Received: 27 April 2020 | Revised: 25 July 2020 | Accepted: 29 September 2020

DOI: 10.1002/spe.2915

SPECIAL ISSUE PAPER

WILEY

A graph-based convolutional neural network stock price prediction with leading indicators

Jimmy Ming-Tai Wu[1] | **Zhongcui Li**[1] | **Gautam Srivastava**[2,3] | **Meng-Hsiun Tasi**[4] | **Jerry Chun-Wei Lin**[5]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, Shandong, China

[2] Department of Math and Computer Science, Brandon University, Brandon, Canada

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

[4] Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan

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

Correspondence
Jerry Chun-Wei Lin, Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Inndalsveien 28, Bergen 5063, Norway.
Email: jerrylin@ieee.org

Abstract

The stock market is a capitalistic haven where the issued shares are transferred, traded, and circulated. It bases stock prices on the issue market, however, the structure and trading activities of the stock market are much more complicated than the issue market itself. Therefore, making an accurate prediction becomes an intricate as well as highly difficult task. On the other hand, because of the potential benefits of stock prediction, it attracts generation after generation of scholars as well as investors to continuously develop various prediction methods from different perspectives, a myriad of theories, a multitude of investment strategies, and different practical experiences. In this article, aiming at the task of time series (financial) feature extraction and prediction of price movements, a new convolutional novel neural network that can be called a framework to improve the prediction accuracy of stock trading is proposed. The method that is proposed is called SSACNN, a short form of stock sequence array convolutional neural network. SSACNN collects data including historical data of prices and its leading indicators (options/futures) for a stock to take an array as the input graph of the convolutional neural network framework. In our experimental results, five Taiwanese and American stocks were used as a benchmark to compare with the previous algorithms and proposed algorithm, the motion prediction performance of SSACNN has been improved significantly and proved that it has the potential to be applied in the real financial market.

KEYWORDS

convolutional neural network, options and futures of stocks, prediction, stock history

1 | INTRODUCTION

The financial market is a market coexisting with the commodity market as well as the labor market. It is a known fact that the allocation of financial resources and the development of financial markets are important indicators of a country's economic, financial, and institutional development.[1–3] Currently, many different financial time series forecasting methodologies are known to exist within any current financial market globally. In particular, stock price forecasting is a goal pursued by investors and researchers that has proved to be a very difficult egg to crack. It plays an important role

Abbreviations: CNN, convolutional neural network; LSTM, long short-term memory networks; SSACNN, stock sequence array convolutional neural network in trading strategies to sell and buy stocks as well as to identify different known financial opportunities tied to the stock market. The production of trading strategies is not only the result of historical market behavior but also influenced by macroeconomic information, investor sentiment, and other information. Various methods and data sources are used for stock market forecasting.[4–8] The most common method is to establish the relationship model between historical behavior and future price trend and use historical market samples to predict future price trends or values.[6] Over time, traditional forecasting methods include statistical methods, linear discriminant analysis, quadratic discriminant analysis, random forests,[9] logistic regression and evolutionary computation algorithms.[10,11] Recently, a genetic algorithm is a tool and technology that can extract features from the original financial data according to a set of variables that has shown some success.[12–14] The key part of the forecasting process is feature extraction, which assumes that the future price trend is the result of historical behavior. Notably, people have designed the features subjectively, and the model based on the technical analysis features are based on some assumptions of the market model. The success of the model is mainly determined by the correctness of these assumptions.

In recent years, neural networks (NNs)[15–21] play an increasingly important role in social life and are often used in image recognition, speech recognition, NLP, and text recognition. The basic principle of NNs is to use a series of available NN feature extractors to design a feature-network for specific samples, so as to achieve the desired effect and complete its own project. For example, Oliverira et al.[22] proposed an effective method, which builds a neural framework for the financial market. It allows predictions of stocks closing price future behavior in the short term, combining technical analysis, using financial theory and the economy, analyzing time series, and fundamental analysis, to predict price behavior. With the rapid development of the deep learning field, its application range is more and more extensive. Therefore, based on the application of traditional NNs to the financial field, deep learning is reflected in it.[23] The main difference between deep learning and traditional NNs is that deep learning provides a group of units, such as, convolution, recursion, long short-term memory networks (LSTM),[24,25] and so forth, and algorithms like deep multilayer perceptron (MLP),[26] autoencoder,[27] restricted Boltzmann machine,[28,29] and so forth. It is based on specific data to form specific data consistency, and target consistency, especially since convolutional neural network (CNN) is up-and-coming in deep learning research at present. As far as we know, CNNs have been applied in some studies of stock market prediction.[4,30–32] For example, a previous CNN work uses stock candlestick charts to be the input image and feeds into the input layer directly,[32] or is to seek out a general framework for mapping the historical data of a market to its future fluctuations,[31] or a CNN is used which took a one-dimensional input for realizing prediction based only on the history of closing prices, ignoring other possible variables, such as technical indicators made use of a CNN which was able to use technical indicators for each sample.[4,30] However, it was capable of considering the correlation which may exist between stock markets as another possible source of information. In Reference 30, LSTM, MLP, and CNN were used in the historical data of the closing price of the S&P 500 index. The results showed that CNN was better than LSTM and MLP.

In the aspect of financial time series and sample feature extraction at present, the opening price, closing price, highest price, lowest price, and trading volume of historical data are usually used for data prediction in feature extraction. This article intends to use stock related leading indicators based on historical data. The so-called stock leading indicators are the statistical data of economic indicators that affect future economic development. Market analysts often use these indicators to analyze future economic growth and its impact on the development direction of the exchange rate. The leading indicators here are mainly futures and options. Thus in this work, we propose a novel CNN framework that can be used to simulate principle image output as well as integrate data already in existence into image form. The method as proposed takes data as pieces and uses these image pieces to train network weights. From this, useful features are extracted which assist in the identification of extreme values in the market through validation of signals. Historic data from both the Taiwanese and American stock markets are used as input data which are also used an input vector for the proposed CNN. Through experimental evaluation, different data as well as different indicators are made use of to test the appropriate signals. Through our results, we present a simplistic forecasting trading strategy to help validate the stability and profitability. Our main progress can be summarized as follows:

1. In financial markets, since financial market behavior is affected by many factors, it is essential to collect as much information as possible. Based on the framework of CNN, the input of the initial variable covers all the relevant information of the stock. And it can be easily extended to cover other aspects of the stock for data extraction and market prediction.
2. This article proposes a two-dimensional tensor as the input data in the proposed CNN network. It then uses a feature extraction method to train the system then predict the stock market.
3. The algorithm proposed in this article is compared with several previous algorithms, which proves that this algorithm can better avoid too much noise, save more useful information, and prevent overfitting phenomenon.

