Multimedia Systems
https://doi.org/10.1007/s00530-021-00758-w

SPECIAL ISSUE PAPER

A graph-based CNN-LSTM stock price prediction algorithm with leading indicators

Jimmy Ming-Tai Wu[1] · Zhongcui Li[1] · Norbert Herencsar[2] · Bay Vo[3] · Jerry Chun-Wei Lin[4]

Received: 14 October 2020 / Accepted: 30 January 2021

Abstract
In today's society, investment wealth management has become a mainstream of the contemporary era. Investment wealth management refers to the use of funds by investors to arrange funds reasonably, for example, savings, bank financial products, bonds, stocks, commodity spots, real estate, gold, art, and many others. Wealth management tools manage and assign families, individuals, enterprises, and institutions to achieve the purpose of increasing and maintaining value to accelerate asset growth. Among them, in investment and financial management, people's favorite product of investment often stocks, because the stock market has great advantages and charm, especially compared with other investment methods. More and more scholars have developed methods of prediction from multiple angles for the stock market. According to the feature of financial time series and the task of price prediction, this article proposes a new framework structure to achieve a more accurate prediction of the stock price, which combines Convolution Neural Network (CNN) and Long–Short-Term Memory Neural Network (LSTM). This new method is aptly named stock sequence array convolutional LSTM (SACLSTM). It constructs a sequence array of historical data and its leading indicators (options and futures), and uses the array as the input image of the CNN framework, and extracts certain feature vectors through the convolutional layer and the layer of pooling, and as the input vector of LSTM, and takes ten stocks in U.S.A and Taiwan as the experimental data. Compared with previous methods, the prediction performance of the proposed algorithm in this article leads to better results when compared directly.

Keywords Convolution neural network · Long–short-term memory neural network · Stock price prediction · Leading indicators

✉ Jerry Chun-Wei Lin
jerrylin@ieee.org

Jimmy Ming-Tai Wu
wmt@wmt35.idv.tw

Zhongcui Li
17685458562@163.com

Norbert Herencsar
herencsn@feec.vutbr.cz

Bay Vo
vd.bay@hutech.edu.vn

[1] Shandong University of Science and Technology, Qingdao, China

[2] Department of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic

[3] Faculty of Information Technology, Ho Chi Minh City University of Technology, Ho Chi Minh, Vietnam

[4] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway

1 Introduction

The capital market includes money as well as the market, namely the financial market. So financing refers to the process of economic operation; both supply and demand of funds use various financial instruments to adjust the capital surplus of activity. Financial markets are trading financial instruments, such as bonds, savings certificates, stock, etc. To seek greater benefits from it, generations of scholars and investors continue to explore the secrets and develop many prediction methods [1, 2]. Especially in stocks, because fluctuations in stock prices are influenced by many factors, including economic trends, economic cycles, economic structure, and other macro factors, as well as industry development, listed companies' financial quality, and other factors. Even great in part, it depends on the influence of investors' psychological games and other micro-factors. These factors are important in stocks. Researchers and investors are trying to find opportunities to gain profit in stocks.

Published online: 22 February 2021

Therefore, the stock's methods forecasting is widely spread. [3–7]. As we all know, traditional methods include linear discriminant analysis, statistical methods, random forests [8], quadratic discriminant analysis, evolutionary computation algorithms [9, 10], logistic regression and genetic algorithm [11–13]. Among them, the stochastic forest algorithm is used to build the stock model based on the historical price information [8] for the trend prediction in the process of stock investment. First of all, taking the stock portfolio as the research object, genetic programming is used to derive a more accurate prediction function. Secondly, the genetic algorithm is used in the possible stock permutation and combination to perform the random number generation, selection, exchange, mutation, etc. The survival probability of chromosomes is determined according to the profit fitness evaluation, and the better combination is found [11–13]. Establishing a model of the relationship between future price trends and historical behavior, and use the sample in historical market trends to predict future price [5]. However, the key part of many forecasting methods is the extraction of features. They all assume that the future price trend is the result of historical behavior. It is worth noting that people design features subjectively, and models based on technical analysis are generally based on some assumptions of the market framework. The model's success depends mainly on The validity of these assumptions.

With the continuous development of information technology, the application of machine learning is becoming more and more extensive. Especially, an artificial neural network (ANN) which is very important in social life, such as completing some signal processing or pattern recognition function, speech recognition [14], constructing expert system [15], making robot and so on. ANN is a method of simulating human thinking. It is a non-linear dynamic system and his characteristic is the parallel distributed information processing and storage synergistic [16–18]. Although the structure of the artificial neural network is very simple and its functions are limited, it is connected by many neurons with adjustable connection weights. It is characterized by large-scale parallel processing, distributed information memory, good self-organization, and self-learning capabilities. Therefore, rapid judgment, decision-making, and processing can be made on many issues. After all, they are all learning and training with the help of a series of adaptive, and self-organizing abilities of neural networks, so that they have a good and predictive ability. For example, constructing a neural network framework for the financial market, and predict the short-term closing price of the stock through a combination of technical analysis, financial theory and economic analysis, time series analysis, and basic analysis [19]. With the increase of the neural networks, deep learning has attracted more and more heed [20–23]. The difference between ANN and deep learning is that the number of layers of the network is different. Deep learning is to train many layers of neural networks with more layers. A series of new structures and new methods that can be worked and evolved. New structures include a CNN, LSTM, ResNet, etc. There are different units in CNN and LSTM, and there are mainly convolutional units [24] and the unit of pooling on CNN. There is mainly recurrent unit [25], long-short term memory unit [26] in LSTM [27, 28]. In addition, there are some algorithm, such as, Restricted Boltzmann Machine (RBM) [29, 30], deep multilayer perceptron (MLP) [22], and autoencoder (AE) [31], among others.

The original steps of a multilayer neural network are mapping features to value. It is characterized by manual selection. The step of deep learning is to input the signal at first, then extract the feature, and output the expected value at last. The most important feature is that the network chooses itself. Among them, CNN and LSTM are the most widely used in the network. CNN solves the problem that the traditional deep network parameters are too many and hard to train. It uses the concepts of local receptive fields and shared weights, thus greatly reducing the number of network parameters. Local receptive field refers to the input data of neural networks in a multi-dimensional vector, and the neurons of the next layer will only be connected to the input neurons under part of the window. Weight sharing means that $N \times N$ hidden layer neurons connected with the layer of inputting, the parameters of each hidden layer's neurons are not different, that is to say, different windows and their corresponding hidden layer neurons share a set of parameters. With the above characteristics of CNN, many researchers apply it to the prediction of stock [3, 32–35]. LSTM is a special type of recurrent neural network (RNN). It is mainly applied to solve gradient disappearance and explosion during long sequence training. As we all know, RNN is a neural network used to process sequence data. It is mainly used to process data that changes in sequence. For example, according to the previous content, words will have different meanings, and RNN can solve such problems well. In simple terms, LSTM can perform better than ordinary RNN in longer sequences. Similarly, many scholars use LSTM to relate to time series and believe that applying it to the financial market will achieve good results [36–38]. In [34], it uses the stock candlestick chart as the input image, and directly input to the input layer. Another study is in [33], it used to map the market's historical data to its future volatility to seek a framework. To reduce overfitting, the establishment of a one-dimensional input to predict only be achieved by the CNN framework in [32]. At the same time, it is based on the closing price history, but ignore other possible variables, such as technical indicators. Based on the above shortcomings, [3] proposed another CNN based model that used technical indicators for each sample. However, it can take into account the possible interrelationship as another probable source of message between the stock market. In [32], historical data of the closing price of the S&P 500 index as the input of LSTM, MLP, and CNN, and the result shown that the experimental results of CNN and LSTM are better than MLP. In [37], it proposed a model to predict the stock rankings future returns. [38] analyzed LSTM that is a special RNN structure, suitable for learning from experience, classification, processing, and prediction of time series with unknown size delay, and [36] propose a framework to analyze and forecast the company's future growth using the LSTM model and the company's net growth algorithm.

At present, in the aspect of financial time series, especially for stocks, the feature vectors are usually the closing price, the opening price, the lowest price, the highest price, and the transaction volume of historical data. And then the data is predicted by inputting them into relevant algorithms. In this paper, the historical data and the leading indicators of stocks are used to predict stocks. Leading indicators' index is changed firstly before the overall economic recession or growth. It can predict the turning point in the economic cycle, and estimate the fluctuation range of economic activities, and speculate the trend of economic fluctuation. From an economic perspective, this indicator has a clear and positive leading relationship with its benchmark cycle, such as inflation rate, futures, and options. This article will propose a new neural network framework, and this method is called stock sequence array convolution LSTM (SACLSTM) whose input method is mainly to imitate the elements of CNN network image input and confirm to the currently existing data into the form of an image. Then through these image training network weights to extract useful features, the extracted features are input into the LSTM network to identify the extreme market to generate appropriate signals. The historical data and leading indicators of Taiwanese and American stock markets are used as input data, and the input vector of the proposed network is transformed by preprocessing. In the experiment, this paper combines different types of data and different indicators to test and get the appropriate signal. On this basis, this paper tests a simple commercial strategy based on prediction to evaluate the profitability and stability of the network. Some contributions are described in this article as:

1. Because stocks are susceptible to various factors, it is necessary to collect more reference indicators. Based on the operating principles of CNN and LSTM, the input of the initial variable covers the historical data and the leading indicators of the stock, and it can further reach other places of the stock, and then realize the prediction of the data.
2. This paper firstly proposes a two-dimensional vector that simulates the image input style and then inputs it into the proposed network to predict stocks. This input framework makes stock price prediction possible.
3. In this article, the proposed arithmetic will be compared with the previous algorithm, collect some data, and experiment to verify the usefulness of the algorithm.

The others of this article are arranged as follows. The second part firstly introduces the related work of stock forecasting, and then summarizes the methods and applications of related algorithms. The third part will briefly introduce the background knowledge of related technologies in this field. The fourth part introduces the proposed methods. The fifth part will introduce experimental results of classification and market simulation. The sixth part shows the experimental results, and finally, the last part summarizes the experimental results and puts forward suggestions and shortcomings for further improvement.

2 Related work

Financial time series is a type of time series data [39]. They have a strong temporarily. The data has a strong dependency before and after the data, and the order cannot be adjusted. Generally, they are two-dimensional data. Since time series have strong sequence lines, and there are generally dependencies, periods, etc. before and after the data, the future data can be predicted according to the existing data through statistical knowledge. Rani and Sikka [1] believe that time series clustering is one of the key concepts of data mining. They are applied to comprehend the mechanism of generating time series and predict the future value of a given time series. There are usually two different methods for financial sequence prediction. The first algorithm attempts to improve the capability of the prediction by improving the model, and it focuses on the characteristics of the improved prediction.

