raised, there were both support and opposition, thus becoming one of the most controversial investment theories [16, 17]. Some researchers believe that if the trading signals [18] of the stocks can be found, the stocks can be bought and sold at the appropriate time to obtain relatively high profits. Finding accurate stock trading signals is the key to obtaining huge returns, that is, in stock trading, when a buy signal appears, the stock is bought, and when a sell signal appears, the stock is sold. In other words, the stocks should be purchased at a lower price but sold out at a higher price if you would like to obtain good profit value during the trading. Stock trading signals have always been the main research objects of investors and investment companies. Researchers have also tried many times to predict the best buying and selling signals, but because many factors affect stock prices, the prediction results are not satisfactory.

Support vector machine (SVM) is a binary classification model [19] and has good classification performance in the field of machine learning. Huang et al. [20] found in 2005 that SVM has better prediction results than traditional statistical methods. Nayak et al. [21] proposed a hybrid model for the prediction of Indian stock indexes. This model mainly uses two machine learning algorithms, namely SVM and k-nearest neighbor (KNN). Studies have shown that the hybrid model they proposed has good scalability and predictability for high-dimensional data. Although the prediction results of SVM are significantly improved compared to other models [22, 23], it is still not satisfactory for stock prediction. The neural network model has received particular attention. Because neural network models can identify nonlinear relationships in data sets, many researchers have analyzed, and predicted stock price fluctuations based on various neural networks have also obtained relatively good results. For the stocks of the Tokyo Stock Exchange, Kimoto et al. [24] conducted a systematic study. They proposed a stock prediction system based on neural networks, and the system achieved relatively good profits in training.

With the rise of machine learning, deep learning [25] has attracted the attention of various stock researchers. It stimulates the mechanism of the human brain to process data and can detect nonlinear relationships in the data. In this document [26], the application of machine learning in financial markets is explained in detail. A recurrent neural network [27, 28] is a kind of deep learning based on nonlinear prediction. In 2017, Fischer and Krauss [29] did a large-scale study. They used daily S&P 500 data to measure the prediction performance of LSTM compared to other neural networks. The research results show that the prediction performance of the LSTM model is the best. This paper [30] uses LSTM's RNN to predict stocks and calculate

returns based on closing prices. Experimental results show that the performance of the LSTM model is better than the feedforward artificial neural network.

Before the emergence of recurrent neural networks, both artificial neural networks and convolutional neural networks (CNNs) achieved good prediction results [31, 32]. For example, by entering a picture, the convolutional neural network can accurately predict whether the image is a cat or a dog. However, since the inputs of the convolutional neural network are independent of each other and cannot be predicted based on the data of the previous moment, the prediction effect of the CNN is not very good on the problem of time series. Therefore, people have developed a recurrent neural network with a memory function. The input of the RNN consists of the input of the current moment and the output of the previous moment, so it has a simple memory function. Through multiple iterations, the input layer and hidden layer, the hidden layer at the previous moment and the hidden layer at the current moment, and the weight between the hidden layer and the output layer are repeatedly modified to obtain the optimal weight parameters and achieve better prediction results. In 2018, Chen et al. [33] used the RNN-boost model to study the Shanghai-Shenzhen 300 Stock Index (HS300). They select official accounts from Sina Weibo and analyze the content grabbed from these accounts by extracting emotional features. Next, they input the extracted features and technical indicators into the RNN-boost model to predict the stock price fluctuations.

At present, there have been many applications implemented with RNNs, such as speech recognition [34], text generation [35], machine translation [36], and so on. But at the same time, RNN has two significant shortcomings: one is that the learning ability of RNN is limited; somehow the performance for prediction will have a negative effect if the size of the sequence is relevantly long and the other is that the gradient of RNN will disappear, resulting in long-term memory failure. Later, people improved the RNN, mainly adding a forget gate, input gate, and output gate to retain and control information. The improved RNN has a better prediction effect on both short and long sequences, so researchers define it as long short-term memory, or LSTM for short.

This research [37] combines the technical indicator-moving average with the globe best-guide quantum-inspired tabu search algorithm to propose a new dynamic trading system. Among them, the moving average is used to find stock trading signals. In this paper, the proposed method is compared with the buy-and-hold strategy, and the experimental results show that this method significantly improves investment returns. This article mainly studies how to use deep learning algorithms and intelligent optimization algorithms to find the best trading signals. The framework extracts useful features from different data and then trains a predictive model based on these features. In this study, the output obtained by the predictive model is combined with the intelligent optimization algorithm to obtain a buying and selling signal. In stock trading, finding the best buying and selling signals is the key to gaining income. In this work, a new long short-term memory neural network framework is

used. It integrates the collected data into the form of images, inputs them in the form of images, and uses these images to train neural networks. This research combines the LSTM framework and GA to form a stock trading system. From the extensive experiments, it can be observed that the proposed stock trading system is superior to other trading strategies. The key significances of this article are described as follows:

- (1) This research combines the LSTM framework with genetic algorithms and proposes a stock trading system. This system can find a stock trading signal relatively well so that investors can get a certain amount of income.
- (2) In this study, we have added two data sets: futures and options. Leading indicators are indicators that change first before economic growth or recession. This article only uses two leading indicators, namely, futures and options. Extensive experiments indicated that after adding futures and options data sets, the prediction accuracy is higher.
- (3) An LSTM framework with a two-dimensional tensor as input is proposed. This research is improved based on LSTM, and the output is divided into three categories: rising, falling, and unchanged. Experimental results show that the improved framework has better performance.

This article mainly includes the following parts. Section 2 introduces related works. Section 3 introduces the methods used in this article. Section 4 mainly describes the framework proposed in this research. Section 5 is the experimental results of this research. Finally, a detailed summary of this research is made in the conclusion section.

2. Related Works

The prediction of stock prices is mainly the analysis of historical behaviors, such as people's historical emotions, historical market information, etc., and then useful features are extracted from them to train better predictive models. The value of stock prices is a time series. Revealing the development and changes in stock prices is an objective record of stock historical behavior. In the early days of the stock market, a host of investors relied on their own experience to judge stock price movements, which seemed very subjective and lacked a scientific basis. Besides, stock prices are also affected by many other factors. For instance, Zheng et al. [38] studied the relationship between exchange rates and stock prices on the Hong Kong stock market. They built a model to analyze the correlation between Hong Kong company stocks, mainland company stocks, and exchange rates. The experimental results show that the exchange rate is negatively correlated with the stocks of local companies in Hong Kong and positively correlated with the stocks of mainland companies. The price of stocks is also affected by external information and human emotions [39]. Many researchers have studied human emotions by analyzing the content on the forums or Weibo to more accurately predict stock price volatility. We can also find that many factors can affect stock price volatility, so we need some tools to analyze

and predict stock. Several studies [40, 41] were then developed to investigate the deep learning models used in stock trading.

Intelligent optimization algorithms are used to find the optimal solution. In recent years, researchers have been more enthusiastic about using genetic algorithms to solve problems. In the financial market, many researchers also use genetic algorithms (GAs) to find stock trading signals. Allen et al. [42] utilized GAs for discovering the technical trading rule according to the S&P 500 index of the collected daily records (prices) in a range of 1928 to 1995. In this research [43], based on a series of technical indicators that generate buying and selling signals, GA is used to propose a trading strategy. Hirabayashi et al. [44] proposed a GA-based system to find suitable trading signals and automatically generate trading rules based on technical indicators. The focus of this system is not to predict the price of the transaction but to find the right trading opportunity. Lin et al. [45] used a GA-based model to set optimal values for the parameters of the problem and bought or sold stocks at the appropriate trading time.