The rest of this article is organized as follows. The second part mainly introduces the current work in stock forecasting and then summarizes the methods and applications of machine learning and deep learning in this aspect. The third part presents the proposed preprocessing method and CNN framework. The fourth part introduces the training methods, prediction skills, and experimental results of classification and market simulation. Finally, the fifth section summarizes the experimental results and puts forward some suggestions and shortcomings for further improvement.

2 | RELATED WORK

Before introducing the algorithm proposed in this article, first of all, we look at the research on the financial market, second, it analyzes the common methods of stock forecasting in recent years and their shortcomings, last of all it introduces the algorithm proposed in this article. We can define the financial time series as the organization of values of random financial variables in a certain period. The most striking feature is that it is closely linked to "time." There are strong dependence and periodicity before and after the data. It is impossible to adjust its order. Generally speaking, financial time series variables, also known as financial time series variables, are composed of two elements, namely, time and sequence frequency. Ran and Sikka stated that time series clustering is one of the essential concepts of data mining.[1] They are used to understand the mechanism of generating time series and predict the future value of a given time series. However, the time and frequency in the stock are often different, so it is not easy to predict the stock. It is usually necessary to regulate uneven time-series data. For example, selecting an appropriate frequency, and taking the last large transaction price in the frequency period as the observation value of the period, unlike pictures, words, voice samples, each data implies very much. There are many problems, and the correlation between the data is not very high, and it does not have strong dependence and periodicity. Therefore, to carry out stock prediction, much useful information is needed in the feature. In other words, feature extraction and selection play an important role in stock forecasting. Nowadays, there are many traditional methods applied to stock forecasting. Through considerations for stock prices, Chen et al.[14] presented an improved methodology to give a more feasible stock portfolio application for investors. Furthermore, a group stock portfolio that is sequence-based was derived[12] to give sound investment advice. They also proposed an optimization algorithm that uses many aspects of evolutionary computation.

On the other hand, feature extraction and selection play a crucial role in stock price forecasting. Traditional statistical and machine learning methods have been applied in financial sequence modeling and prediction. Some specific methodologies build on feature extraction and election in these models. Due to the high uncertainty and volatility of stocks,[5] the conventional methods of feature extraction are technical analysis and statistical methodologies. Furthermore, for stock forecasting, the use of direct methodologies has shown to be the most effective using technical analysis. Through the detailed use of technical analysis, users assume that historic data should be correlated to future trends.[33] Hence, many technical indicators can be defined that assist in patterns being applied in IES, short form for investment expert systems. And most of these indicators are used to describe the specific characteristics of the assumed pattern, which is a mathematical expression of historical price series. It can be seen that technical analysis is a postmortem analysis, which uses historical data to predict the future and data, graphics, and statistical methods to explain problems.[33] At the same time, features extracted by designed indicators are based on presumed patterns so that some information may be lost in this approach. Moreover, the application of statistical methods in technical analysis is more common. It is mainly about the computer-based data to build a mathematical probability model and use the model to predict and analyze the data. Learn probabilistic statistical models from data, and then use them to analyze and predict new data. It focuses on the compression of information and the reduction of dimensions, and a machine learning method is used to extract features. The most common way is the principal component analysis.[2] The dataset is performed in the feature extracted, and they mainly focused on the distribution of samples. At present, statistical methods and machine learning methods implement features extracted, which are commonly used in financial time series prediction. Statistical methods' and machine learning methods' projections are based on verified assumptions, and statistical models are used to validate assumptions of the market behind features. Machine learning methods provide considerable ability to learn the potential relationships between patterns in labels and features. It does not need to establish explicit relationships and mathematical models of complex nonlinear systems and can overcome many limitations and difficulties of traditional quantitative prediction methods. Therefore, they further avoid the influence of many human factors.

In recent years, with the rapid development of the NN, the NN plays an increasingly important role in social life. The NN is often used for image recognition, speech recognition, text recognition, and so forth. One of the most basic principles is to use a series of existing NN feature extractors to design a featured network in a specific sample to achieve the desired effect and complete their project. With the development of deep learning, it is gradually applied to financial time series.[8] In deep learning, deep learning methods have achieved remarkable results in speech recognition[34] and some path recognition. The important difference between deep learning and traditional methodologies is that specific data formation and the objective task can be tailored to the network structure. In deep learning, it provides a lot of units, for example, recurrent unit,[35] convolutional unit,[36] long-short memory term unit[37] with different characteristics for feature extraction on samples. What is more, the network can integrate into the feature extraction process, and according to characteristics of samples, it can be purposely to design the structures of networks. According to the pixel data, many of the architectures were designed to enhance the abilities to learn information.

LSTM makes use of a memory function that is unique and a sequential time concept. Hence, the financial time series is an adequate use of LSTM. Several algorithms[34,38] were presented to introduce an LSTM-based method that is used to predict stock market action. Technical indicators are fed directly into LSTM to predict the general direction of the South American Brazilian stock market.[39] In general, through intensive literature review, LSTM tends to be more successful than MLP. With the continuous deepening of the network, this article finds that the learning ability of the network can be improved. With the development of network technology, some deep learning methods have been applied to stock price forecasting. Because the learning ability of convolutional networks is particularly strong in image recognition and its ability to extract effective features has also been proven in many fields, a CNN is a deep learning algorithm applied to the stock market after LSTM.

In the past, a previous CNN work, which is applied to the traditional CNN framework, used stock candlestick charts to be the input images to predict a stock tend. Siripurapu proposed CNN-corr to uses stock candlestick charts to be the input image and feeds into the input layer directly,[32] or CNNpred algorithm proposed by Hoseinzade and Haratizadeh seeks out a general framework for mapping the historical data of a market to its future fluctuations.[31] Nevertheless, the input images contain too much noise, useless information, and are prone to overfitting. Recurrent neural networks is also applied in stock price prediction. In Reference 39, it uses the LSTM NN to deal with time series and applies it to stock forecasting. Still, it just uses the historical stock prices to be as input data; it thus cannot effectively extract enough features to enhance the performance. As well as the initial use of the traditional NN for stock forecasting, the conventional NN will produce a lot of noise, as well as an overfitting phenomenon. Therefore, most of the researchers always have used only historical price data or technical indicators for prediction.