In the first category of algorithms focused on predictive models, a variety of means have been applied, including ANN, Naive Bayes, support vector machines (SVM), and random forests. In [40], the latest literature reviews the techniques used to predict stock market trends in the area of artificial intelligence and machine learning. ANN is considered the main machine learning technology in the field. It explores the possibility of using the non-linearity of daily returns to improve short-term and long-term stock forecasts. They have compared five competitive patterns, namely LSTAR, the linear AR model, and ESTAR smooth transition autoregressive model, and JCN and MLP. The results showed that the nonlinear neural network model may be a better prediction method. The use of artificial neural network models can more accurately predict stock returns, and neural network technology has certain improvements in predicting stock returns relative to AR models and STAR models.

Feedforward neural networks are currently popular neural network types, usually using back propagation algorithms [41]. Zhang et al. [42] proposes a PSO-based selective neural network integration (PSOSEN) algorithm, which can be used for Nasdaq 100 index and S and P 300 index analysis. The algorithm firstly trains each neural network through the PSO algorithm and then combines the neural networks according to a preset threshold. Experimental results show that the improved arithmetic is valid on the stock index prediction problem, and its performance is powerful than the selection integration algorithm based on a genetic algorithm.

In [2], this article presents an efficient and complete data mining method. Three-degree dimensionality reduction fuzzy robust principal component analysis (FRPCA), techniques of principal component analysis (PCA), and kernel-based principal component analysis (KPCA) that are used to the entire data set to rearrange and simplify the original data structure. Then using artificial neural networks (ANN) is to classify the converted data sets and to predict the daily direction of future market returns. The results show that the risk-adjusted profit of the trading strategy based on the comprehensive classification and mining process of PCA and ANN is significantly higher than the comparison benchmark and higher than the trading strategy predicted based on the KPCA and FRPCA models.

The simplicity of shallow models prevents them from achieving an effective mapping from input space to successful prediction. Therefore, using the availability of many data and emerging effective learning methods is to train deep models. Investigators have turned to market prediction. A significant aspect of a deep model is that they are often able to extract predictions and rich features from the original data. From this perspective, the depth model usually combines the feature extraction stage and single-stage prediction.

Deep ANN is one of the earliest depth methods in this field. It is a neural network with multiple hidden layers. Long et al. [7] is a new model, multi-filter neural network (MFNN), for the task of sample price movement and feature extraction prediction of financial time series. By fusing convolution and recursive neurons, a multi-filter structure is constructed to obtain information on not the same market views and feature spaces. The neural network is used for signal-based trading simulation and the extreme market prediction of the CSI 300 index.

An RNN is a specially designed neural network with internal memory, which can make predictions and extract historical features based on it. Therefore, they seem to be suitable for market forecasting. LSTM is one of the most popular types of RNN. Graves et al. [14] and Pan et al. [43] proposed an LSTM method to obtain useful information and predict immature stock markets from financial time series. In [44], they provide technical indicators to LSTM to achieve the forecast of the stock price direction of the Brazilian stock market. The results show that LSTM is better than MLP.

CNN is another deep learning algorithm applied to stock market prediction after MLP and LSTM, and its effective feature extraction ability has also been verified in many other fields. In [3, 32, 34], the CNN and other algorithms are used to measure the same set of data. Experiments show that the CNN prediction results are ideal.

According to some reported experiments, CNN has a significant role in the input data processing method on the quality of the final prediction and the extracted feature set. For example, [33] can be applied to data sets from different sources, including different markets, and extract features to predict the future of these markets. The evaluation results showed that compared with the most advanced baseline algorithm, the prediction performance is significantly improved.

In these models, some specific methods are based on feature extraction and selection. Due to the high uncertainty and volatility of stocks, the traditional feature extraction methods are technical analysis and statistical methods [4]. Technical analysis is the most direct and basic method in stock forecasting. In [45], people consider that some historical patterns are related to future behavior. Therefore, a large number of technical indicators have been defined to describe these models to be applied to investment expert systems.

The above summarizes the interpretation paper from the initial variable set, feature extraction algorithm, prediction method, and so on. Because these algorithms can automatically extract features from the original data, a trend toward deep learning models has appeared in recent publications. However, most researchers use only one market's technical indicators or historical price data to make predictions, and multiple variables can improve the accuracy of stock market forecasts. In this article, we will introduce a new framework based on CNN and LSTM, which aims to aggregate multiple variables (historical data and leading indicators), automatically extract features through CNN, and then input them into LSTM to predict the direction of the stock market.

3 Background

Before introducing the method proposed in this article, in this section, we will review the CNN and the LSTM as the main elements of the framework of the proposed algorithm.

3.1 Convolutional neural network

With the development of DNN, convolutional neural network has been proposed [3, 32–34] and is currently one of the most famous algorithms. It has successfully been applied in various fields such as detection [46] and segmentation [47]. CNN provided outstanding performance than the previous traditional machine learning algorithms in the above fields. It mainly includes several convolutional layers, pooling layers, and fully connected layers. The details of each component in CNN are introduced as follows.

3.1.1 Convolutional layer

The function of the convolution layer is to collect the features of input data, which contains several convolution kernels. Each cell of the convolution kernel corresponds to a bias vector and a weight coefficient, similar to a neuron of the feedforward neural network. Each neuron in the convolutional layer is related to multiple neurons in the area close to the previous layer. The size of the area depends on the size of the convolution kernel that is called "receptive field" in the literature. Its meaning can be compared to the receptive field of visual cortical cells. During the convolution kernel is working, the input features will be scanned regularly, and the input features will be summed and multiplied by the matrix elements in the receptive field, and the deviations are superimposed:

$$\begin{aligned}Y^{l+1}(c,d) &= [Y^l \bigotimes w^{l+1}](c,d) + b\\ &= \sum_{K_l}^{f=1}\sum_{f}^{e=1}\sum_{f}^{y=1}[Y_l^k(s_0c+x, s_0d+y)w_{l+1}^k(e,f)] + b\end{aligned} \tag{1}$$

$$(c,d)\epsilon\left\{0,1,\ldots,Z_{l+1}\right\}, Z_{l+1} = \frac{Z_l+2q-f}{s_0}+1 \tag{2}$$

The summation part of Eq. (1) is equivalent to solving a cross-correlation. b is a deviation, Y^l and Y^{l+1} represent the input and output of the convolution of $l+1$ layer, also known as the feature map. In Eq. (2), $Y+1$ is the size of Y^{l+1}, it is assumed that the feature maps have the same length and width here. $Y(c, d)$ corresponds to the pixels of the feature map. K is the channel number of the characteristic graph. s_0, f, q are the parameters of the convolution layer, corresponding to the size of convolution step (stride), the convolution kernel, and padding (padding) layers.

Figure 1 takes the two-dimensional convolution kernel as an example, and the one dimensional or three-dimensional convolution kernel works similarly. In particular, when the convolution kernel is of size $f = 1$, step size $s_0 = 1$ and does not contain the filled unit convolution kernel, the cross-correlation calculation in the convolution layer is equivalent to matrix multiplication, and a fully connected network is constructed between the convolution layers. At this time, the output of layer $l+1$ is Eq. (3)

$$Y^{l+1} = \sum_{K_l}^{k=1}\sum_{f}^{e=1}\sum_{f}^{f=1}\left(Y^l_{c,d,k}w^k_{l+1}\right) + b = w_T^{l+1}Y_{l+1} + b, Y^{l+1} = Y \tag{3}$$

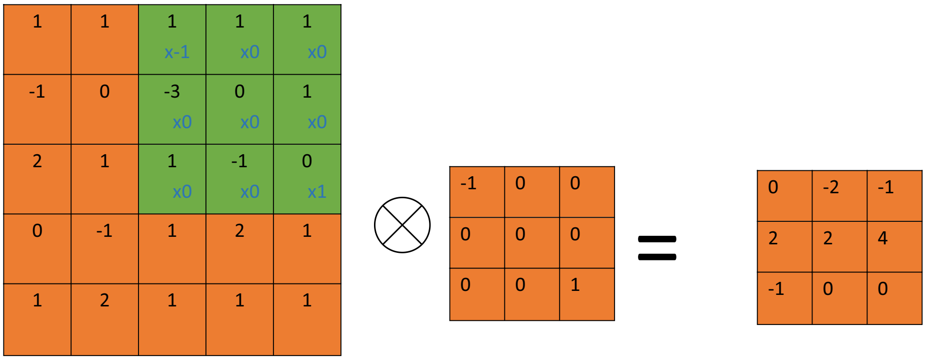
The parameters of the convolution layer include the size of the convolution kernel, the step size, and the filling. The three determine the size of the output feature map of the convolution layer that is the hyperparameter of the CNN. The size of the convolution kernel is specified as any value smaller than the size of the inputting of the image. The convolution kernel is larger, the extracted input features are more complex.

When it sweeps the feature map twice, the convolution step describes the distance between the positions of the convolution kernel. The convolution kernel will sweep through the elements of the feature map one by one when the convolution step is 1, and they will skip $n-1$ pixels in the next scan when the step size is n.

The convolutional layer contains an activation function that helps express complex features, and its representation is described in Eq. (4).

$$A^l_{c,d,k} = f\left(Y^l_{c,d,k}\right) \tag{4}$$

Among them, the common activation function is Relu, which usually refers to the nonlinear function represented by its variants and the ramp function. It is defined as Eq. (5):



1	1	1 x-1	1 x0	1 x0
-1	0	-3 x0	0 x0	1 x0
2	1	1 x0	-1 x0	0 x1
0	-1	1	2	1
1	2	1	1	1

⊗

-1	0	0
0	0	0
0	0	1

=

0	-2	-1
2	2	4
-1	0	0

Fig. 1 Example of the convolution

$$f(e) = \max(0, e) \tag{5}$$

As the activation function of the neuron, the nonlinear output of the neuron after the linear transformation $\mathbf{w}^{\mathbf{T}}\mathbf{e} + \mathbf{b}$ is defined. For the input vector, **e** from the previous neural network entering the neuron, using the linear rectification activation function of the neuron will output $\mathbf{max}(\mathbf{0}, \mathbf{w}^{\mathbf{T}}\mathbf{e} + \mathbf{b})$, and it will as the output of the entire neural network as go to the next layer of neurons.