The price of stocks changes with time, and the phenomenon of rising or falling occurs, so the data of stock prices is a typical time series. When we use the model to predict the price of stocks, the input of the model can be not only digital data but also image data. At this time, we thought that the CNN model could be used to predict the price of the stock. Research has proved that CNN is outstanding in image recognition, and its accuracy is higher than other models. Di Persio et al. [46] proposed a model that uses wavelet and CNN methods to predict stock trends based on data from the past few days. Experimental results show that the proposed new method is superior to other models and basic neural networks.

The stock price is usually a time series data. For this kind of data, the LSTM neural network model has a better effect than other models. The LSTM model is an improvement of RNN. LSTM adds a forget gate, input gate, output gate, and memory unit to solve the problem of disappearing the RNN gradient. Because the LSTM model has better performance in terms of time series, many researchers use LSTM to predict stock price fluctuations. This document [17] proposes an integrated LSTM for detecting stock prices within a day, using technical indicators as input to the network. They evaluated the predictive power of the proposed model on several US stocks, and the experimental results showed that the proposed integrated LSTM model performed better than the benchmark model. Nelson et al. [47] established a model that uses the LSTM neural network to predict stock price trends based on technical indicators and historical price data. Through a series of experiments to evaluate the performance of the proposed model and other machine learning algorithms, experimental results show that the prediction accuracy of the proposed model is relatively high, and the average accuracy is 55.9%.

3. Methodology

In the first part of this section, the SSACNN framework [48], LSTM, and the genetic algorithm will be introduced and integrated as a newly designed framework. This SSACNN

framework was proposed in our previous research. The second part will introduce the basic LSTM neural network. In the third part of this section, we will introduce the genetic algorithm used in this research. This algorithm is used to find the optimal solution of stock trading signals to maximize returns.

3.1. Stock Sequence Array Convolutional Neural Network. Convolutional neural networks have good results in image recognition, such as handwritten digit recognition, input several pictures of handwritten digits, and repeatedly modify the weight parameters through multiple training of CNN. The resulting neural network model can accurately recognize handwritten digits. CNN includes a convolutional layer, pooling layer, and fully connected layer. The convolution layer mainly extracts local features of the input data. The researcher defines a convolution kernel inside the convolution layer. Its shape is a square matrix that is used to extract a certain feature. The convolution kernel is multiplied by the corresponding bits of the digital input matrix and then added to obtain the output value of a convolution layer. The higher the output value, the higher the degree of matching between the two. Because one convolution kernel recognizes one feature and the input data may have multiple features, there may be numerous convolution kernels in one convolution layer to extract multiple features. The output of the obtained convolution layer should be used as the input of the pooling layer.

However, before entering the value into the pooling layer, an activation function is usually added to solve the nonlinear problem. At present, the activation function ReLU (Rectified Linear Unit) is commonly used. The pooling layer is mainly used to reduce the number of training parameters and reduce the dimension of the feature vector output by the convolution layer. The most common pooling layers are maximum pooling and mean pooling. In this article, we choose maximum pooling, that is, the maximum value in a specified area is selected to represent the entire area. The output value of the pooling layer is then expanded as the input of the fully connected layer to generate the final output. After several times of convolution, excitation, and pooling, the model will learn a high-quality feature map, and then the feature map is used as the input to the fully connected layer to get the final output.

In this study, the stock data for some time is converted into an image, and this image is used as the input of the CNN framework. The input here is 30-day stock data, and the generated “input image” is input to the convolutional layer, pooling layer, dropout layer, and norm layer. Then, this process is looped three times. After a series of experiments, it is concluded that when the convolutional neural network is used for image recognition, the size of the convolution kernel is 3×3 , and the size of the pooling layer is 2×2 , the experimental effect obtained is the best. Therefore, to achieve better results for this research, the size of the convolutional layer and pooling layer is set to 3×3 and 2×2 , respectively.

3.2. Long Short-Term Memory Neural Network. If we want to predict the events that will happen in the future, we need to call our memory to review it, select useful information from it, and predict the development of events based on this useful information. RNN is a neural network with a memory function. Unlike the traditional neural network, there is a loop inside the RNN. It can pass the information retained at the current moment to the next moment and use it to maintain the continuity of the information. LSTM is an improved neural network based on RNN, which solves the problem that the RNN gradient disappears [27] and cannot be long-term dependent. LSTM adds three gates (forget gate, input gate, and output gate) and a cell state based on RNN to forget and retain information. The internal structure of LSTM is shown in Figure 1. Several status information of LSTM is shown as follows:

$$\begin{aligned}
 f_t &= g(x_t \cdot w + h_{t-1} \cdot v + b_0), \\
 i_t &= g(x_t \cdot w + h_{t-1} \cdot v + b_1), \\
 o_t &= g(x_t \cdot w + h_{t-1} \cdot v + b_2), \\
 \tilde{c}_t &= \tanh(x_t \cdot w + h_{t-1} \cdot v + b), \\
 c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t, \\
 h_t &= o_t \cdot \tanh(c_t).
 \end{aligned} \tag{1}$$

where g represents the activation function, which is used to decide what information is allowed to pass. The range of g is from 0 to 1. When g is set to 1, it means that all information is allowed to pass. When g is set to 0, it means that no information is allowed to pass. \tanh stands for hyperbolic tangent function, which is used to control new information, that is, how much new information is reflected, and the range is set as -1 to $+1$. w and v are the weight matrix, and b stands for bias. h_{t-1} represents the output at the previous moment, and x_t represents the input at the current moment. Among them, the forget gate is marked as f_t ; the input gate is marked as i_t ; and the output gate is marked as o_t . x_t and h_{t-1} are input to the three gates labeled f_t , i_t , and o_t . One of the differences between RNN and LSTM is that there is an extra conveyor belt named “cell state” at the top of LSTM, which is marked as c in Figure 1 and used to memorize information.

The long short-term memory neural network is a special recurrent neural network. As a network with memory, LSTM’s memory function is in a sequence, not between sequences. For example, we decompose an article, and each sentence is a training object. $X[i]$ represents the i -th sentence, and each word in each sentence represents a time step. $X[i][0]$ means the first word in the i -th sentence. The information of this word can be memorized and passed to the fifth word $X[i][4]$. It can be seen that the memory function of LSTM works in one sentence.

3.3. A Proposed LSTM Framework with Leading Indicator, LSTMLI. In this section, the process proposed by the LSTMLI framework is specifically introduced.

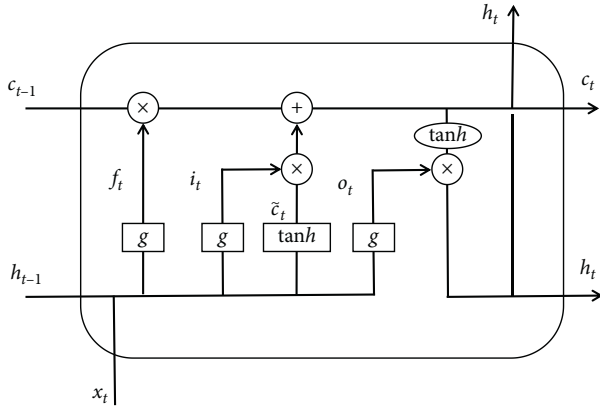


FIGURE 1: Schematic diagram of the internal structure of LSTM.