A machine learning method of support vector machine (SVM) is introduced to establish a stock selection model, which can classify stocks nonlinearly. However, the accuracy of the SVM classification is very sensitive to the quality of the training set. Therefore, an SVM-based model[2] is proposed that avoided to use complex financial data to ignore this problem. There is too much valuable information in the stock market to be ignored but the previous researches did not refer to it. Therefore, in our proposed method, the developed stock sequence array convolutional neural network (SSACNN) is proposed to help resolve previous issues and improve the performance of stock market prediction. We focus SSACNN on the meaningful information and then use the generated information to create a data array which is used in a CNN. More than anything, its input data is not only related to the indexes of historical data, but also to stock related the indexes of leading indicators (options and futures). Our experimental work shows that the developed SSACNN is a clear improvement over its predecessors. Next, we introduce algorithmic details in the following section.

3 | STOCK SEQUENCE ARRAY CNN

In this section, the process and the detailed definitions of the proposed SSACNN is proposed. First of all, there are many leading indexes and historical price data for a stock used to produce the input image in SSACNN. Second, a normalization function is going to be introduced to modify the input data, and it can cause the proposed method to focus on the trend of the prices, not on the actual value. In the last part of this section, the article will be showed to describe the process for the proposed SSACNN.

3.1 | Datasets

Before making stock forecasts, an index sequence $y_1, y_2, \ldots\ y_t$ is generated as an input data. It includes the historical data of prices and two leading indexes, futures and options of stock. In the experiments, some stocks from the American stock market and Taiwanese stock market will be applied. The so-called leading indicator of the stock is the statistics of

TABLE 1 The historical data of the five stocks

s_i	d_{i1}	d_{i2}	d_{i3}	d_{i4}	$d_{i.}$
s_1	246.5	244.5	246.5	243	…
s_2	117	117.5	117.5	116	…
s_3	262.5	266	266	260	…
s_4	264	260	264	259	…
s_5	3635	3825	3880	3635	…

TABLE 2 The futures of the five stocks

s_i	t_{i1}	t_{i2}	t_{i3}	t_{i4}	$t_{i.}$
s_1	246	244	246.5	243.5	…
s_2	117	117	117.5	117	…
s_3	262.5	265	265.5	262.5	…
s_4	263.5	262	263.5	258	…
s_5	3660	3815	3865	3635	…

TABLE 3 The option data of the five stocks

s_i	z_{i1}	z_{i2}	z_{i3}	z_{i4}	$z_{i.}$
s_1	3.4	2	6.85	2	…
s_2	3.25	20	0.51	15	…
s_3	5.2	70	7.3	11	…
s_4	14.85	1	0.27	1	…
s_5	297	0	30.1	0	…

the economic indicators that affect future economic development. Market analysts often refer to these indicators to analyze future economic development and its impact on the future direction of exchange rate development. Here, the article mainly uses the option and futures in the leading indicators. These two indexes and stock prices make up the characteristics of each sample. First of all, this article shows the relevant information on five stocks in Taiwan. Its attributes include the historical data of stock and the attribute of the stock's futures and options. Please see Tables 1-3 for details.

In Tables 1 and 2, where s_i is denoted the five Taiwanese stocks, they are *DVO*, *CFO*, *CDA*, *DJO*, *IJO*. $d_{i.}$ are denoted the present price, opening price, closing price, the highest price, and other attributes of the stocks, and $t_{i.}$ are denoted the attributes of the futures (present price, opening price, closing price, the highest price, among others). In Table 3, since there are two kinds of options: the right to buy and sell, where $z_{i.}$ are denoted the settlement price and open interest of the right to buy and sell, among others. Note that the proposed SSACNN selects 20 options (10 call options and 10 put options) where the contract prices are closest to the current stock price to generate the option data array.

3.2 | Normalization function

To train a deep learning network for general situations, SSACNN normalizes all of the input values. Its main purpose is to limit the preprocessed data to a certain range, to eliminate the adverse effects caused by the singular sample data. If there is no normalization, the difference between the values of the different features in the feature vector will cause the objective function to become "flat." The price range of training data may be quite different from that of test data. Therefore, to obtain the concept of price's trend can need to normalize the input values. It can avoid the network is just available for a certain range of prices. The proposed function is given in Equation (1) as:

$$\dot{X}_t = \frac{X_t - \text{mean}}{\max - \min}, \tag{1}$$

TABLE 4 The normalization of historical data of the five stocks

s_i	d_{i1}	d_{i2}	d_{i3}	d_{i4}	$d_{i.}$
s_1	0.390278	0.386517	0.656061	0.392177	…
s_2	0.622453	0.13237	0.12548	0.13422	…
s_3	0.622453	0.134237	0.12548	0.13422	…
s_4	0.639662	0.486275	0.54878	0.560417	…
s_5	0.33647	0.46389	0.53128	0.36392	…

where X_t is the indexes vector for time t (*open*, *high*, *low*, *close*, …), $\dot{X}_t$ is the index vector after the normalize process. The *mean*, *max*, and *min* are the average value, maximal value and minimal value of the index vector in a certain period. In the experiments, the length of the period is set to a period. The data will be collected in 120 days to establish an input array. Take the value 246.5 of s_1 as an example in Table 2, the mean of the same property was taken for the first 120 days, the highest and the lowest price and use Equation (1), the normalized value is 0.390278. After normalization in the same way, all the normalized data are shown in Table 4.

3.3 | Stock prediction model based on CNN

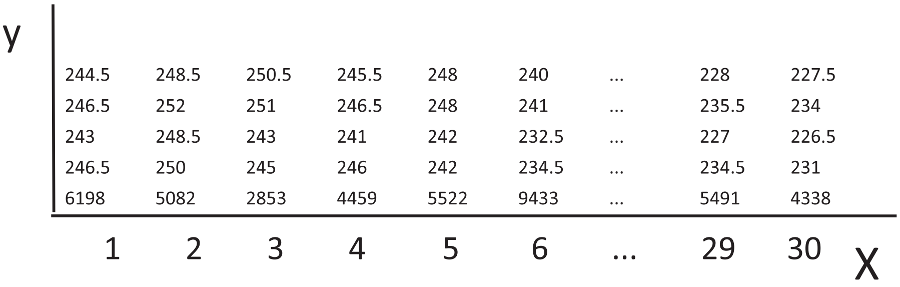
3.3.1 | Stock indexes and the input image

Futures are the same as stocks, and the sale is not a real product, but it is a contract for futures transactions. Conducting two-way trading can make money even in the situation as the market is not good. For example, you know that the apple tree in the neighbor's apple tree is good, but it takes 3 months to get ripe. Right away, the market price is 3 dollars per catty. You want to order 3 dollars per catty to the neighbors, but the neighbor thinks that the output is low this year and want to increase to 3.5 dollars. The last, you make a deal for 3.2 dollars to order 1000 pounds, and you pay 10% of the deposit (320 dollars). So you and your neighbor sign a futures contract finally. A month later, a massive natural disaster occurred in the apple producing areas. The price of an apple is rose to 6 dollars per catty. At this time, your friend who engages fruit business knows that you have an apple order contract and ask you to transfer it to him. You agree to give him a wholesale of 5 dollars per catty. Therefore, you transfer your futures contract and look at your earnings, which is calculated as $(5 - 3.2) * 1000 = 1800$ dollars. The rate of return is $\frac{5}{3.2}$ times.