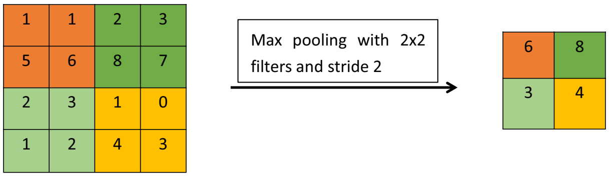
3.1.2 Pooling layer

After feature extraction in the convolutional layer, the output feature map is going to be passed to the pooling layer for information filtering and feature selection. The pooling layer contains preset pooling functions whose function is to replace the result of a single point with the statistics of the feature map of its neighboring area in the feature map. The selection of the pooling area in the pooling layer is not different from the steps of the convolution kernel scanning feature map, controlled by the pooling size, step size, and filling. It is generally expressed as Eq. (6):

$$A_k^l(c,d) = \left[\sum_{x=1}^{f}\sum_{y=1}^{f} A_k^l\left(s_0c + e, s_0d + f\right)^p\right]^{\frac{1}{p}} \tag{6}$$

In Eq. 6, the meaning of step size s_0 and pixel (c, d) is not different as that of the convolutional layer, and s is a prespecified parameter. Pooling takes the average value in the pooling area, which is called average pooling when $p = 1$. When $p \to \infty$, pooling takes the maximum value in the area which is called max pooling. Mean pooling and maximum pooling are pooling methods that have been used for a long time in the design of CNN. Both retain texture information and the background and of the image at the size of the feature map or the expense of partial information. For example, Fig. 2 is a process of max pooling, the size of filters is set as 2×2 and each stride is set as 2. After the pooling process, it can be seen that the original 4 ×4 matrix is compressed into a 2×2 matrix.

As the convolution layers are stacked, according to the cross-correlation calculation of the convolution kernel, the size of the feature map will gradually reduce. For example, an input image of 16×16 undergoes a convolution kernel of 5×5 with unit steps and no padding. After that, a feature map of 12×12 will be output. For this reason, the filling is a method to increase the size of the feature map to offset the influence of size shrinkage. The common filling methods are filling by replication padding and 0.



1	1	2	3
5	6	8	7
2	3	1	0
1	2	4	3

Max pooling with 2x2 filters and stride 2

6	8
3	4

Fig. 2 Example of the convolution

3.1.3 Fully connected layer

In the traditional feedforward neural network, the fully connected layer in CNN is equal to the hidden layer. The fully connected layer is located in the last part of the hidden layer of CNN and only transmits signals to other fully connected layers. The featured graph loses the spatial topology in the fully connected layer and is expanded into a vector and passes the excitation function. The convolutional layer and pooling layer in the CNN can perform feature extraction on the input data. The role of the fully connected layer is to nonlinearly combine the extracted features to obtain an output. Fully connected layers play the role of ”classifier” in the entire convolutional neural network. The convolutional layer, pooling layer, and activation function layer map the original data to the hidden layer feature space. The fully connected layer plays the role of mapping the learned “distributed feature representation” to the sample label space. The fully connected layer isn’t expected to have features extracting ability, but trying to use the existing high-order features to complete the learning goal.

3.2 Long–short-term memory networks

LSTM is a type of time RNN. It is specially devised to solve the long-term dependence problem of general RNN [36–38, 48, 49]. It has been successfully used in various fields such as machine translation [50], speech recognition [51], image description generation [52], video tagging [53], and financial time series [54]. All RNN has a chain form of the repetitive neural network module. It mainly includes forgetting gate, input gate, and output gate.

3.2.1 Forgetting Gate

The determine useful information is:

$$z_t = \delta\left(E_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{7}$$

Through the input of the current time and the output of the previous time, the cell state is multiplied by the output of the sigmoid function through the sigmoid function. If the sigmoid function outputs 0, the part of the information needs to be forgotten, otherwise, the part of the information continues to be transmitted in the united state. In Eq. (7), z_t is current output value, E_f is the weight of current output, b_f

is a biased of current output, and h_{t-1} is the output value of the previous layer.

3.2.2 Input gate

The confirm updated information is

$$j_t = \delta\left(E_i \cdot \left[h_{t-1}\right] + b_i\right) \tag{8}$$

$$\widetilde{B}_t = \tanh\left(E_B \cdot \left[h_{t-1}, x_t\right] + b_B\right) \tag{9}$$

The gate function is to update the old unit status. The previous forget gate layer determines what information is forgotten or add, and is implemented by the gate layer. The gate composed of the second sigmoid + tan$h$ function determines which information needs to be added to the state. Here, it is divided into two parts. One is that the sigmoid layer determines which values will be updated. This part is the same as the first layer, and the tanh layer will create new information to be added to the state. For example, replacing the subject state of the previous sentence. See Eqs. (8) and (9) for specific formula. j_t and B_t are current output values in the input gate which represents two parts. The input h_{t-1} and x_t at time $t - 1$ are activated by another linear transformation and sigmoid (this is called the input gate) and output j_t. At the same time, after h_{t-1} and x_t are activated by another linear transformation (tanh), they are multiplied by j_t to obtain an intermediate result. This intermediate result is added to the intermediate result of the previous step to get $\widetilde{B}_t$.

3.2.3 Output gate

The output information is

$$p_t = \delta\left(W_p\left[h_{t-1}, x_t\right] + b_p\right) \tag{10}$$

$$h_t = p_t * \tanh(B_t) \tag{11}$$

The front two doors are mainly used to update the status of the penetration line. The third door is used to calculate according to the information on the penetration line and the output of the current input information calculation module. The gate control device is used to control how much the state value $m(t)$ is visible to the outside at time t. What information is updated is still what information needs to be discarded and what information needs to be added. Equationa (10) and (12) are shown, After the input h_{t-1} and x_t at time $t - 1$ are activated by another linear transformation + sigmoid (this is called the output gate), the output p_t, p_t is multiplied by B_t through tanh to obtain h_t.

4 Proposed SACLSTM framework

CNN and LSTM have many parameters, including the number of layers, the number of filters in each layer. The initial indication of the input data, which should be chosen to obtain the desired result. The size of each layer of filters on CNN is very important. Because the filter of 5×5 and 3×3 are very common in the field of image processing. In this paper, it proposed that the size of each filter followed the previous work in the area of image processing. Here, it introduces the architecture of SACLSTM, which is a general stock market prediction framework based on CNN and LSTM. This paper divides the framework into three main steps: input data representation, the extraction of continuous features, and final prediction.

Input data representation SACLSTM uses this information to predict the future of these markets and obtains information from different markets. Its goal is to find an ordinary model that maps historical market data to future fluctuations. The general model of this paper is talking about refers to a model that applies to multiple markets. In this paper, it is assumed that from history to the future market for many real mapping functions are correct. To achieve this goal, this paper needs to plan a single model that can predict the future of the market based on the market's history. However, to extract the required mapping function, this framework needs to be trained by specimens from not the same markets. But in addition to using market history modeling and various other variables (futures, options) as input data, it uses the ten pieces of data that are close to the data variables of the day. In this algorithm, all these messages are aggregated and provided to a designed framework in the form of a two-dimensional tensor.

The extraction of the continuous feature The historical data of each day is represented by a series of variables, for example, the closing price, the opening price, the lowest price, the highest price, and volume. The traditional market forecasting method is to analyze these variables, such as in the form of candlesticks [34], and may predict the future trend of the market by constructing high-level features based on them. The idea behind the first layer design of SACLSTM comes from the recognition of images by CNN. In the first step of SACLSTM, a convolutional layer's task is to merge daily variables into higher-level features to represent each day in history. Some useful messages from the trend of the market over time may also have a certain effect on predicting the future behavior of the market. This information may find patterns and provide us with information on market behavior trends, and they can be used to predict the future trends. Therefore, it is important that it combines 30 consecutive days of data variables as a "picture " to collect high-level features that represent trends or reflect market behavior within a specific time interval. The convolutional layer and the layer of pooling in CNN generate more complex features within a certain time interval to summarize the data.

Final prediction The advanced features extracted in the pooling layer and the convolutional layer are input into the LSTM, further operated by the LSTM unit, and finally the flattening operation is used to convert the features generated in the previous layer into a one-dimensional vector, and the vector is provided to map features to predicted fully connected layers. In the next section, this paper will explain the overall design of SACLSTM and how they are used in the dataset this paper used in the specific experiments in this article. In our experiment, this paper used data from 10 stocks from 2 places. In addition to its historical data, options, and futures, each stock also collected ten pieces of data close to its data to better predict the results.

4.1 The process of the proposed SACLSTM

The detailed progress of the developed SACLSTM is described as follows.

Expression of input data As previously earlier, the input of SACLSTM is a two-dimensional matrix. The matrix's size depends on the number of variables, and the number of days used to make a backtracking history of predictions. If the input applied for prediction is g days, and each g day is represented by the f variable, then the size of the input tensor is $g \times f$.

The extraction of the continuous feature In SACLSTM, to extract the 30-days change feature, an initial variable filter is used. The filter of 3×3 most commonly used by CNN in images are adopted, and these filters can be used to combine them into a matrix with more advanced features. It can use this layer to construct different combinations of host variables. The network can also delete worthless variables by setting the corresponding weight to 0 in the filter. Therefore, the layer serves as a feature selection module. The subsequent different effects of convolution and pooling build higher-level features, aggregate available information over a certain period of time, and combine lower level features from their inputs to higher-level inputs. It applies 64 filters in the second layer. Each filter continuously filters for three days. This method is inspired by observations. The most famous candlestick patterns [34], trying to find unique patterns in three subsequent days. This paper uses this setting as a sign to extract the potentially useful messages from a time window of three subsequent time units in collecting data. The pooling layer executes a maximum pooling of 2×2 in the third layer. To build more complex features and aggregate information over longer time intervals, SACLSTM uses other convolutional layer with 128 filters, which is then similar to the first pooling layer another pooling layer. To obtain more accurate information, this paper then used two layers of pooling and convolution, each with 256 filters, the size of the pooling layer is not different from the previous pooling layer.

Final prediction The features produced by the last pooling layer, and are input into the LSTM unit to extract deeper features, and then are tiled into the final feature vector to realize the final prediction. SACLSTM's prediction of the market may be interpreted as the possibility of a price rise on the next day of the market. It is conscious to invest more money in stocks that are more likely to rise. Stocks with a low upside are a good option for short selling. However, in our experiment, this paper discrete the output to 0, 1, or -1, which is closer to the predicted value.

Example configuration of SACLSTM As mentioned earlier, the input of this paper for each prediction includes 30 days, and every 30 days is represented by several variables. The input of 2D-CNNpred is a $30 \times$ variable number matrix. The first convolutional layer uses one filter of 3×3, described by three convolutional layers are 128, 256, 256 filters, each filter is followed by a max-pooling layer of 2×2. It is then inputted into the LSTM unit to generate the final output. Figure 3 describes the visualization of graphical process.