3.3.1. LSTMMLI Framework. This paper divides the data into two parts according to the ratio, which is used for training and testing. This experiment integrates 30-day stock data into a matrix, inputs it into the LSTMMLI framework, and outputs 3 nodes, as shown in Figure 2. An example of an input matrix is shown in Table 1. The study defines the stock's ups and downs as a label for comparison with the predicted value. The model is then trained by the developed LSTMMLI to predict the stock price fluctuations for the next day, and improve the prediction accuracy by repeatedly modifying the weights. Besides, this study defines labels as three types: +1, 0, and -1. The actual meaning is that the price of the stock has risen, fallen, and has not changed significantly, as shown in the following equation:

$$\text{label} = \begin{cases} +1, & l > 1 \\ 0, & \text{others,} \\ -1, & l < -1. \end{cases} \quad (2)$$

This experiment operates on the data of the column where the historical data is rising and falling. When l is greater than or equal to 1, the label is defined as 1. When l is less than -1, the label is defined as -1. In other cases, the label is defined as 0. Thus, the LSTMMLI model will give a qualitative output based on the input data. This experiment sets the learning rate to 0.001; the weight is to take the specified number of values randomly from the sequence that obeys the specified normal distribution and offset to a two-dimensional tensor. All elements of the tensor are filled with 0.1. The pseudocode of the specific process is shown in Algorithm 1.

3.3.2. Detailed Architectures of Input Images. In the proposed LSTMMLI architecture, its input image is very complicated. Therefore, this section describes the detailed architecture of the input image used in the experiments. First, the futures, options, and historical prices generated images are explained below. It is to be noted that this section focuses on the data structure of the architecture; all values will be standardized using the methods presented above.

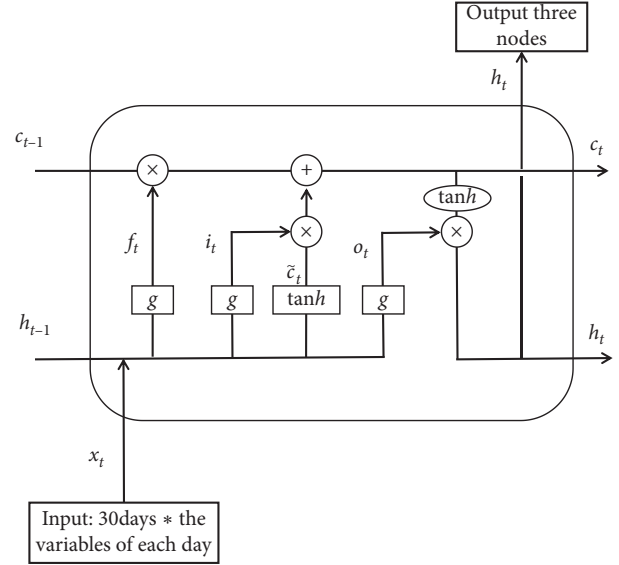


FIGURE 2: LSTMMLI algorithm experiment flowchart.

TABLE 1: Example of the input matrix.

	1	2	3	...	30
Open	266	262	262	...	230
High	266	264	262.5	...	234.5
Low	260	261	257	...	230
Close	262.5	263	257.5	...	231.5
Label	-2.5	0.5	-0.5	...	0.5

(1) *Historical Price Image.* From the above definition, the image includes *opening price*, *lowest price*, *highest price*, *closing price*, and *volume* of stock within 30 days. We only need to put these values in a column with a certain order and expand them for 30 days as 30 columns to build an image. An example is shown in Figure 3 in which O_n is the *opening price*, H_n is the *highest price*, L_n is the *lowest price*, C_n is the *closing price*, and V_n is the *volume* at n -th day.

(2) *Futures Image.* It shows *opening price*, *highest value*, *lowest value*, *closing price*, and *volume*, which are similar to historical prices. In addition, the futures products of a particular underlying asset (stock) can have several different expiry dates. In the developed model, we choose a future whose three expirations are closest to the current date. The relevant attributes is put within one day of the futures in one column, and 30 days is expanded as a matrix. Figure 4 shows an example, and we can indicate the following parameters such as O_n is the *opening price*, H_n is the *highest price*, L_n is the *lowest price*, C_n is the *closing price*, and V_n is the *volume* at n -th day.

(3) *Options Image.* Compared to futures and historical prices, the situation with options is more complicated. There are two types of options (call options and put options) in the options market. In this study, we only select the near-month option data of stocks. In this method, 10 different options (10

Input: D is the data of training; K is the data of testing; z is the number of the maximum iteration as the terminal condition.
Output: the optimal weights w ; evaluation result accuracy.

- (1) initialize the input data D to D^* . D^* is an input array for the information generated by historical prices, futures, and options and assume it consists of m input images.
- (2) label each individual in D^* and generate the label array D_l .
- (3) generate the initial weights w randomly for the LSTM network.
- (4) **for** each round 1 to z **do**
- (5) **for** each round $t = 1$ to m **do**
- (6) input D_t^* to the proposed LSTM network and generate class label vector l .
- (7) compare D_l^t and l to update w by Adam method [49].
- (8) **end for**
- (9) **end for**
- (10) evaluate accuracy using the optimal weights w for testing data K .
- (11) output w and accuracy.

ALGORITHM 1: The proposed LSTMLI framework and evaluation.

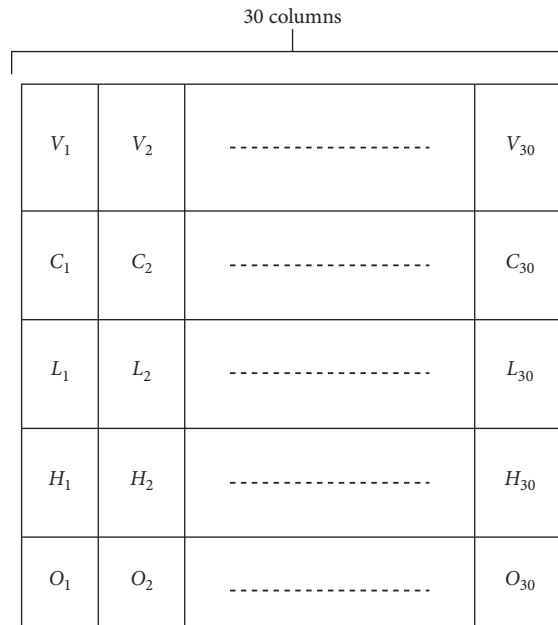


FIGURE 3: This image is built with several options.

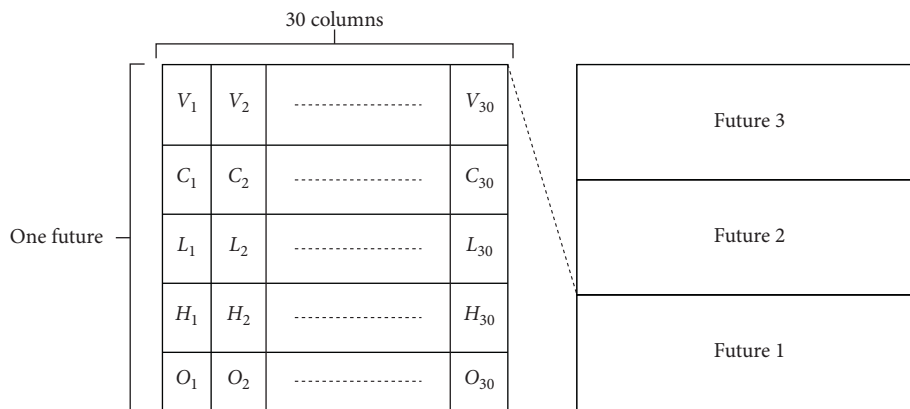


FIGURE 4: This image is built with several futures.

call rights and 10 put rights) are selected, and its settlement price is closest to the current price of the stock, and attributes are extracted from these options to construct an image. Attributes include *volume*, *open interest*, *closing price*, and *settlement price*. Its composition is similar to the previous two images, and the 30-day data is also expanded into a matrix. Figure 5 shows an illustrated example, and we can find that S_n is the *settlement price*, C_n is the *closing price*, V_n is the *volume* at n -th day, and O_n is the *open interest* at n -th day.