On the other hand, options are similar to futures. It is also a kind of contract. Options are the right to buy (or sell) a certain amount of the underlying assets (or commodities) to the other party at a fixed price before the maturity date of the contract. Options can be divided into the following two kinds: the right to buy and the right to sell. The most significant advantage of the option is that the party who buys options after paying the right obtains the right to perform it or not, but does not have to bear the obligation to perform it. In options, the call and sell rights include several attributes, respectively, the *settlement price* (after the end of the transaction, the trading margin and the base price of the profit and loss settlement for the unliquidated contract will be made), the *ups and downs* (the difference between the spot price on the trading day and the closing price on the trading day), the *closing price* (the last trade on the trading day of the stock option), the *volume*, the *open interest* (a specific market at the end of a trading day, The number of contracts that are held by multiple parties or shorted by empty parties). The attributes included in the futures are *opening price, highest price, lowest price, closing price, settlement price, ups, and downs, basis* (the spot price and futures of certain specific commodities at certain times and places of the difference in price), *the volume, open position*.

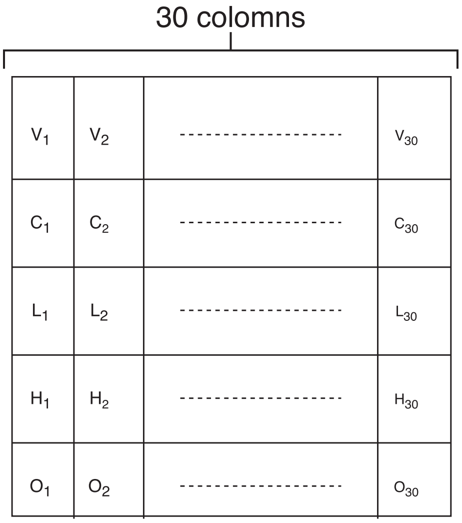
In this article, input images are going to be produced by (collect) the indexes vector information for stocks in 30 days. An example is shown in Figure 1. The x-axis indicates the dates of continuous periods for input images (x-axis is set as 30 consecutive days). The y-axis means the historical datasets for stocks which are the indexes (opening value, highest value, lowest value, and so forth) of historical prices, futures, and options in these dates for input images. In this article, we are going to put this in as a picture, as an input vector, and this is going to be a predicted value.

In the experiments, the article predefined the width of a sliding window is 30 days in the sequences of stock indexes. Each window can generate an input image, move to the next window by shifting one date, and establish the following image. Finally, the method can get the sequence of the input images. It can be expressed as $y_1, y_2, \ldots, y_m$. Two adjacent images mean that their sliding windows are placed in different ways by 1 day. The labeling function is described in



1	2	3	4	5	6	...	29	30
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

FIGURE 1 Example of the input image



V_1	V_2	-------------------	V_{30}
C_1	C_2	-------------------	C_{30}
L_1	L_2	-------------------	L_{30}
H_1	H_2	-------------------	H_{30}
O_1	O_2	-------------------	O_{30}

FIGURE 2 An example of the image established by historical prices

Equation (2).

$$Z_t = \begin{cases} +1, & l_t >= 0.01 \\ -1, & l_t < -0.01 \\ 0, & \text{Others.} \end{cases} \tag{2}$$

Here, Z_t is indicated the label of the sample y_t, l_t is the percentage change in the price of the current stock on the next date. When l_t is greater than or equal to 0.01, it will be defined as +1 (price increasing), if l_t is less than -0.01, it will be defined as -1 (price decreasing); otherwise, it will be labeled as 0, meaning within this range.

3.3.2 | Detail architectures of input images

In the proposed SSACNN, the architectures of input images are complicated. Therefore, this session provides a detailed description of architectures for the input images using in the experiments. First, the images generated by the historical prices, futures, and options are explained separately below. Note that the description of the architectures focuses on discussing the data structure here; the values will all be normalized by the method proposed above.

Historical Price Image. According to the above definition, the image should include the *opening price*, *highest value*, *lowest value*, *closing price*, and *volume* for a specific stock in 30 days. We simply put these values in 1 day in a column with the same order and extend 30 days as 30 columns to establish an image. An illustrated example is shown in Figure 2. Note that O_n, H_n, L_n, C_n, and V_n are respectively, the *opening price*, *highest value*, *lowest value*, *closing price*, and *volume* at *n*th day.

Historical Finance Data Image. It is similar with historical prices, also includes the *opening price*, *highest value*, *lowest value*, *closing price*, and *volume*. Nonetheless, a specific underlying asset (stock) can have several futures products