5 Data preprocessing and environment setting

In this section, the paper will show the settings used to assess the model, including data sets, evaluation methods, network parameters, and baseline algorithms. Then, this paper will report the optimization framework.

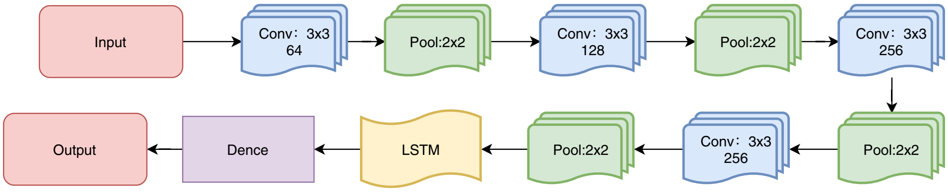


Fig. 3 Visualization of graphical process about the SACLSTM

Table 1 Historical data of the five stocks in Taiwan

m_i	n_{i1}	n_{i2}	n_{i3}	n_{i4}	n_{i5}	n_{i6}
m_1	246.5	244.5	246.5	243	6198	5
m_2	117	117.5	117.5	116	12568	0
m_3	262.5	266	266	260	39645	– 2.5
m_4	264	260	264	259	1284	1.5
m_5	3635	3825	3880	3635	891	– 145

Table 2 Future of the five stocks in Taiwan

m_i	n_{i1}	n_{i2}	n_{i3}	n_{i4}	n_{i5}	n_{i6}
m_1	246	244	246.5	243.5	609	5
m_2	117	117	117.5	117	124	0.5
m_3	262.5	265	265.5	262.5	4771	– 2.5
m_4	263.5	262	263.5	258	34	2.5
m_5	3660	3815	3865	3635	39	– 130

Table 3 Option data of the five stocks in Taiwan

m_i	Call					Put				
	z_{i1}	z_{i2}	z_{i3}	z_{i4}	z_{i5}	z_{i1}	z_{i2}	z_{i3}	z_{i4}	z_{i5}
m_1	20.3	0	0	0	0	0.39	0.26	0.7	1	5
m_2	17	0	0	0	0	0.01	0	0	0	1
m_3	22.7	0	0	0	0	0.42	0.33	0.53	3	26
m_4	8.45	0	0	0	0	0.83	0.62	0.7	1	1
m_5	221	0	0	0	0	130.5	0	0	0	2

Table 4 Historical data of the five stocks in America

t_i	n_{i1}	n_{i2}	n_{i3}	n_{i4}	n_{i5}	n_{i6}
t_1	114.37	114.19	114.57	113.68	20525950	– 0.04
t_2	151.21	150.79	151.625	151.16	3305324	– 0.29
t_3	164.46	168.33	168.79	162.56	33786480	– 4.38
t_4	2003	2004.41	2026.52	1996.4648	3991574	9.98
t_5	225.74	224.79	225.84	224.02	22560440	0.79

Table 5 Option data of the five stocks in America

t_i	Call					Put				
	z_{i1}	z_{i2}	z_{i3}	z_{i4}	z_{i5}	z_{i1}	z_{i2}	z_{i3}	z_{i4}	z_{i5}
t_1	5.34	5.34	5.55	73	1361	0.02	0.01	0.01	38	802
t_2	5.03	– 1.18	5.25	92	219	0.02	– 0.01	0.03	1	378
t_3	14.4	– 2.1	12.8	1	38	0.01	0	0.01	516	2410
t_4	12.95	– 12.7	15.15	1668	1257	0.01	– 1.84	0.09	3759	1529
t_5	12.85	0.2	13.5	200	1301	0.01	– 0.03	0.01	1782	12103

5.1 Dataset

The use of the data set in this work is ten stocks from two markets, namely AAPL, IBM, MSFT, FB, AMZN five stocks in the American market, and CDA, CFO, DJO, DVO, IJO in five stocks of Taiwan. Each sample has several variables (mainly historical data, options, and futures) and the 10 most similar data. Tables 1, 2, 3, 4 and 5 show the relevant information of five stocks in Taiwan and America. Its attributes include the historical data of stock and the attribute of the stock's future and option. It has three main categories, namely, historical data, futures, and options.

In Tables 1 and 2, where m_i is denoted the five Taiwanese stocks, they are *DVO*, *CFO*, *CDA*, *DJO*, *IJO*. In Tables 4 and 5, where t_i is denoted the five American stocks, they are *MSFT*, *IBM*, *FB*, *AMZN*, *AAPL*. $n_{i.}$ are denoted the present price, the highest price, the opening price, the lowest price, volume, and ups and downs. In Tables 3 and 5, because there are two kinds of options: call and put. Where $z_{i.}$ are denoted the settlement price (After the transaction is completed, the transaction margin of the uncleared contract and the profit and loss settlement base price are settled.), ups and downs (The not same between the closing price and the spot price of the day.), volume, closing price and open position (The number of contracts held by multiple parties or short in a particular market at the end of a trading day.). Furthermore, the algorithm selects the 20 options (10 call options and 10 put options) whose contract prices are closest to the current stock price to generate an array of option data.

5.2 Normalization function

Because the size of the data currently used is relatively large, it is a good method to scale the data so that it falls into a small specific interval. This method is called data standardization. The use of data standardization can not only speed up the optimal solution of gradient descent but also improve accuracy. The function is shown in Eq. (12):

$$Y_t = \frac{Z_t - \text{mean}}{\max - \min}, \tag{12}$$

where Z_t is the exponent vector for time t (the highest price, the opening price, the closing price, the lowest price, ...), Y_t is the exponents vector after the normalize evolution. Min max and mean are the minimal value, maximal value and average value of the indexes vector in a some period. The data will be collected in 120 days to establish an input array. Take the value 246.5 of m_1 as an example in Table 1, the mean of the same property was taken for the first 120 days, the highest, the lowest price and use Eq. (12) for calculation, the result is 0.390278. In the same way, after normalization, all normalized data are inputted in the algorithm, The following shows a type of data after normalization in Table 6.

These data are from October 2018 to October 2019. First, 60 percent of data is used to train the model, the next 20 percent from the test data, and the final 20% is the verification data.

5.3 Assess methods

The quality of the results often requires evaluation indicators to compare the results of the proposed algorithm with other algorithms. Accuracy is the most common indicators used in the field. In an unbalanced data set, it might be biased towards models and tended to predict more frequent classes. To solve this problem, this paper defines a formula that roughly divides the data into three categories, The function is shown in Eq. (13).

$$C_t = \begin{cases} +1, & A_t \geq 0.05 \\ 0, & \text{Others} \\ -1, & A_t < -0.05 \end{cases} \tag{13}$$

There, C_t is indicated the label of the sample, A_t is the percentage change in the current stock's price on the next date. When A_t is greater than or equal to 0.05, it is defined as + 1 (price increasing), if A_t is under to −0.05, it is defined as −1 (price decreasing). Otherwise, it is labeled as 0, meaning not to rise or fall in this range.

5.3.1 Detail architectures of input images

In the proposed algorithm of SACLSTM, the architecture of the input image is very complicated. Therefore, this section details the architecture of the input image used in the experiment. First, the images generated by historical prices, options, and futures are explained separately below. Note that the description of the architecture here focuses on relevant specific data. These values will be standardized by the above method.

Historical Price Image According to the above definition, for a specific stock, the picture should contain lowest price, opening price, highest price, closing price and volume in 30 days. We just need to put these values in one column with the same attributes for one day, and expand them in 30 rows for 30 days to build the image. An example is shown in Fig. 4.

Table 6 Normalization of historical data of the five stocks in Taiwan

m_i	n_{i1}	n_{i2}	n_{i3}	n_{i4}	n_{i5}	n_{i6}
m_1	0.622	0.134	0.125	0.134	0.130	0.0828
m_2	0.616	0.589	0.554	0.561	– 0.053	– 0.047
m_3	0.3865	0.356	0.392	0.393	0.132	0.0142
m_4	0.486	0.549	0.560	0.629	0.123	0.115
m_5	0.463	0.531	0.364	0.311	– 0.201	– 0.026

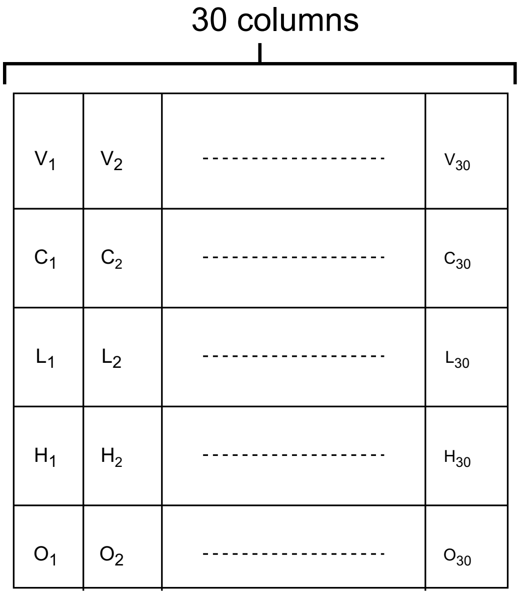

30 columns
V1 V2 ------------------- V30
C1 C2 ------------------- C30
L1 L2 ------------------- L30
H1 H2 ------------------- H30
O1 O2 ------------------- O30

Fig. 4 Example of the image established by historical prices (L_n, O_n, H_n, C_n and V_n are the lowest price, opening price, highest price, closing price and volume at n-th day)

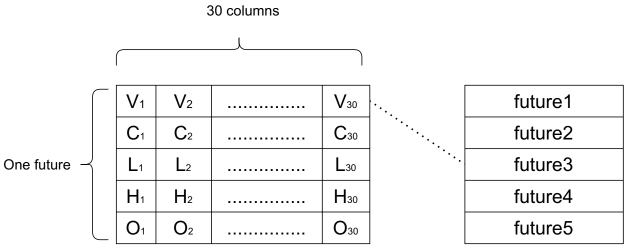

30 columns
One future
V1 V2 .... V30
C1 C2 .... C30
L1 L2 .... L30
H1 H2 .... H30
O1 O2 .... O30
future1
future2
future3
future4
future5

Fig. 5 Example of the image established by futures (L_n, O_n, H_n, C_n and V_n are the lowest price, opening price, highest price, closing price and volume at n-th day)

Futures Image It is similar to historical prices, and also includes lowest price, opening price, highest price, closing price and volume. However, stocks can have several future products with different expiration dates. In this paper, we select five futures with expiry dates closest to the current date. The relevant attributes of each future in a day will be listed in a row and will be extended by 30 days in a matrix. An example is shown in Fig. 5.