(4) *Combination Image*. The final input form for the system is the combined image. In this structure, the first three images are then used to produce an image by combing the information such as prices, futures, and options, and Figure 6 shows an illustrated example for this combination image.

3.4. *Genetic Algorithm*. Darwin's theory of evolution is the root of a genetic algorithm. The genetic algorithm was first proposed by John Holland. This algorithm converts the process of solving actual problems into a process similar to biological evolution and finds the optimal solution by simulating biological evolution. In this section, the components of the method are described. The specific process is as follows.

3.4.1. *Chromosome Representation*. The main goal of this method is to find trading signals for stocks, namely, buy and sell signals. Therefore, in this method, the chromosome is composed of two parts, as shown in Figure 7.

The first part represents the threshold when the stock has a buy signal, and the second part represents the threshold when the stock has a sell signal.

3.4.2. *Initial Population*. It is very important to design a strategy for generating population initialization, which will affect the final profit. In this study, two floating-point numbers were randomly generated within a specific range as the threshold for the initial buying and selling signals. The sum of the values of the nodes obtained by the neural network is approximately equal to 1. So during initialization, two floating-point numbers between 0 and 1 are randomly generated each time. An example is given below to explain in detail. Assume that the thresholds of two randomly generated buying and selling signals are 0.76848 and 0.82843, respectively (because this is just an example, not all decimal points have 5 digits), as shown in Figure 8. SSACNN is used to generate the values of 3 nodes, and an example is given to illustrate the generated values, as shown in Figure 9.

It can be seen from Figure 9 that the value of the position of the first node on the first day is 0.85551; because $0.85551 > 0.76848$, the stock is bought at this time. After that, we then start to observe the third node of each day. It can be from Figure 9 that the value of the third node on the second day is 0.79545, which does not meet the condition of greater than 0.82843. Therefore, the stock will continue to be

held on the second day, and neither buy nor sell will be performed. The value of the third node on the third day is 0.83419; because $0.83419 > 0.82843$, the stock is sold on the third day. After that, the buying and selling operations are performed on the stocks following the above rules. In this research, buying operation of stocks refers to buying with all funds, and the selling operation of stocks refers to selling all the stocks.

3.4.3. *Fitness Evaluation*. The fitness function is used to evaluate the quality of chromosomes, that is, to evaluate whether the threshold value of the generated buying and selling signals is the optimal solution. In this method, the fitness function is defined as the cumulative return over a period of time, which can be seen in the following equations:

$$\text{share}_i = \frac{\text{TotalAmount}_i}{\text{B Close}_i}, \quad (3)$$

$$\text{balance}_i = \text{TotalAmount}_i - \text{share}_i * \text{B Close}_i, \quad (4)$$

$$\text{FinalAmount}_i = \text{balance}_i + \text{share}_i * \text{S Close}_i, \quad (5)$$

where S Close_i represents the closing price of the stock when the i -th sell signal appears, B Close_i represents the closing price of the stock when the i -th buy signal appears, share_i is the share of the stock, TotalAmount_i is the total amount before the i -th transaction, and FinalAmount_i is the total amount after the i -th transaction is completed.

3.4.4. *Population Selection*. In each evolution, to retain good chromosomes, chromosomes need to be selected. The method of selecting chromosomes in the experiment is as follows. We randomly select chromosomes, with every two chromosomes as a group, and compare the size of the result obtained by using each chromosome and then select the chromosome corresponding to the larger value of the result that is, a chromosome will be selected in each group. For example, if there are 8 chromosomes in total, 4 chromosomes will be selected.

3.4.5. *Crossover*. The genetic algorithm performs crossover and mutation operations to generate new solutions. The crossover operation needs to find two chromosomes from the previous generation and set them by the condition as one chromosome is treated a father and another one is considered a mother. Then the two chromosomes are cut and spliced together at a certain position to generate a new chromosome. Part of the new chromosome is the father's genes, and the remaining part is the mother's genes. Not every iteration will perform crossover operations on genes. In this study, the crossover ratio was set to 0.8, and its specific process is shown in Figure 10.

3.4.6. *Mutation*. The crossover operation is only to operate on the original chromosomes and exchange their gene order.

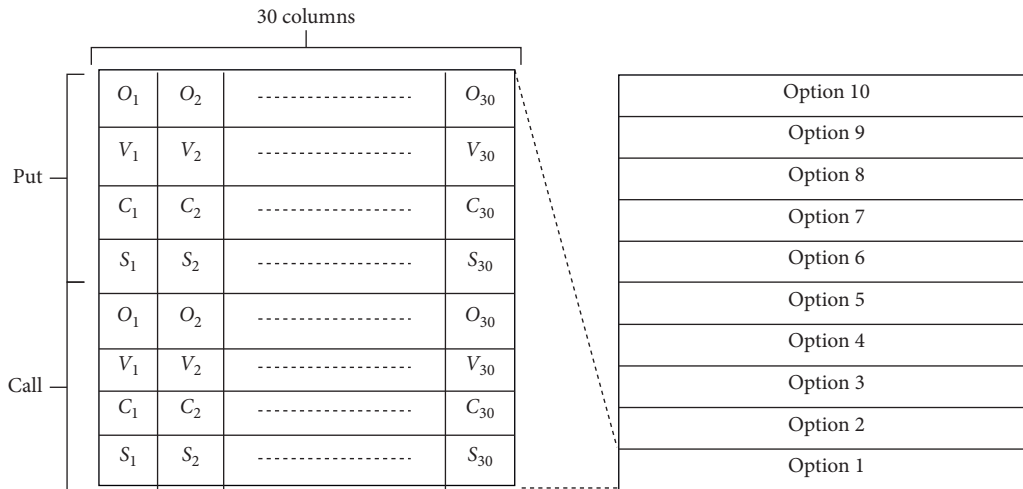


FIGURE 5: This image is built with several options.

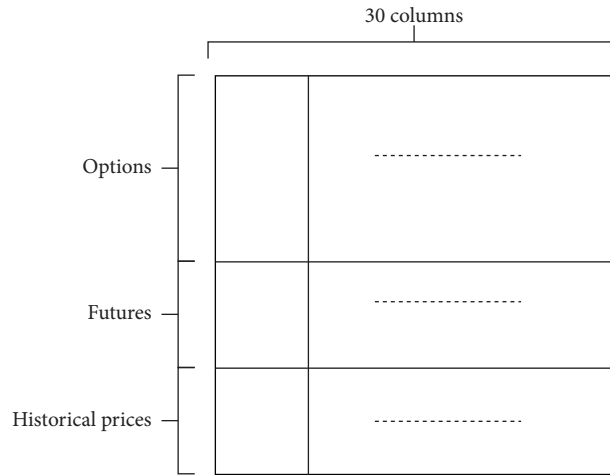


FIGURE 6: An example of the image established by combination data.