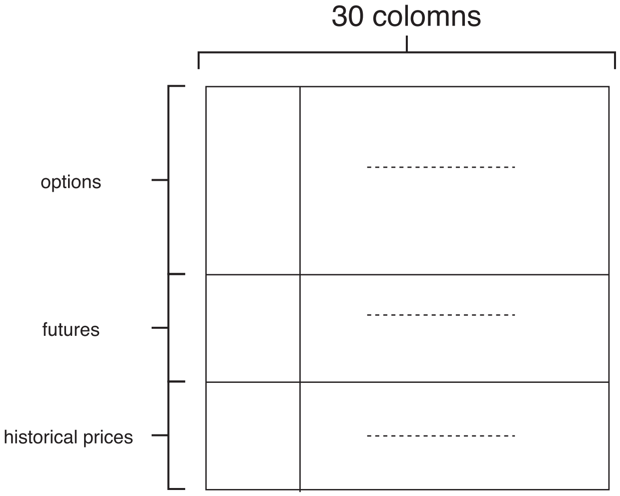


FIGURE 3 An example of the image established by combination data

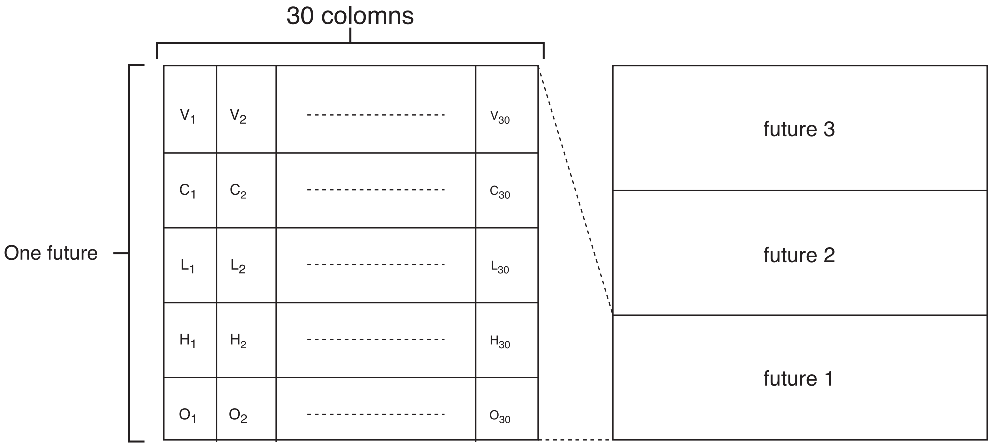

30 colomns
One future
V_1 V_2 ------------------ V_{30}
C_1 C_2 ------------------ C_{30}
L_1 L_2 ------------------ L_{30}
H_1 H_2 ------------------ H_{30}
O_1 O_2 ------------------ O_{30}
future 3
future 2
future 1

FIGURE 4 An example of the image established by futures

with different expiration dates (Figure 3). In the proposed framework, we select three futures whose expiration dates are closest to the current date. The related attributes of each future in 1 day will be listed in one column and also extend 30 days as a matrix. An illustrated example is shown in Figure 4.

Options Image. The situation of options is more complicated than futures and historical prices. Because there are two kinds of options (call and put rights) in the options market. Here, we only select the data from the nearly month options for a specific stock. In the proposed method, it selects 10 different options (10 call rights and 10 put rights) whose settlement prices are most close to the current price of the underlying asset (stock) and gets the attributes from these options to build the image. The attributes include *settlement price*, *closing price*, *volume*, and *open interest*. It is similar to the previous two images, also extends 30 days data as a matrix. An illustrated example is shown in Figure 5.

Combination Image. The combination image is the final input of the proposed framework. It combines the information of historical prices, futures, and options. In fact, it simply binds the previous three images to generate a new image. An illustrated example is shown in Figure 3.

3.3.3 | Advanced SSACNN optimization framework

Because various factors influence the stock markets, it is a difficult task for stock markets that is to find trading signals. In the past, a tradition method is using trading strategies to get trading signals, which be produced by fundamental or indicators.[7,13,40] With the develop of deep learning NN, the CNN is proposed which a the most famous algorithm.[4,30–32] It mainly includes several convolutional layers, pooling layers, and full connection layers. It has been proved to have the ability to identify images.

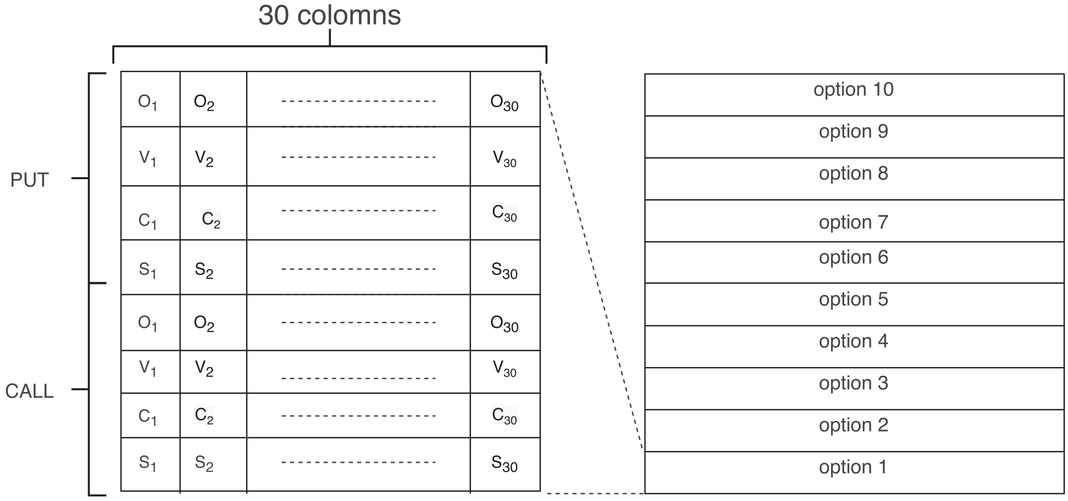


FIGURE 5 An example of the image established by options (S_n, C_n, V_n, and O_n are the *settlement price*, *closing price*, *volume*, and *open interest* at nth day)

The convolutional layer is usually used to make the convolution operation on the dataset. The input can be regarded as a vector. The filter that it uses is another vector and the convolution operation is an algorithm to measure the changes caused by the application of the filter on the input. The size of the filter shows the coverage of the filter. Each filter uses a set of shared weights to perform a convolution operation. The weights are updated during the process of training.

Before entering the next layer, an activation function is usually be added in the output of each filter. Nowadays, Relu is better than other activation functions, because it can solve nonlinear problems better. It is shown in Equation (4). In general, the input image matrix and convolution kernel are all square matrices. Here, let the input matrix size is w, The convolution kernel size is k, The pace is s, the number of zero filling layers is p, The formula for calculating the size of the convoluted feature map is as follows: Equation (5)

$$V_{a,b}^{L} = \iota\left(\sum_{m=0}^{K-1}\sum_{n=0}^{K-1} w_{m,n} V_{a+m,b+n}^{L-1} + \text{biase}^{L-1}\right). \tag{3}$$