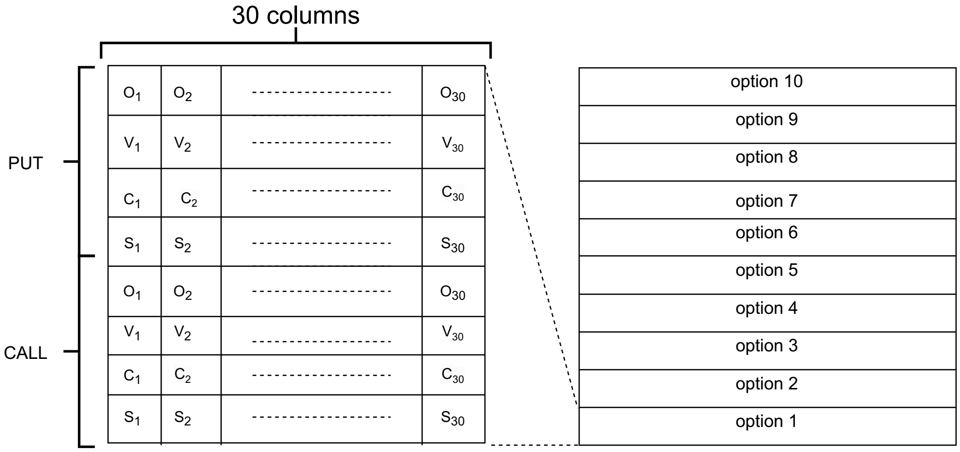
Options Image The attributes of options are more complicated than the above two. Because there are two options in the options market (call options and put options). Here, we only select data from the recent month options for a specific stock. In the proposed method, it selects ten different options (10 call rights and 10 put rights) whose settlement price is closest to the current price of the stock and obtains attributes from these options to construct the image. These attributes include closing price, settlement price, open position and transaction volume. It is similar to the previous two images, with 30 days of data extended as a matrix. An example is shown in Fig. 6.

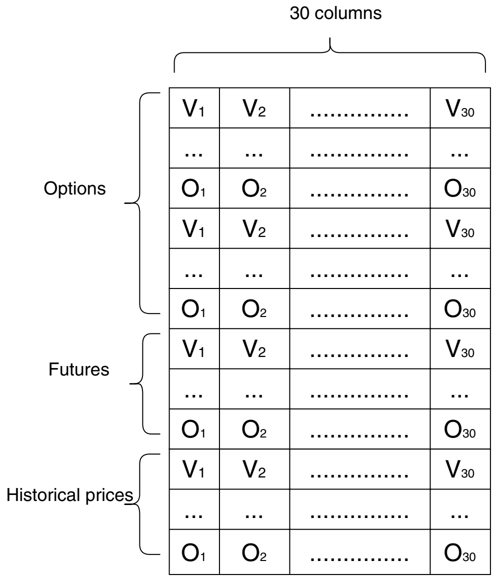
Combination Image The combined image is the final input form of the proposed framework. It combines information about historical prices, futures, and options. It just binds the first three images to generate a new image. An example is shown in Fig. 7.

5.4 Network parameters

With the continuous development of deep learning packages and software, Tensorflow is used to implement CNN and LSTM. And the activation function of each layer in the CNN framework is Relu. Each convolutional layer is composed of 64, 128, 256, and 256 filters respectively. What's more, Adam [55] and LSTM were applied to train network.

Fig. 6 Example of the image established by options (S_n, C_n, V_n and O_n are the settlement price, closing price, volume and open interest at n-th day)


30 columns
PUT
CALL
O1 O2 ------------------- O30
V1 V2 ------------------- V30
C1 C2 ------------------- C30
S1 S2 ------------------- S30
O1 O2 ------------------- O30
V1 V2 ------------------- V30
C1 C2 ------------------- C30
S1 S2 ------------------- S30
option 10
option 9
option 8
option 7
option 6
option 5
option 4
option 3
option 2
option 1



	30 columns			
Options	V_1	V_2	....	V_{30}
	...	...	....	...
	O_1	O_2	....	O_{30}
	V_1	V_2	....	V_{30}
	...	...	....	...
	O_1	O_2	....	O_{30}
Futures	V_1	V_2	....	V_{30}
	...	...	....	...
	O_1	O_2	....	O_{30}
Historical prices	V_1	V_2	....	V_{30}
	...	...	....	...
	O_1	O_2	....	O_{30}

Fig. 7 Example of the image established by combination data

5.5 Baseline algorithms

This paper compares the capability of the proposed method with algorithm used in subsequent research.

1. Siripurapu proposed the CNN-corr algorithm [34] that uses a stock candlestick chart as an input image and directly input to the input layer.
2. Hoseinzade and Haratizadeh [33] use the CNNpred algorithm to seek out a common framework and map the market's historical data to its future fluctuations.
3. Support vector machine (SVM) is proposed by Zhong [2] that builds a stock selection model, which can classify stocks non-linearly.
4. The indexes are applied to train a simple ANN for prediction.

5.6 Optimization framework

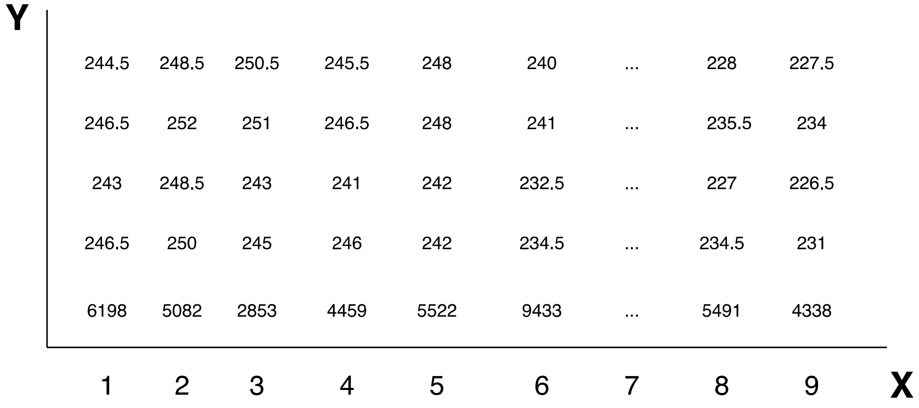
This article will collect stock index vector information within 30 days to generate an input image. The *x*-axis indicates the date of the continuous cycle of the input image. The *y*-axis indicates the index of the historical data set of stocks on these dates of the input image. An example is described in Fig. 8.

In the experiment, a sliding window of predetermined herein stock index in a width sequence for 30 days. Each window generates an input image, you can move the date to get the next image from the present window. At last, the method may obtain a series of input images. Two adjacent images indicate that their sliding window is not the same way to place the day.

The algorithm is based on CNN and LSTM. First, this feature of the convolutional neural network of Gunduz and Siripurapu [3, 34] and others are used to convert the data into an image. In addition to pooling and LSTM operations, this article also uses other technical operations, including dropout and norm used in deep neural networks. Because the technology of dropout is to avoid learning too much data. In training, it randomly samples the parameters of the weight layer according to a particular probability. It uses the subnetwork as the target network for this update. It is conceivable that if the entire network has the *n* parameter, the number of available subnets is 2^n. In addition, when *n* is large, the subnet used in each iteration update will not be repeated basically to avoid overfitting the training set by a certain network. In this paper, a method for converting a stock index value into a series of images is proposed. These images are used as the input image frames—(collection) stock index vector information 30 days. Then through the designed framework, the stock forecast is realized. The specific framework is shown in Fig.9.

Figure 10 is flowchart of the developed algorithm, which shows that the stock datasets is firstly divided into testing and training datasets. At the same time, the optimized SACLSTM is used to generate a trading strategy

Fig. 8 Example of the input image



Y									
	244.5	248.5	250.5	245.5	248	240	...	228	227.5
	246.5	252	251	246.5	248	241	...	235.5	234
	243	248.5	243	241	242	232.5	...	227	226.5
	246.5	250	245	246	242	234.5	...	234.5	231
	6198	5082	2853	4459	5522	9433	...	5491	4338
X	1	2	3	4	5	6	7	8	9

Fig. 9 Framework for improving the accuracy of stock trading forecasts

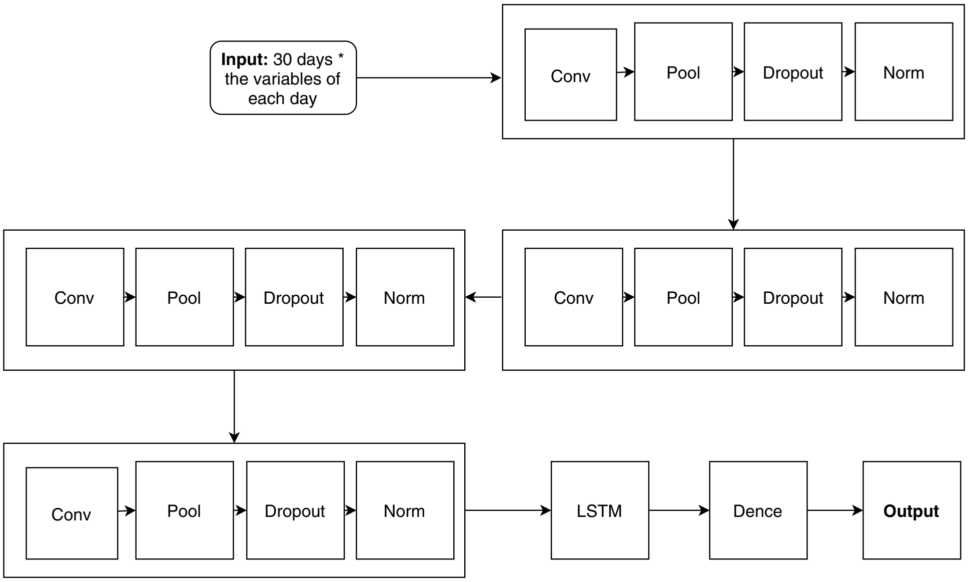
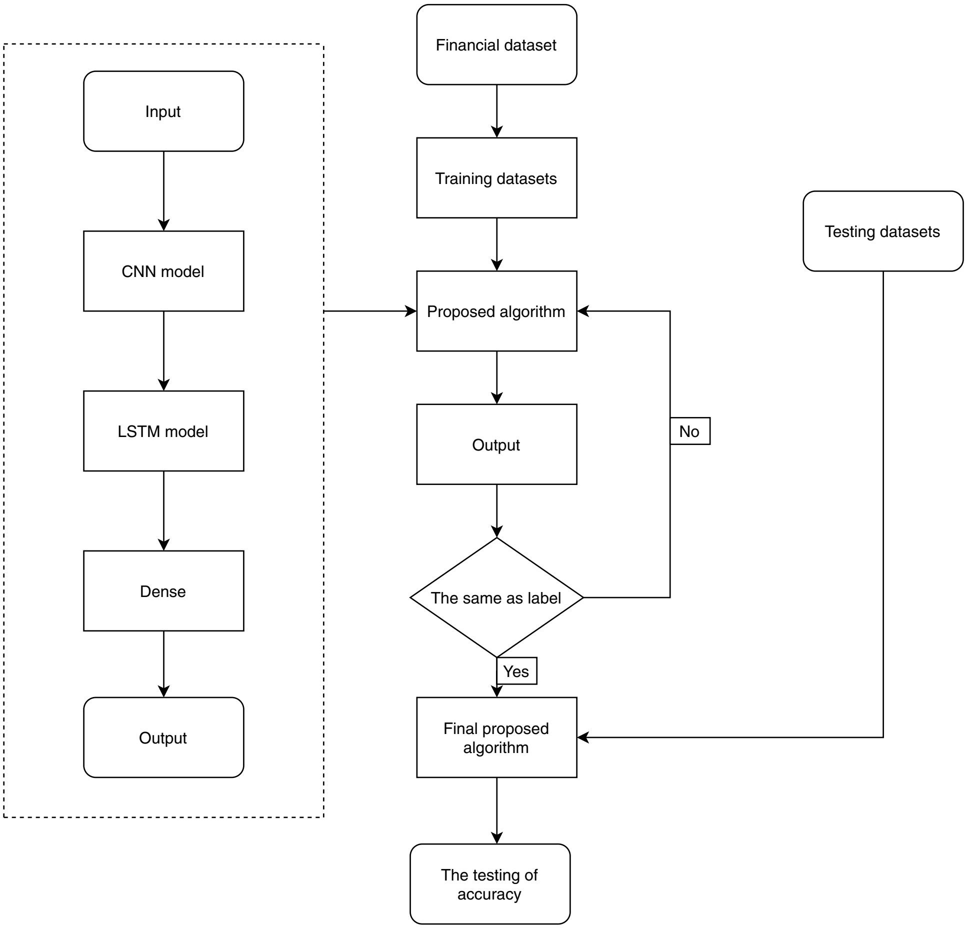