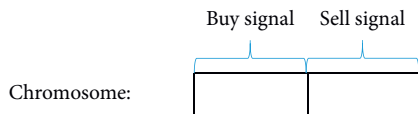


FIGURE 7: Representation of the chromosome.



FIGURE 8: An example of the threshold of randomly generated buying and selling signals.

	Buy signal	Sell signal
1	0.85551	0.02341
2	0.14344	0.79545
3	0.17535	0.83419
.....
.....	0.71714	0.15575

FIGURE 9: An example of the values of three nodes output by SSACNN.

This can only guarantee a local optimal solution after multiple evolutions. To achieve the global optimal solution, a mutation operation is added. After that, we can introduce new genes into existing chromosomes by randomly modifying genes. In this study, a gene was selected for mutation.

We then randomly generate a floating-point number between 0 and 0.1 and add the floating-point number and the value of the selected gene as the mutational gene value. The mutation process is shown in Figure 11. From the observed results, we can see that if the program chooses to mutate the

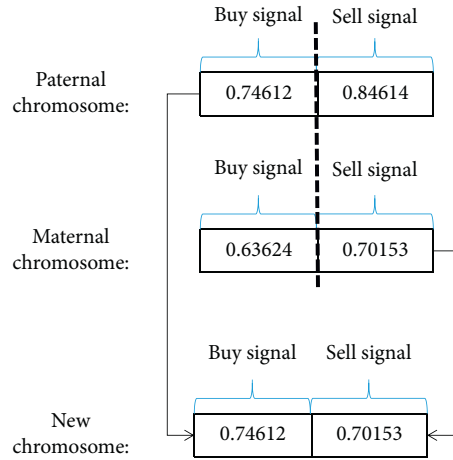


FIGURE 10: An example of chromosome crossover.

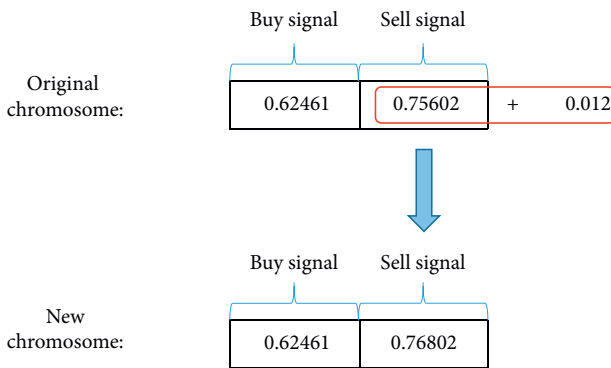


FIGURE 11: An example of chromosome mutation.

second gene, it changes the value of 0.75602. Then a floating-point number 0.012 is randomly generated. The sum of these two values is taken as the value after the mutation, which is 0.76802. Not every iteration will perform mutation operations on genes. In this study, the mutation ratio was set to 0.2.

3.4.7. Iteration. We then set an integer value in the experiments. When the number of iterations reaches this value, the program stops running. The number of iterations set in the experiments is 1,000.

4. The Proposed Stock Trading System Combining LSTMLI Framework and GA

This paper proposes a stock trading system based on LSTMLI and GA. In the first subsection of this chapter, the data set used in the experiment is briefly introduced. The second part is to initialize the data set and use the processed data as input to the framework. The third part introduces the proposed stock trading system.

4.1. Data Sets. The data sets used in the experiments are historical data, futures, and options because, in financial

markets, stock prices are affected not only by historical behaviors but also by leading indicators. Leading indicators can reflect the state of future economic development and are also called predictive indicators, which can provide predictive information for future economic conditions. Using these indicators, researchers can know the turning point of the economy in advance to adopt appropriate trading strategies. For example, when the monetary authorities reduce the money supply, on the one hand, it shows the policy intentions of the authorities, implying the current trend of overheating in the economy. On the other hand, it will bring an increase in interest rates, which will increase the cost of enterprises and reduce profits, thereby reducing the attractiveness of investors. The increase in interest rates increases the opportunity cost of stock investment, which inevitably leads to a reduction in investment and a decline in stock prices. Therefore, this study focuses on the correlation between two leading indicators of futures and options in the stock market and uses leading indicators to predict the trend of stock prices.

Five important indicators of historical price are often used: *opening price*, *lowest price*, *highest price*, *closing price*, and *volume* for stock price analysis. The futures indicators used in our experiments include *opening price*, *highest value*, *lowest value*, *closing price*, and *volume*. The indicators used here include *volume*, *open interest*, *closing price*, and *settlement price*.

First, let's look at a data set of five stocks in the Taiwan market. The five stocks are CDA, CFO, DJO, DVO, and IJO. Tables 2–4 are the historical data, futures data, and option data of the five Taiwan stocks, respectively.

In Tables 2–4, d_i , t_i , and z_i are the various factors that affect the stock price. d_i represents the open, high, low, close, or volatility attributes of the stock. t_i represents the current price, the opening price, the highest price, and the closing price of the futures. z_i represents the attributes of open interest and settlement price of options. Among them, options include buy and sell options.

4.2. Data Initialization. Before the experiment, input data needs to be further processed, that is, to standardize the data

TABLE 2: The historical prices of the five stocks.

	d_{i_1}	d_{i_2}	d_{i_3}	d_{i_4}	d_{i_5}
CDA	266	266	260	262.5	...
CFO	118	118	115	115	...
DJO	260	264	259	264	...
DVO	244.5	246.5	243	246.5	...
IJO	3,825	3,880	3,630	3,635	...

TABLE 3: Futures data of the five stocks.

	t_{i_1}	t_{i_2}	t_{i_3}	t_{i_4}	t_{i_5}
CDA	262.5	265	265.5	262.5	...
CFO	117	117	117.5	117	...
DJO	163.5	262	263.5	258	...
DVO	246	244	246.5	243.5	...
IJO	3,660	3,815	3,865	3,635	...

TABLE 4: Option data of the five stocks.

	z_{i_1}	z_{i_2}	z_{i_3}	z_{i_4}	z_{i_5}
CDA	5.2	70	7.3	11	...
CFO	3.25	20	0.51	15	...
DJO	14.85	1	0.27	1	...
DVO	3.4	2	6.85	2	...
IJO	297	0	30.1	0	...

because the experiments may be affected by the size of the data and the results are not ideal. Thus, the first priority thing to do is to standardize the original data to ensure the consistency in terms of the range of all collected data. The processed data conforms to the standard normal distribution and helps improve the accuracy of the prediction, which can be referred to the following equation:

$$x^* = \frac{x - x_{\text{mean}}}{x_{\text{max}} - x_{\text{min}}}, \quad (6)$$

where x_{mean} is the average value of data samples within 120 days, x_{max} is the maximum value of the data sample within 120 days, and x_{min} is the minimum value of the data sample within 120 days. Because the stock price within 120 days will not fluctuate too much, the data within 120 years is used as a set of values to normalize. Taiwan's stock data is taken as an example to standardize historical prices, and the standardized data is shown in Table 5.