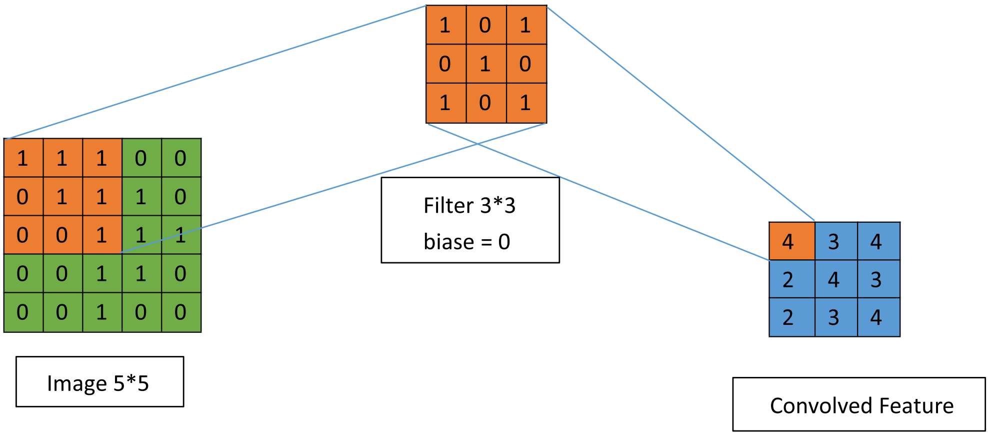
In Equation (3), $V_{a,b}^{L}$ is the value of layer L at row a, column b, $w_{m,\,n}$ is the weight of convolution filter at row m, column n. ι is a activation function and bias^{L-1} is represent the bias of $L-1$. An example is shown below. The input layer $L-1$is set as a 5×5 matrix and use the 3×3 convolutional filter. The layer of input L is calculated by Equation (3), which is set as 3×3. Figure 6 shows an example that defines a filter (3×3) to the input vector (5×5) for obtaining the vector of the next layer (3×3).

$$f(x) = \max(0, x), \tag{4}$$

$$w' = \frac{w + 2p - k}{s} + 1. \tag{5}$$

The pooling layer is also called down-sampling, which is opposite to up-sampling. The convoluted characteristic map usually needs a pooling layer to reduce the amount of data. Because this operation is a way of handling the overfitting problem. When a model of training makes too fit to the training data, overfitting is a case that arises. Using the pooling layer can help to reduce the risk of overfitting. All values in the pool window are converted to only one value. This transformation reduces the input size of the following layers, thus reducing the number of parameters that the model must learn, thus reducing the risk of overfitting. The maximum pool is the most common pool type, where you select the maximum value in a window.

At the ultimate layer on CNN, there is a traditional neutral network which is called a fully connected layer. It is responsible for converting features extracted in the previous layer to the final output. The relationship between two adjacent



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	0	1	0	0

1	0	1
0	1	0
1	0	1

4	3	4
2	4	3
2	3	4

FIGURE 6 An example of the convolution network [Color figure can be viewed at wileyonlinelibrary.com]

layers is defined by Equation (6).

$$V_a^b = \iota\left(\sum_K V_K^{b-1} w_{K,a}^{b-1} + \text{biase}^{b-1}\right). \tag{6}$$

In Equation (6), V_a^b is the value of layer b in neuron a, ι is an activation function, and $w_{K,a}^{b-1}$ is a weight which connect between neuron K from layer *b-1* and neuron a from layer b.

Based on the CNN, the algorithm first transforms the data into an image which uses this feature of CNN by References 4,32. Except for pooling, this article also uses other technical operations, including dropout and norm that were used in deep NNs. Because the technique of dropout is to avoid the framework from too much learning of the data. In training, this only needs to sample the parameters of the weight layer randomly according to a certain probability p, and take this subnetwork as the target network of this update. It can be imagined that if the whole network has n parameters, then the number of available subnetworks is 2^n. Moreover, when n is large, the subnetworks used for each iteration update will not be repeated basically, so as to avoid a certain network being excessively fitted to the training set. The norm layer normalizes the local area of input to achieve the effect of "side suppression." The pseudo-code of the proposed algorithm is shown in Algorithm 1.

Algorithm 1. Algorithm 1

```
Require: $d$ is the data of training; $K$ is the data of testing; $Z$ is the number of iteration; $A$ is batch size; Algorithm SGD is named $Adam$.
Ensure: the train model $m$; evaluation result $accuracy$
Initialize algorithm
$d \leftarrow Initializealgorithm$
$S \leftarrow$ (split $d$ in equal parts of $A$)
for each round $t = 1, 2, \dots, z$ do
    $\{verify, train\} \leftarrow \{S_t, S - S_t\}$
    $(tf, vf) \leftarrow$ (generate feature of $train$ and $verify$)
    $m_t \leftarrow \text{modelFit}(Adam, tf)$
    $r_t \leftarrow \text{modelEvaluate}(m_t, vf)$
end for
$m \leftarrow \text{bestModel}$
$K \leftarrow m$
$accuracy \leftarrow \text{modelEvaluate}(m, test)$
```

The proposed method is presented, which transfers a period of the stock indexes value to a sequence of images based on the proposed method, These images will be the input images for the CNN framework. Input: (collect) the indexes vector information for stock in 30 days × the variables of each day. Then, input the "input image" to convolutional layer,