Fig. 10 Flowchart of proposed SACLSTM



based on the formed stock data set. Algorithm 1 is the pseudo-code of the proposed algorithm.

Algorithm 1 The designed SACLSTM algorithm

```
Require: $b$ is the data of training; $c$ is the data of testing; $I$ is the number of iteration; $B$
    is batch size; Algorithm SGD is named $Adam$.
Ensure: the train model $n$; evaluation result $accuracy$
Initialize algorithm
$b \leftarrow Initialize\ algorithm$
$P \leftarrow$ (split $b$ in equal parts of $B$)
for each round $t = 1, 2, ..., z$ do
    $\{verify, train\} \leftarrow \{P_t, P - P_t\}$
    $(tf, vf) \leftarrow$ (generate feature of $train$ and $verify$)
    $n_t \leftarrow$ modelFit$(Adam, tf)$
    $r_t \leftarrow$ modelEvaluate$(n_t, vf)$
end for
$n \leftarrow$ bestModel
$c \leftarrow n$
$accuracy \leftarrow$ modelEvaluate$(n, test)$
```

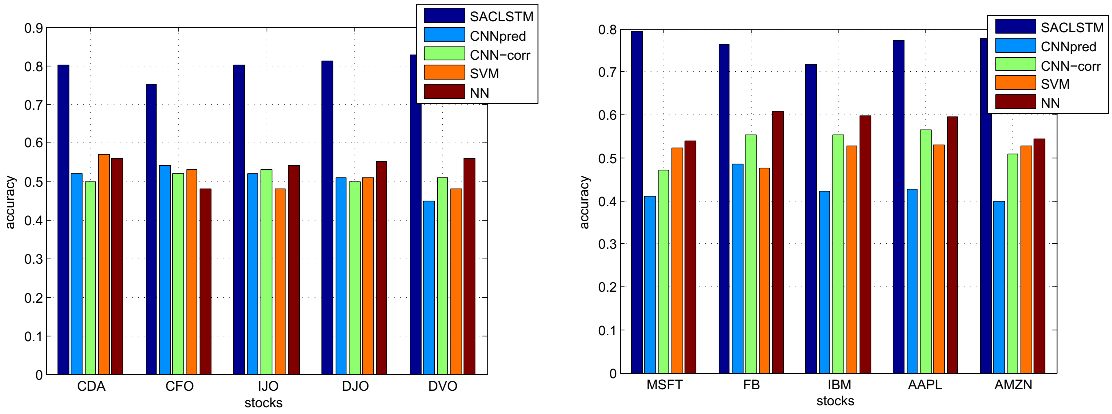
Table 7 Numbers of levels tested in different parameter settings for algorithm

Parameters	Levels
Epochs	200,300...700
Learning rate	0.1, 0.01, 0.001, 0.00001
Activation functions	relu/tanh
LSTM layers	1, 2, 3
Number of hidden layer neurons	64,128...512
The number of hidden layer	3,4,5

6 Experimental results

As mentioned earlier, this paper proposes a different classification prediction framework in classification. Because there are many parameters in the framework, this paper designed different parameter settings. Specific display in Table 7.

To further prove that the performance of the algorithm in the stock prediction market is relatively good, the experiment simulates whether the prediction of SACLSTM based on prediction trading can generate profits. The experimental design mainly includes market classification and attribute selection (options and future). Additionally, this paper provides a comparison of predicted more stock, including 10 financial markets, the three attributes of each stock (options, future, and historical data), as well as six different



(a) Taiwanese stocks

(b) American stocks

Fig. 11 Bar chart of prediction accuracy for all four model specifications, using the data of history

classification algorithms (SVM, CNN-cor, CNNpred, NN, the proposed algorithm).

The proposed framework uses the best prediction framework. The quantity of convolution layers is four and the number of fully connected layers is three. 5 different classification algorithms, CNNpred, CNN-corr, NN, SVM, and the proposed SACLSTM. The first part of the experiment is to set historical prices as input data for all comparison algorithms. The second and third parts use futures, and options as input data instead of historical prices. The last part combines historical prices, futures and options as input data to execute all algorithms. Note that due to the limitation of CNN-corr, only the first part is compared. The reason is that CNN-corr uses the original candlestick chart as the input chart, but futures and options include multiple target prices (options) or different periods (futures); data cannot be transferred to the signal candlestick chart.

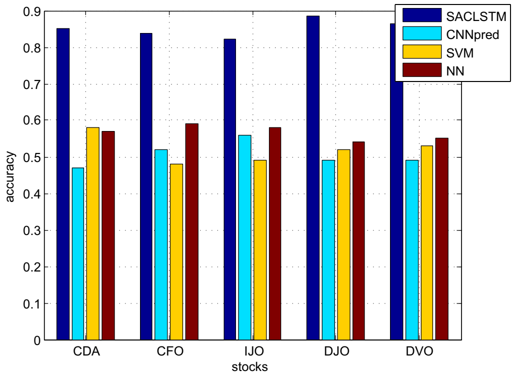
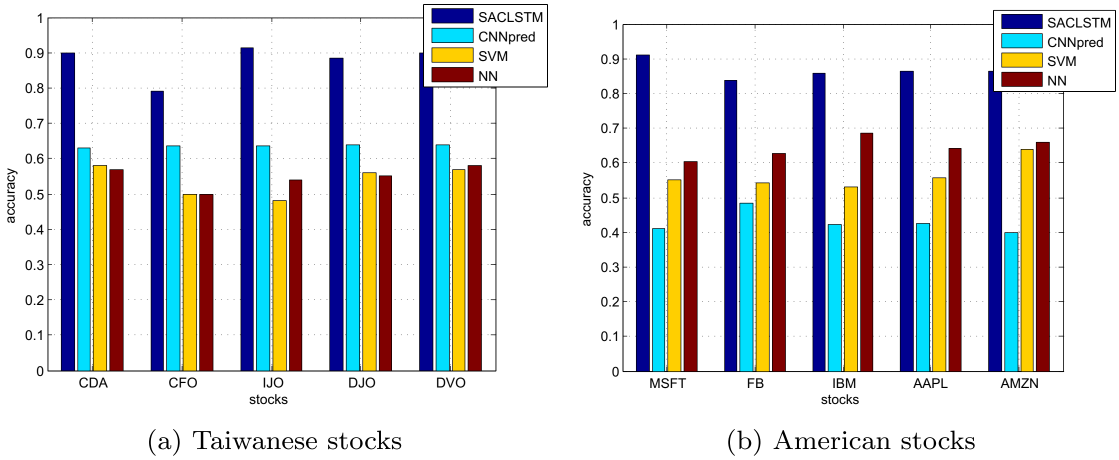


Fig. 12 Bar chart of prediction accuracy for all four model specifications, using the data of future

First, the paper uses the historical date-sets to comparing with the other method (SVM, CNNpred, CNN-corr, NN). Prediction experiments are carried out in the stock of Taiwan and America, and the prediction results between the two markets are compared. Figure 11 shows the prediction results of the set of experiments. It can clearly show that the proposed algorithm is relatively good, and then for the individual historical data, the traditional neural network prediction is relatively better than the rest, because CNN-corr and CNNpred are easy to generate large noise, and the accuracy of SVM is relatively low due to the non-constant sensitivity to the quality of training set.

The framework uses the best prediction model. First of all, only using historical data sets as input data is compared with other methods (SVM, CNNpred, CNN-corr, NN). Figure 11 shows the prediction result of this group of experiments. It can clearly show that the proposed algorithm is relatively good, and traditional neural network prediction is relatively better than others.