4.3. The Proposed Stock Trading System. In this section, the proposed stock trading system will be introduced in detail. This system consists of the LSTM framework and GA. To make the predicted stock price closer to the actual value, this research uses three data sets, including historical prices, futures, and options. First, input three data sets into the LSTM framework to get the values of three nodes. These three nodes represent three meanings: the first node represents a buy signal, the second node represents a hold signal, and the third node represents a sell signal. Here, only the first and the third nodes are used. The reason is that in

this paper, you only need to know the buy and sell signals of the stock. Next, two nodes are used to find the optimal threshold of stock trading signals. The nodes are considered as the input data in GA, and the thresholds of the two buying and selling signals can be obtained through operations such as crossover, mutation, and iteration. Finally, the value of the node obtained through the test data set is compared with the threshold to obtain a more correct buying and selling point, and this rule is used to calculate the cumulative income over a period of time. The flowchart of this stock trading system is shown in Figure 12.

5. Experimental Results

This experiment mainly uses leading indicators to study stock trading signals. For investors, finding accurate trading signals is crucial, that is, stock buy and sell signals. Because trading signals are not only affected by one factor, trading signals are difficult to be accurately predicted. In this research, a new LSTM framework is proposed based on leading indicators. Combining this framework with genetic algorithms, a pair of trading signals can be found. Using this trading signal for stock trading, a certain amount of income can be gained. The data used in the experiments are the Taiwan stock market data (<https://www.twse.com.tw/en/>) from September 28, 2018, to October 30, 2019, and the US stock market data (<https://finance.yahoo.com/>) from September 28, 2018, to August 30, 2019.

In the experiments, a total of 10 stocks were used: 5 stocks in the US market, including Microsoft Corporation

TABLE 5: The historical prices of the five stocks after standardization.

	d_{i_1}	d_{i_2}	d_{i_3}	d_{i_4}	d_{i_5}
CDA	1.42711759	1.36374702	1.28817718	1.29732091	...
CFO	4.4408209	4.2911262	3.92386591	3.58357233	...
DJO	3.48827139	3.75538658	3.65931874	3.92310615	...
DVO	0.136035	0.09801487	0.17144214	0.18939852	...
IJO	0.0073875	0.02757984	0.05991671	0.07369991	...

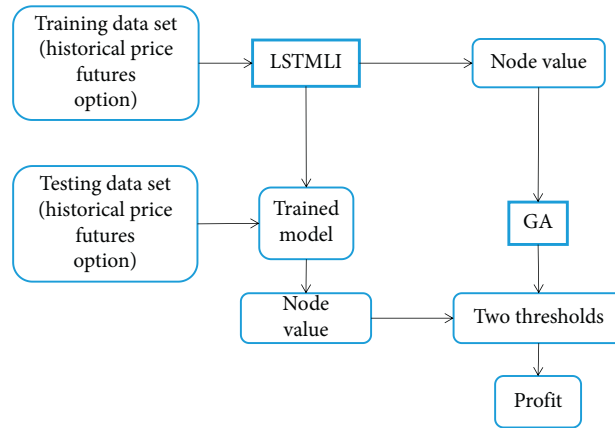


FIGURE 12: The flowchart of the proposed stock trading system.

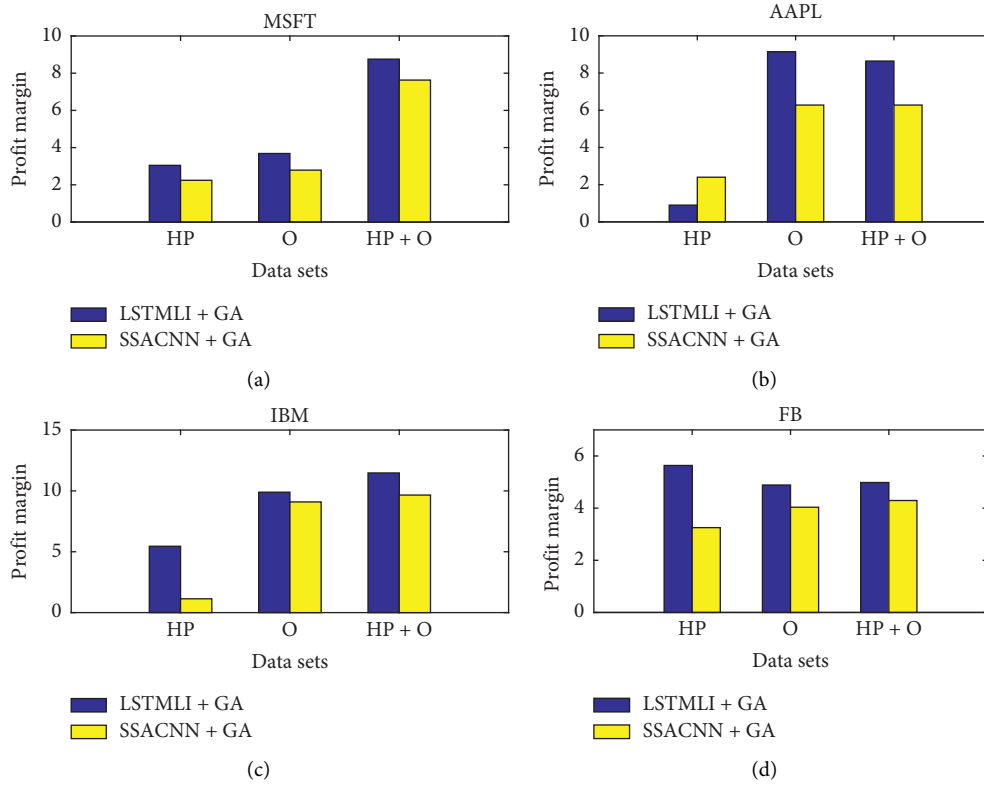


FIGURE 13: Continued.

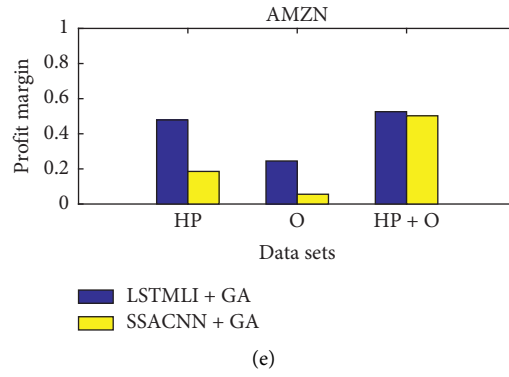


FIGURE 13: On the three data sets of the US market, the annual returns of five stocks are calculated based on the proposed framework: the annual rate of return of (a) MSFT stock, (b) AAPL stock, (c) IBM stock, (d) FB stock, and (e) AMZN stock.