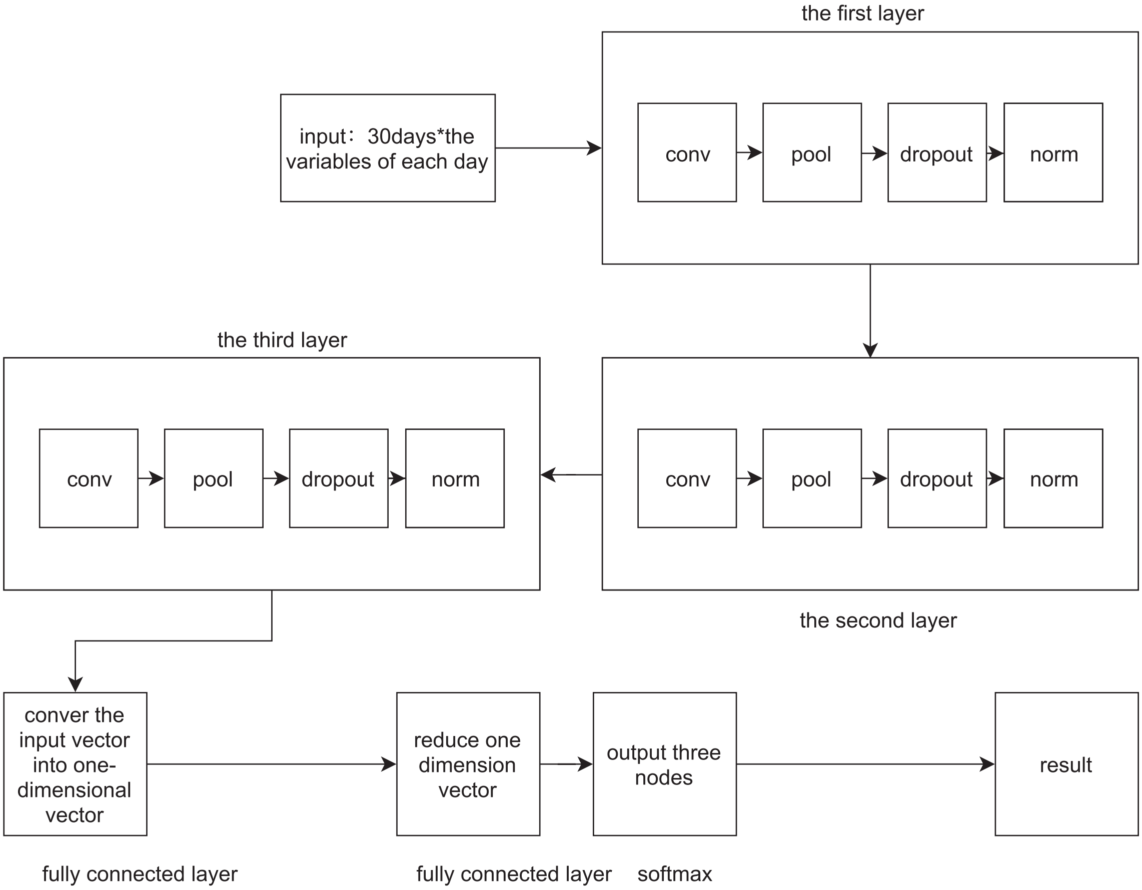


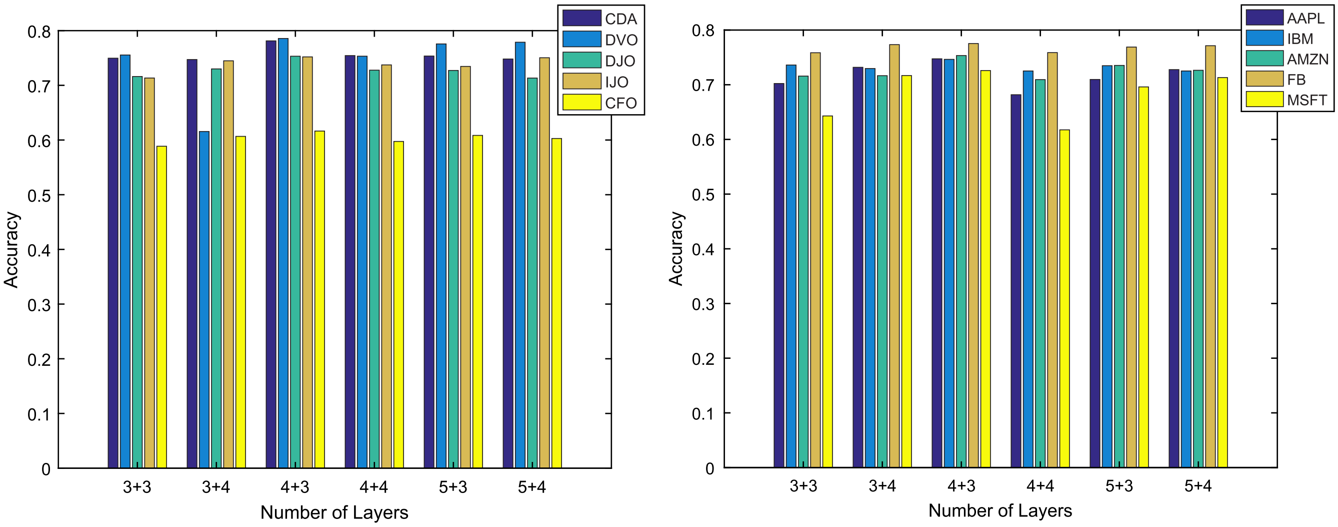
FIGURE 7 The designed framework for stock trading prediction

pool layer, dropout layer, norm layer and initially loop this sequence three times (here, it defines convolutional layer, pool layer, dropout layer, norm layer as a layer). According to our experimental observations, when the CNN is used for image recognition, the best effect can be obtained when the size of the convolutional kernel is set as 3 × 3 and the size of the pooling layer is set as 2 × 2. Therefore, to achieve the highest accuracy of prediction, we adopted the convolution kernel size as 3 × 3 and the pooling layer size as 2 × 2. Finally, for the input of full connection layer and in the last full connection layer, we added the *softmax*function, the probability of each output is analyzed with the *softmax* function and set a label for the input image. The label by the same process described Equation (2), and the specific framework is shown in Figure 7.

4 | EXPERIMENTAL RESULTS

In this experiment, five stocks of America and Taiwan are used as the input data. For other stock markets, the reference value of the fundamental analysis of stocks in America and Taiwan is relatively high. In order to increase the accuracy of the prediction, the data also uses five levels of the price in 1 day. This experiment is carried out under the windows system of TensorFlow by using Python language.

This article proposed a different framework for stock prediction in classification. To compare the performance of the classification frameworks, We have added several layers of convolutional layers, pooled layers, and fully connected layers based on the original design model. Results of the accuracy of datasets of each nodal are presented in Figure 8. It clearly can show that the accuracy of the four-layer convolutional layer, the pooled layer, and the three-layer full connection layer are the highest, which indicates that samples with a more significant profit of an eventual fall or rise show powerful dependency between labels and features. And it shows that the higher the number of layers, the higher the accuracy is not necessarily the highest. Because the accuracy of the network is not only related to the depth of the network but also



(A) Taiwanese stocks

(B) American stocks

FIGURE 8 The forecasting accuracy of different layers across the historical data of all stocks in the training dataset [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 5 The numbers of levels tested in different parameter settings for SSACNN

Parameters	Levels
Epochs	5000, 10,000 … 30,000
Learning rate	0.1, 0.01, 0.001, 0.00001
Activation functions	relu/tanh
Hidden layers	3, 4, 5
FC layers	3, 4
Number of neurons	64,128 … 1024

Abbreviation: SSACNN, stock sequence array convolutional neural network.