Because futures and options are the leading indicators of stocks and they can predict the future development trend of stocks, only options (futures) are used as input data to realize the prediction results and compare with other algorithms. Figures 12 and 13 show the prediction results of this group of experiments. It can be seen from the figure that the proposed framework is the best result (because there is no futures information in the US stock market, the experiments about futures are therefore performed by the Taiwanese stock market). The proposed algorithm is superior to other prediction methods (SVM, CNNpred, NN), and the overall accuracy is increased compared to the single prediction historical data. From the experiments, the results show that using the leading indicators as experimental data is better than only using historical data, and compared with the use



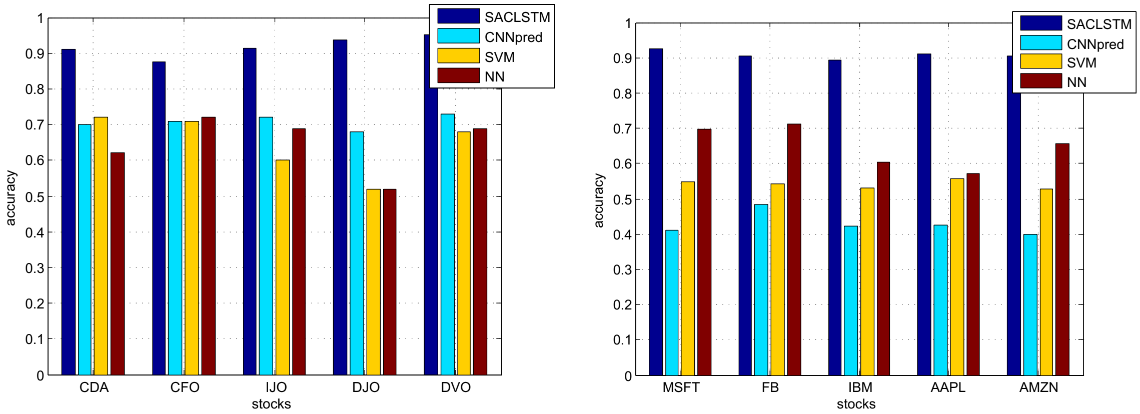
(a) Taiwanese stocks

(b) American stocks

Fig. 13 Bar chart of prediction accuracy for all four model specifications, using the data of option

of futures alone and options, the accuracy of using options is higher than that of using futures.

This paper combines historical data, the data-sets of options, and futures to comparing with the other method (SVM, CNNpred, CNN-corr, NN). Because this paper believes that the stock related indexes are more and the stock day relationship is more close, so the combination of all indicators can make the stock forecast more accurate and prediction experiments are carried out in the stock of Taiwan and America, and the prediction results between the two classes of stocks are compared. Figure 14 shows the prediction results of the set of experiments. It can clearly be shown that the historical data and futures options to achieve better prediction accuracy. What's more, the accuracy of all algorithms is improved obviously. It further showed that the basic analysis is more, the accuracy is higher. Moreover, whether the algorithm proposed in this paper predicts historical data, futures, and options separately, or combines the three, the prediction accuracy is higher than other algorithms.



(a) Taiwanese stocks

(b) American stocks

Fig. 14 Bar chart of prediction accuracy for all four model specifications, using the data of option, future and history

Table 8 Prediction accuracy of historical data in different time windows

Stocks	SSACNN			SSALSTM			SACLSTM		
	1 day	3 days	7 days	1 day	3 days	7 days	1 day	3 days	7 days
CDA	0.73	0.684	0.661	0.756	0.722	0.709	0.802	0.772	0.750
CFO	0.714	0.547	0.546	0.788	0.709	0.662	0.751	0.646	0.612
IJO	0.714	0.624	0.619	0.733	0.677	0.568	0.801	0.783	0.704
DJO	0.675	0.712	0.716	0.758	0.648	0.634	0.812	0.683	0.67
DVO	0.63	0.619	0.604	0.713	0.681	0.627	0.828	0.803	0.761
MSFT	0.742	0.739	0.733	0.755	0.614	0.591	0.795	0.755	0.761
FB	0.693	0.654	0.639	0.716	0.637	0.633	0.763	0.666	0.631
IBM	0.758	0.654	0.583	0.765	0.7	0.694	0.716	0.679	0.644
AAPL	0.644	0.64	0.649	0.688	0.611	0.604	0.774	0.749	0.735
AMZN	0.716	0.696	0.681	0.726	0.613	0.612	0.778	0.752	0.702

Table 9 Prediction accuracy of future in different time windows

Stocks	SSACNN			SSALSTM			SACLSTM		
	1 day	3 days	7 days	1 day	3 days	7 days	1 day	3 days	7 days
CDA	0.812	0.718	0.7	0.845	0.815	0.743	0.85	0.824	0.821
CFO	0.78	0.547	0.546	0.819	0.803	0.786	0.839	0.808	0.792
IJO	0.88	0.514	0.514	0.895	0.815	0.729	0.823	0.829	0.827
DJO	0.705	0.707	0.618	0.86	0.785	0.756	0.887	0.871	0.865
DVO	0.878	0.73	0.707	0.864	0.811	0.769	0.864	0.853	0.85

Table 10 Prediction accuracy of option in different time windows

Stocks	SSACNN			SSALSTM			SACLSTM		
	1 day	3 days	7 days	1 day	3 days	7 days	1 day	3 days	7 days
CDA	0.726	0.72	0.715	0.801	0.804	0.799	0.898	0.867	0.832
CFO	0.63	0.547	0.546	0.849	0.807	0.775	0.791	0.725	0.688
IJO	0.738	0.715	0.514	0.864	0.811	0.774	0.915	0.865	0.839
DJO	0.715	0.705	0.625	0.715	0.773	0.745	0.884	0.843	0.816
DVO	0.731	0.728	0.72	0.865	0.84	0.782	0.898	0.843	0.835
MSFT	0.833	0.787	0.642	0.851	0.812	0.782	0.91	0.892	0.879
FB	0.826	0.738	0.651	0.829	0.817	0.769	0.838	0.797	0.757
IBM	0.727	0.656	0.656	0.831	0.786	0.807	0.857	0.842	0.834
AAPL	0.888	0.886	0.871	0.867	0.851	0.787	0.865	0.828	0.828
AMZN	0.777	0.76	0.748	0.8	0.771	0.749	0.864	0.842	0.818

Table 11 Prediction accuracy of all data in different time windows

Stocks	SSACNN			SSALSTM			SACLSTM		
	1 day	3 days	7 days	1 day	3 days	7 days	1 day	3 days	7 days
CDA	0.78	0.734	0.731	0.904	0.889	0.818	0.911	0.87	0.859
CFO	0.791	0.773	0.677	0.874	0.846	0.814	0.877	0.826	0.795
IJO	0.84	0.74	0.719	0.894	0.826	0.762	0.913	0.866	0.836
DJO	0.8	0.763	0.719	0.874	0.821	0.755	0.936	0.9	0.828
DVO	0.88	0.842	0.736	0.91	0.875	0.79	0.951	0.929	0.89
MSFT	0.785	0.77	0.768	0.855	0.824	0.798	0.927	0.915	0.904
FB	0.886	0.746	0.734	0.886	0.862	0.794	0.905	0.896	0.879
IBM	0.815	0.725	0.583	0.857	0.857	0.786	0.893	0.879	0.877
AAPL	0.861	0.733	0.71	0.86	0.821	0.778	0.91	0.882	0.892
AMZN	0.818	0.764	0.795	0.852	0.833	0.805	0.905	0.888	0.883

This paper believes that the more relevant stocks in the index are more closely related to the day of the stock. To obtain higher prediction accuracy, this paper combines historical data, options, and futures. Figure 14 shows the prediction result of this group of experiments. It can be seen that it is better to combine all indices, and the accuracy of all algorithms has been significantly improved. It can be drawn from the side that the basic analysis is more, the accuracy is higher. It can be seen from Figs. 11, 12, 13 and 14, whether the algorithm proposed in this paper predicts historical data, futures, and options separately, or combines the three, the prediction accuracy is higher than other algorithms.

To prove that the frame is formed by combining CNN and LSTM is better, it is used to compare with the frame is formed by using CNN [56] and LSTM alone. In addition, this paper uses three different time windows (1 day, 3 days, and 7 days) to conduct prediction experiments, and analyzes how the prediction results change with the change of the prediction time point. The results are described in Tables 8, 9, 10 and 11. According to the chart, the prediction accuracy of the three frameworks for the next day is relatively high. The prediction results by combining CNN and LSTM are better than only using CNN and LSTM. Thus, it shows that the better performance by combining CNN and LSTM has achieved.

Table 12 Loss of accuracy with historical data in Taiwan and America

Stocks	1 day	3 days	7 days
CDA	0.5043978	0.6718477	0.6527476
CFO	0.70406383	0.76469076	0.8652994
IJO	0.52903223	0.57157063	0.75546515
DJO	0.57422733	0.7819671	0.7880473
DVO	0.48612314	0.54723656	0.6087184
MSFT	0.59399843	0.5860628	0.65412676
FB	0.67507094	0.76914316	0.83222073
IBM	0.7023954	0.7292076	0.7830649
AAPL	0.62354255	0.6307928	0.656635
AMZN	0.5792342	0.6566861	0.7045637

Aiming at forecasting future fluctuations in a future day, it is found that the error value has the smallest fluctuation in the next day of prediction. The change in accuracy is inversely proportional to the error value. The error is bigger, the accuracy is lower. The error is shown in Tables 12, 13, 14 and 15.

Table 13 Loss of accuracy with the data of future in Taiwan

Stocks	1 day	3 days	7 days
CDA	0.48684865	0.5022441	0.496224
CFO	0.4479517	0.571236	0.6024883
IJO	0.5209426	0.5138144	0.52633363
DJO	0.3827717	0.4371534	0.4428909
DVO	0.4128047	0.41952732	0.4514946

Table 14 Loss of accuracy with the data of option in Taiwan and America

Stocks	1 day	3 days	7 days
CDA	0.32209763	0.43711987	0.4898555
CFO	0.62145257	0.68822795	0.7298958
IJO	0.36444992	0.43182352	0.46717995
DJO	0.42761612	0.49330416	0.53118193
DVO	0.31992638	0.4887478	0.5145777
MSFT	0.32706252	0.3975222	0.37687218
FB	0.54278606	0.58843964	0.61832947
IBM	0.47456282	0.4691932	0.4288829
AAPL	0.4968128	0.5148604	0.5273625
AMZN	0.44606668	0.4463848	0.51139814

Table 15 Loss of accuracy with historical data and the data of future and option in Taiwan and America

Stocks	1 day	3 days	7 days
CDA	0.34772763	0.4496867	0.43273693
CFO	0.47327614	0.53362805	0.5926226
IJO	0.33787438	0.4237443	0.5268749
DJO	0.1742997	0.43805385	0.50643253
DVO	0.24088037	0.30687672	0.38472039
MSFT	0.28458542	0.31529334	0.33080247
FB	0.31984925	0.36343572	0.40864927
IBM	0.36103746	0.37637982	0.3859181
AAPL	0.32419607	0.37362128	0.37869915
AMZN	0.39042678	0.38110036	0.32946447

7 Conclusion

The noise and nonlinear behavior of prices in financial markets has proven that forecasting the trends of financial markets is not trivial, and it is better to consider the proper variables for stock prediction. Thus, the designed SACLSTM uses a variety of news collections, including options, historical data, and futures and involves the stock sequence array convolution LSTM algorithm for stock prediction. In the designed SACLSTM, the convolutional layer is used to extract financial features, and the classification task is to classify and predict the stocks through a long and short-term memory network. It is verified that the neural network framework combined with convolution and long–short-term memory units achieved better performance for statistical methods and traditional CNN and LSTM in prediction tasks. To avoid the data being too scattered and reduce useless information, firstly, integrating the data directly into a matrix, and using convolution to extract high-quality features is designed in the SACLSTM. In addition, the designed SACLSTM refers to some leading indicators that is used to improve the prediction performance of stock trends. Overall, the framework effectively improves the effectiveness of stock price prediction.