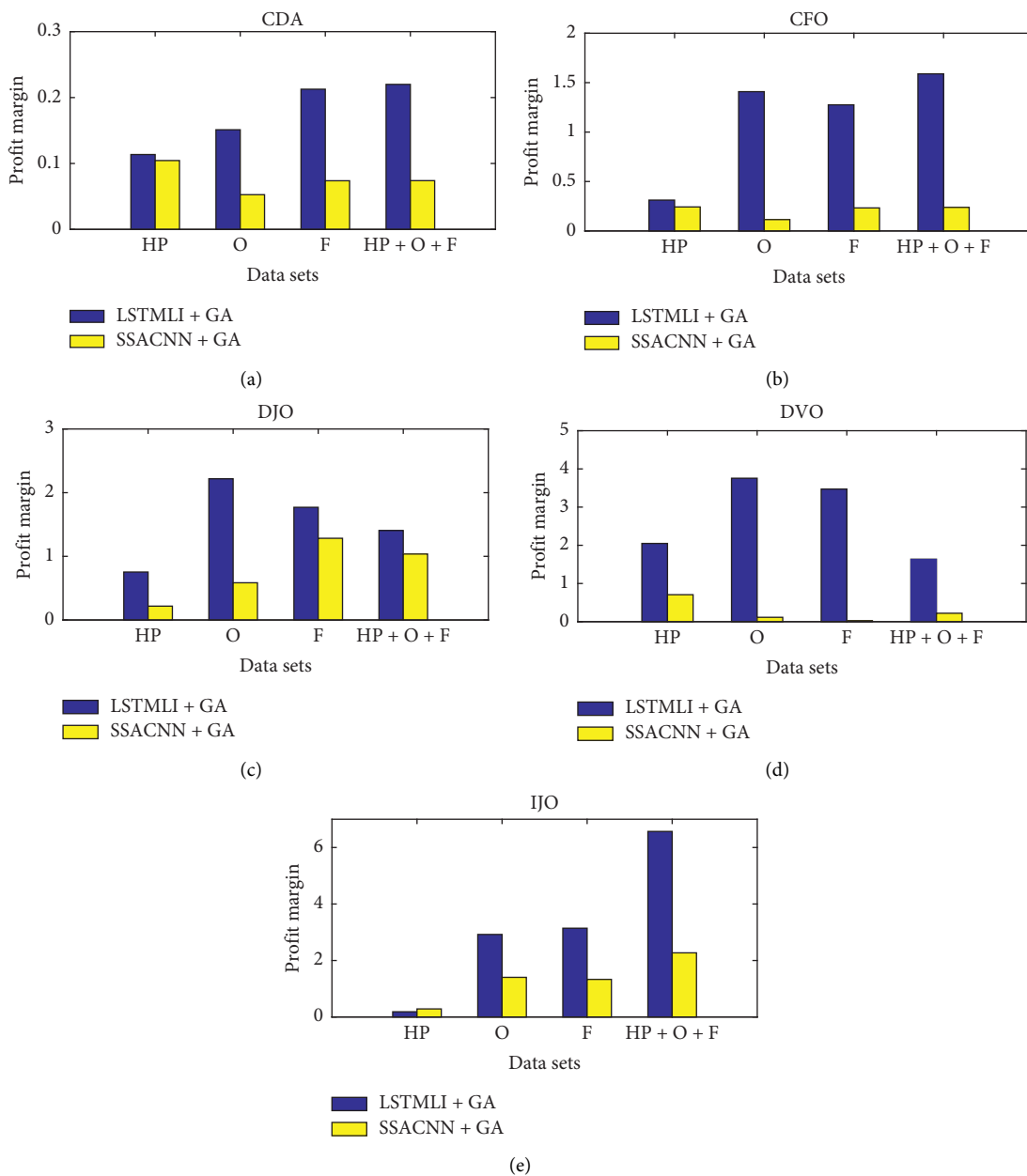


FIGURE 14: On the four data sets of the Taiwan market, the annual returns of five stocks are calculated based on the proposed framework: the annual rate of return of (a) CDA stock, (b) CFO stock, (c) DJO stock, (d) DVO stock, and (e) IJO stock.

(MSFT), International Business Machines, Facebook, Inc. (FB), Amazon.com, Inc. (AMZN), and Apple Inc. (AAPL), and 5 stocks in the Taiwan market, including MediaTek (DVO), Formosa Plastics (CFO), TSMC (CDA), Asustek (DJO), and Largan (IJO). In the experiments, three data sets are used, which are historical stock prices, futures, and options. The data set is divided into two parts: one part is used for training and the other part is used for testing. This research compares the proposed stock trading system with another trading strategy (SSACNN combined with GA). We assume that the initial funding is 100,000 USD in the designed trading system. When a buy signal appears, all funds are used to buy stocks, and when a sell signal appears, all stocks are sold. The specific experimental process is shown below.

First, a study has been conducted on five stocks in the US stock market, and the results are shown in Figures 13(a)–13(e). Among them, the abscissa represents the data set; “HP” represents historical prices; “O” represents options; and “HP + O” represents historical prices and options. Because there is no leading indicator of futures in the US stock market, only two data sets such as historical prices and options are used. In addition, the two data sets of historical price and option rights are combined to form a third data set, namely, “HP + O”. The ordinate represents the cumulative income during the test period. Experimental results show that our proposed stock trading system is superior to the trading strategy composed of SSACNN and GA. It can be seen from these five figures that in general, the profit margin obtained by using LSTMli is higher than that obtained by using SSACNN. On the AAPL stock option data set, the profit margin obtained by the LSTMli model reached 31.7%, which is about 10% higher than the result obtained by the SSACNN model. Next, the experiments will compare a data set with leading indicators and a data set with only historical prices. In general, whether it is LSTMli or SSACNN, the profit margin obtained by using the data set with the leading indicator is higher than the profit margin obtained by the data set with only historical prices.

Then, we conduct the experiments with the five stocks in the Taiwan stock market, and the results are then shown in Figures 14(a)–14(e). In the experiments of five stocks in Taiwan, data sets such as historical price (HP), futures (F), and option (O) were used. Besides, the three data sets are combined to form the fourth data set “HP + O + F”. It can be seen from the figures that the stock trading system proposed in this study can obtain relatively good returns. The experimental results show that the profit margin obtained by the LSTMli framework is much higher than that of SSACNN. Overall, the experimental results obtained using the leading indicator data set are higher than the experimental results obtained using only the historical price data set.

6. Conclusion

This paper mainly proposes a stock trading system, which is composed of a long short-term memory neural network with leading index (LSTMli) and GA. We first propose a new

architecture with leading indicators, LSTMli, based on the LSTM model. Compared with SSACNN, this model can obtain higher benefits. Besides, genetic algorithms are used to find the optimal value of stock trading signals. In the input of LSTMli architecture to reduce useless information, the designed model also integrates the data into a matrix and then uses the matrix as input instead of inputting it into the model one by one. In other words, the input to the model in the paper is an image. Besides, the output of the model adopts a classification method to divide the stock price into three categories: +0.01, 0, and -0.01, that is, the predicted stock price is no longer a specific value but a category. The output of the LSTMli architecture is three nodes. In this study, only the first and the third nodes are used, that is, only the buy and the sell signals are used. The experimental results show that the proposed stock trading system can help investors obtain certain returns within a period of time. Leading indicators can help researchers predict stock trends more effectively. This study only uses the two leading indicators of futures and options, and many factors affect the stock market, which requires further research. After that, we will further study and improve this stock trading system for further extension.

Data Availability

The data used in the experiments are available at <https://www.twse.com.tw/en/> for the Taiwan stock market and at <https://finance.yahoo.com/> for the US stock market.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially supported by the Shandong Provincial Natural Science Foundation (ZR201911150391) and the Western Norway University of Applied Sciences, Norway.