Since the main purpose of this paper is to predict the rise and fall of the stock market and clearly prove that it can be successfully used in a trading system and obtain results, therefore, the next step will utilize the proposed algorithm to indicate the rise or fall of a specific point, and further establish an expert system for investment.

Funding Open access funding provided by Western Norway University Of Applied Sciences.

References

1. Rani, S., Sikka, G.: Recent techniques of clustering of time series data: a survey. Int. J. Comput. Appl. **52**(15), 1–9 (2012)
2. Zhong, X., Enke, D.: Forecasting daily stock market return using dimensionality reduction. Expert Syst. Appl. **67**, 126–139 (2017)
3. Gunduz, H., Yaslan, Y., Cataltepe, Z.: Intraday prediction of borsa istanbul using convolutional neural networks and feature correlations. Knowl. Based Syst. **137**, 138–148 (2017)
4. Hagenau, M., Liebmann, M., Neumann, D.: Automated news reading: stock price prediction based on financial news using context-capturing features. Decis. Support Syst. **55**(3), 685–697 (2013)
5. Kj, Kim, Han, I.: Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Syst. Appl. **19**(2), 125–132 (2000)
6. Kuo, S.Y., Kuo, C., Chou, Y.H.: Dynamic stock trading system based on quantum-inspired tabu search algorithm. In: IEEE congress on evolutionary computation, pp. 1029–1036 (2013)
7. Long, W., Lu, Z., Cui, L.: Deep learning-based feature engineering for stock price movement prediction. Knowl. Based Syst. **164**, 163–173 (2019)

8. Khaidem, L., Saha, S., Dey, S.R.: Predicting the direction of stock market prices using random forest. arXiv preprint. arXiv:160500003 (2016)
9. Hu, P., Pan, J.S., Chu, S.C., Chai, Q.W., Liu, T., Li, Z.C.: New hybrid algorithms for prediction of daily load of power network. Appl. Sci. **9**(21), 4514 (2019)
10. Pan, J.S., Hu, P., Chu, S.C.: Novel parallel heterogeneous metaheuristic and its communication strategies for the prediction of wind power. Processes **7**(11), 845 (2019)
11. Chen, C.H., Hsieh, C.Y.: Actionable stock portfolio mining by using genetic algorithms. J. Inf. Sci. Eng. **32**(6), 1657–1678 (2016)
12. Chen, C.H., Lu, C.Y., Lin, C.B.: An intelligence approach for group stock portfolio optimization with a trading mechanism. Knowl. Inf. Syst. **62**(1), 287–316 (2020)
13. Chen, C.H., Yu, C.H.: A series-based group stock portfolio optimization approach using the grouping genetic algorithm with symbolic aggregate approximations. Knowl. Based Syst. **125**, 146–163 (2017)
14. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: IEEE International conference on acoustics, speech and signal processing, pp. 6645–6649 (2013)
15. Zhao, Z., Zhang, X., Zhou, H., Li, C., Gong, M., Wang, Y.: Hetnerec: Heterogeneous network embedding based recommendation. Knowl. Based Syst. **204**, 106218 (2020)
16. Chen, Q.a., Li, C.D.: Comparison of forecasting performance of ar, star and ann models on the chinese stock market index. In: International symposium on neural networks, pp. 464–470 (2006)
17. Gardner, M.W., Dorling, S.: Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences. Atmos. Environ. **32**(14–15), 2627–2636 (1998)
18. Baba, N., Kozaki, M.: An intelligent forecasting system of stock price using neural networks. Int. Jt. Conf. Neural Netw. **1**, 371–377 (1992)
19. de Oliveira, F.A., Nobre, C.N., Zarate, L.E.: Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index-case study of petr4, petrobras, brazil. Expert Syst. Appl. **40**(18), 7596–7606 (2013)
20. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: Twenty-fourth international joint conference on artificial intelligence (2015)
21. Zhao, Z., Zhou, H., Li, C., Tang, J., Zeng, Q.: Deepemlan: deep embedding learning for attributed networks. Inf. Sci. **543**, 382–397 (2021)
22. Yong, B.X., Rahim, M.R.A., Abdullah, A.S.: A stock market trading system using deep neural network. In: Asian simulation conference, Springer, pp. 356–364 (2017)
23. Wu, J.M.T., Tsai, M.H., Xiao, S.H., Liaw, Y.P.: A deep neural network electrocardiogram analysis framework for left ventricular hypertrophy prediction. J. Ambient Intell. Humaniz. Comput. pp. 1–17 (2020). https://doi.org/10.1007/s12652-020-01826-1
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
25. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Comput. **1**(2), 270–280 (1989)
26. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
27. Chen, K., Zhou, Y., Dai, F.: A lstm-based method for stock returns prediction: A case study of china stock market. In: IEEE International conference on big data, pp. 2823–2824 (2015)
28. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. Eur. J. Oper. Res. **270**(2), 654–669 (2018)
29. Cai, X., Hu, S., Lin, X.: Feature extraction using restricted boltzmann machine for stock price prediction. IEEE Int. Conf. Comput. Sci. Autom. Eng. **3**, 80–83 (2012)
30. Zhu, C., Yin, J., Li, Q.: A stock decision support system based on dbns. J. Comput. Inf. Syst. **10**(2), 883–893 (2014)
31. Bao, W., Yue, J., Rao, Y.: A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS One **12**(7), e0180944 (2017)
32. Di Persio, L., Honchar, O.: Artificial neural networks architectures for stock price prediction: comparisons and applications. Int. J. Circ. Syst. Signal Process. **10**(2016), 403–413 (2016)
33. Hoseinzade, E., Haratizadeh, S.: Cnnpred: Cnn-based stock market prediction using a diverse set of variables. Expert Syst. Appl. **129**, 273–285 (2019)
34. Siripurapu, A.: Convolutional networks for stock trading. Stanford Univ Dep Comput Sci, pp. 1–6 (2014)
35. Wu, J.M.T., Li, Z., Lin, J.C.W., Pirouz, M.: A new convolution neural network model for stock price prediction. In: International conference on genetic and evolutionary computing, pp. 581–585 (2019)
36. Ghosh, A., Bose, S., Maji, G., Debnath, N., Sen, S.: Stock price prediction using lstm on indian share market. Int. Conf. Comput. Appl. Ind. Eng. **63**, 101–110 (2019)
37. Zhang, X., Tan, Y.: Deep stock ranker: a lstm neural network model for stock selection. In: International conference on data mining and big data, pp. 614–623 (2018)
38. Azzouni, A., Pujolle, G.: A long short-term memory recurrent neural network framework for network traffic matrix prediction. arXiv preprint. arXiv:170505690 (2017)
39. Tsai, H.H., Wu, M.E., Wu, W.H.: The information content of implied volatility skew: evidence on Taiwan stock index options. Data Sci. Pattern Recogn. **1**(1), 48–53 (2017)
40. Krollner, B., Vanstone, B.J., Finnie, G.R.: Financial time series forecasting with machine learning techniques: a survey. In: ESANN, pp. 25–30 (2010)
41. Bahmani-Oskooee, M., Sohrabian, A.: Stock prices and the effective exchange rate of the dollar. Appl. Econ. **24**(4), 459–464 (1992)
42. Zhang, X., Chen, Y., Yang, J.Y.: Stock index forecasting using pso based selective neural network ensemble. In: IC-AI, pp. 260–264 (2007)
43. Pang, X., Zhou, Y., Wang, P., Lin, W., Chang, V.: An innovative neural network approach for stock market prediction. J. Supercomput. **76**(3), 2098–2118 (2020)
44. Nelson, D.M., Pereira, A.C., de Oliveira, R.A.: Stock market's price movement prediction with lstm neural networks. In: The international joint conference on neural networks, pp. 1419–1426 (2017)
45. Taylor, M.P., Allen, H.: The use of technical analysis in the foreign exchange market. J. Int. Money Finance **11**(3), 304–314 (1992)
46. Jiang, H., Learned-Miller, E.: Face detection with the faster r-cnn. In: IEEE international conference on automatic face and gesture recognition, pp. 650–657 (2017)
47. Garcia Garcia, A., Orts Escolano, S., Oprea, S., Villena Martinez, V., Garcia Rodriguez, J.: A review on deep learning techniques applied to semantic segmentation. arXiv preprint. arXiv:170406857 (2017)
48. Wu, J.M.T., Wu, M.E., Hung, P.J., Hassan, M.M., Fortino, G.: Convert index trading to option strategies via lstm architecture. Neural Comput. Appl. pp. 1–18 (2020)
49. Lin, J.C.W., Shao, Y., Djenouri, Y., Yun, U.: Asrnn: A recurrent neural network with an attention model for sequence labeling. Knowl. Based Syst. Vol. 212, p. 106548 (2020)
50. Cui, Y., Wang, S., Li, J.: Lstm neural reordering feature for statistical machine translation. arXiv preprint. arXiv:151200177 (2015)

51. Han, S., Kang, J., Mao, H., Hu, Y., Li, X., Li, Y., Xie, D., Luo, H., Yao, S., Wang, Y., et al.: Ese: Efficient speech recognition engine with sparse lstm on fpga. In: ACM/SIGDA international symposium on field-programmable gate arrays, pp. 75–84 (2017)
52. Kinghorn, P., Zhang, L., Shao, L.: A hierarchical and regional deep learning architecture for image description generation. Pattern Recogn. Lett. **119**, 77–85 (2019)
53. Gui, D., Zhong, Sh., Ming, Z.: Implicit affective video tagging using pupillary response. In: International conference on multimedia modeling, pp. 165–176 (2018)
54. Cao, J., Li, Z., Li, J.: Financial time series forecasting model based on ceemdan and lstm. Phys. A Stat. Mech. Appl. **519**, 127–139 (2019)
55. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint. arXiv:14126980 (2014)
56. Wu, J.M.T., Li, Z., Srivastava, G., Tasi, M.H., Lin, J.C.W.: A graph-based convolutional neural network stock price prediction with leading indicators. Pract. Exp. Softw. (2020). https://doi.org/10.1002/spe.2915

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