References

- [1] V. Dineshreddy and G. R. Gangadharan, “Towards an “internet of things” framework for financial services sector,” in *Proceedings of the 2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pp. 1–5, Dhanbad, India, July 2016.
- [2] J. M. T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C. W. Lin, “A graph-based CNN-LSTM stock price prediction algorithm with leading indicators,” *Multimedia Systems*, 2021.
- [3] W. Chen, M. Jiang, W.-G. Zhang, and Z. Chen, “A novel graph convolutional feature based convolutional neural network for stock trend prediction,” *Information Sciences*, vol. 556, pp. 67–94, 2021.
- [4] P. Y. Hao, C. F. Fung, C. Y. Chang, and J. B. Ou, “Predicting stock price trends based on financial news articles and using a novel twin support vector machine with fuzzy hyperplane,” *Applied Soft Computing*, vol. 98, Article ID 106806, 2021.
- [5] H. Li, J. Hua, J. Li, and G. Li, “Stock forecasting model FS-LSTM based on the 5G internet of things,” *Wireless*

- Communications and Mobile Computing*, vol. 2020, Article ID 7681209, 7 pages, 2020.
- [6] J. Yang, J. Li, and S. Liu, "A new algorithm of stock data mining in internet of multimedia things," *The Journal of Supercomputing*, vol. 76, no. 4, pp. 2374–2389, 2020.
 - [7] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLoS One*, vol. 14, no. 2, Article ID e0212320, 2019.
 - [8] P. A. Samuelson, "Lifetime portfolio selection by dynamic stochastic programming," *The Review of Economics and Statistics*, vol. 51, pp. 239–246, 1969.
 - [9] C. Brooks, "Predicting stock index volatility: can market volume help?" *Journal of Forecasting*, vol. 17, no. 1, pp. 59–80, 1998.
 - [10] D. B. Keim and R. F. Stambaugh, "Predicting returns in the stock and bond markets," *Journal of Financial Economics*, vol. 17, no. 2, pp. 357–390, 1986.
 - [11] J. Z. Git Hiew, X. Huang, H. Mou et al., "BERT-based financial sentiment index and LSTM-based stock return predictability," 2019, <https://arxiv.org/abs/1906.09024>.
 - [12] M. Baker and J. Wurgler, "Investor sentiment and the cross-section of stock returns," *The Journal of Finance*, vol. 61, no. 4, pp. 1645–1680, 2006.
 - [13] C. M. C. Lee, A. Shleifer, and R. H. Thaler, "Investor sentiment and the closed-end fund puzzle," *The Journal of Finance*, vol. 46, no. 1, pp. 75–109, 1991.
 - [14] B. Han, "Investor sentiment and option prices," *Review of Financial Studies*, vol. 21, no. 1, pp. 387–414, 2008.
 - [15] T. T.-L. Chong, B. Cao, and W.-K. Wong, "A new principal-component approach to measure the investor sentiment," *IGEF Working Paper*, no. 24, pp. 1–19, 2014.
 - [16] A. Cowles, "Can stock market forecasters forecast?" *Econometrica*, vol. 1, no. 3, pp. 309–324, 1933.
 - [17] S. Borovkova and I. Tsiamas, "An ensemble of LSTM neural networks for high-frequency stock market classification," *Journal of Forecasting*, vol. 38, no. 6, pp. 600–619, 2019.
 - [18] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS One*, vol. 12, no. 7, Article ID e0180944, 2017.
 - [19] B. B. Hazarika and D. Gupta, "Density-weighted support vector machines for binary class imbalance learning," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4243–4261, 2021.
 - [20] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.
 - [21] R. K. Nayak, D. Mishra, and A. K. Rath, "A naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices," *Applied Soft Computing*, vol. 35, pp. 670–680, 2015.
 - [22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
 - [23] A. F. Sheta, S. E. M. Ahmed, and H. Faris, "A comparison between regression, artificial neural networks and support vector machines for predicting stock market index," *Soft Computing*, vol. 7, no. 8, 2015.
 - [24] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–6, San Diego, CA, USA, June 1990.
 - [25] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
 - [26] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.
 - [27] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," , 1999.
 - [28] K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction: a case study of China stock market," in *Proceedings of the IEEE International Conference on Big Data*, pp. 2823–2824, Santa Clara, CA, USA, November 2015.
 - [29] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
 - [30] N. Naik and B. R. Mohan, "Study of stock return predictions using recurrent neural networks with LSTM," in *Proceedings of the International Conference on Engineering Applications of Neural Networks*, pp. 453–459, Xersonisos, Greece, 2019.
 - [31] H. Gunduz, Y. Yaslan, and Z. Cataltepe, "Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations," *Knowledge-Based Systems*, vol. 137, pp. 138–148, 2017.
 - [32] 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.
 - [33] W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, "Leveraging social media news to predict stock index movement using RNN-boost," *Data & Knowledge Engineering*, vol. 118, pp. 14–24, 2018.
 - [34] S. A. El-Moneim, M. A. Nassar, M. I. Dessouky et al., "Text-independent speaker recognition using LSTM-RNN and speech enhancement," *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 24013–24028, 2020.
 - [35] S. Abujar, A. Kaisar Mohammad Masum, S. M. Mazharul Hoque Chowdhury, M. Hasan, and S. A. Hossain, "Bengali text generation using bi-directional RNN," in *Proceedings of the International Conference on Computing, Communication and Networking Technologies*, pp. 1–5, New York, NY, USA, November 2019.
 - [36] H. Je Song, A. Yeong Kim, and S. B. Park, "Translation of natural language query into keyword query using a RNN encoder-decoder," in *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pp. 965–968, Tokyo, Japan, August 2017.
 - [37] W. Yeoh, Y. Jhang, S. Kuo, and Y. Chou, "Automatic stock trading system combined with short selling using moving average and GQTS algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1570–1575, Miyazaki, Japan, October 2018.
 - [38] L. Zheng, Y. Jiang, and H. Long, "Exchange rates change, asset-denominated currency difference and stock price fluctuation," *Applied Economics*, vol. 51, no. 60, pp. 6517–6534, 2019.
 - [39] S. Heston and N. Sinha, "News versus sentiment: comparing textual processing approaches for predicting stock returns," *SSRN Electronic Journal*, vol. 1, 2013.
 - [40] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, and H. Fujita, "Adaptive stock trading strategies with deep reinforcement learning methods," *Information Sciences*, vol. 538, pp. 142–158, 2020.

- [41] S. Sun, S. Wang, and Y. Wei, "A new ensemble deep learning approach for exchange rates forecasting and trading," *Advanced Engineering Informatics*, vol. 46, Article ID 101160, 2020.
- [42] F. Allen and R. Karjalainen, "Using genetic algorithms to find technical trading rules," *Journal of Financial Economics*, vol. 51, no. 2, pp. 245–271, 1999.
- [43] C. Schoreels, B. Logan, and J. M. Garibaldi, "Agent based genetic algorithm employing financial technical analysis for making trading decisions using historical equity market data," in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 421–424, Beijing, China, 2004.
- [44] A. Hirabayashi, C. Aranha, and H. Iba, "Optimization of the trading rule in foreign exchange using genetic algorithm," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, pp. 1529–1536, Montreal, Canada, July 2009.
- [45] L. Lin, L. Cao, J. Wang, and C. Zhang, "The applications of genetic algorithms in stock market data mining optimisation," *WIT Transactions on Information and Communication Technologies*, vol. 33, pp. 273–280, 2004.
- [46] L. Di Persio and O. Honchar, "Artificial neural networks architectures for stock price prediction: comparisons and applications," vol. 10, no. 1, pp. 403–413, 2016.
- [47] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, Anchorage, AK, USA, 2017.
- [48] J. Ming-Tai Wu, Z. Li, G. Srivastava, M.-H. Tasi, and J. C.-W. Lin, "A graph-based convolutional neural network stock price prediction with leading indicators," *Software: Practice and Experience*, 2020.
- [49] D. P. Kingma and J. Ba, "ADAM: a method for stochastic optimization," 2014, <https://arxiv.org/abs/1412.6980>.