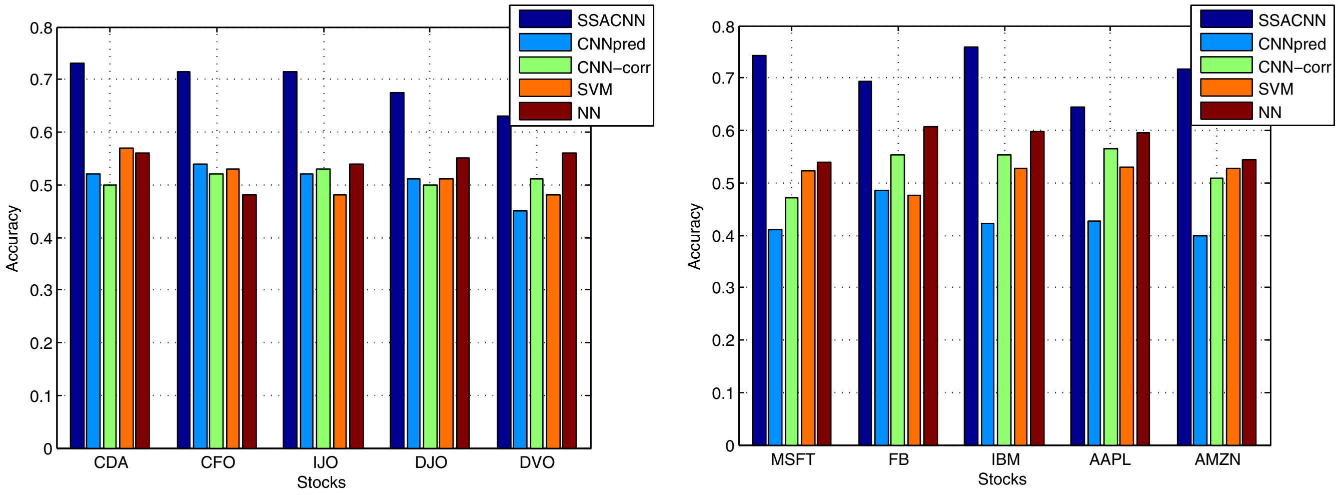
related to the parameters. We designed different parameter settings for the proposed algorithm. The relevant parameters are shown in Table 5.

In the previous part, the proportion of the testing period is 60%, 20% is for the training period, and 20% for the validation period. To further evaluate the effectiveness of our framework, the article uses different windows to divide the testing period and training period. The testing period with the proportion of 30%, 40%, 50%, 60%, 70% (the training period with the proportion of 70%, 60%, 50%, 40%, 30% correspondingly) is used for generating data and sampling sets by the same process described before. Results are implied in Table 6. The a_{i1}, a_{i2}, a_{i3}, a_{i4}, and a_{i5}, respectively, present the training data accounted for 60%, 70%, 50%, 40%, and 30% of the data. The accuracy of the highest value is obtained when the ratio of the testing set to the training set is 0.5 by the three layers of full connection layers and the four layers of convolutional and pool layers.

In order to further assess the performance of the proposed algorithm (SSACNN) on the market of stock prediction. The experiments simulate that based on predictive trading to test whether predictions made by SSACNN can make margins. The experimental design mainly includes the main factors: market classification and the selection of attributes (options and futures). Moreover, the article also provides more comparisons of stock prediction, which include 10 financial stocks, 3 attributes of each stock (options, futures, and historical data), and five different classification algorithms, CNNpred, CNN-corr, NN, SVM, and the proposed SSACNN. The first part of the experiments is that setting the historical prices as the input data for all compared algorithms. The second and third parts use the futures data and the options data as the input data instead of the historical prices. The final part combines the historical prices, the futures data, and the options data as the input data to perform all algorithms. Note that, all of the previous works did not use the futures or options as

TABLE 6 Performance of the proposed algorithm with different proportions for the historical datasets of stocks in the training dataset

Stocks	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}
CDA	0.766165	0.761273	0.869565	0.825331	0.771755
CFO	0.685262	0.61509	0.744099	0.720946	0.668743
IJO	0.777083	0.726916	0.847118	0.867168	0.816719
DJO	0.773072	0.755673	0.804141	0.810717	0.795597
DVO	0.800833	0.685205	0.882707	0.85654	0.774214
AAPL	0.767657	0.740678	0.860714	0.819403	0.847333
AMZN	0.762046	0.716994	0.81791	0.779602	0.800667
FB	0.808911	0.777684	0.886111	0.859702	0.870667
IBM	0.756766	0.715537	0.817064	0.823383	0.764667
MSFT	0.743234	0.708475	0.782937	0.8084478	0.804667



(A) Taiwanese stocks

(B) American stocks

FIGURE 9 The prediction accuracy for all five model specifications (SSACNN, CNNpred, CNN-corr, SVM, NN), using the data of history. CNN, convolutional neural network; NN, neural network; SSACNN, stock sequence array convolutional neural network; SVM, support vector machine [Color figure can be viewed at wileyonlinelibrary.com]

the input data before. In our experiments, we extend the previous works and show their performance in the situations by applying the futures and options data. Moreover, due to the limitation of CNN-corr, it is only compared in the first part. The reason is that CNN-corr uses the original candlestick chart as the input graph but the futures and options include several target prices (options) or different periods (future); the data cannot transfer into a signal candlestick chart.

The proposed framework uses the best prediction model. The number of convolution layers is four and the number of full connection layers is three. First, the article uses the historical date-sets to comparing with the other method (CNNpred, CNN-corr, SVM, NN). Prediction experiments are carried out in the stock of Taiwanese and America, and the prediction results between the two markets are compared. Figure 9 is the prediction results of the set of experiments. It can clearly show that SSACNN algorithm is relatively good. For the individual historical data, the traditional NN prediction is comparatively better than the rest. Because CNN-corr and CNNpred are easy to generate considerable noise, and the accuracy of SVM is relatively low due to the nonconstant sensitivity to the quality of the training set.

Second, this article aims to test the prediction accuracy further and uses the date-sets of futures and options to compare with the other method (CNNpred, SVM, NN). Because futures and options, as the leading indicators of stocks, can predict a trend of stock development in the future and the experiments of prediction are carried out in the stock of Taiwanese and America (The American stock market does not exist futures for any stock, therefore, we just show the results of the Taiwanese stock market in the field of futures experiment.. The results of the predictions between the two markets are

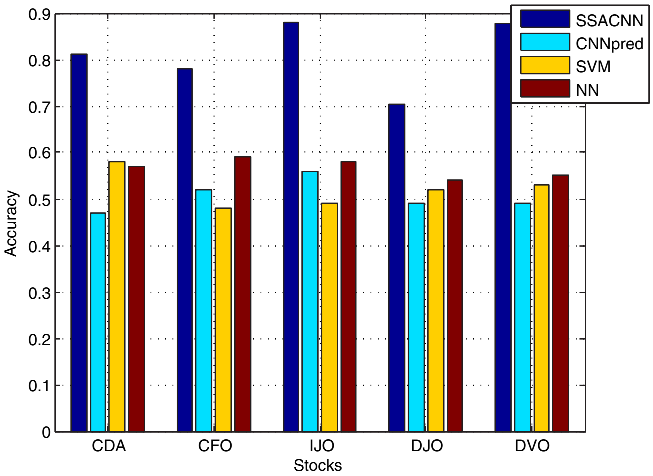
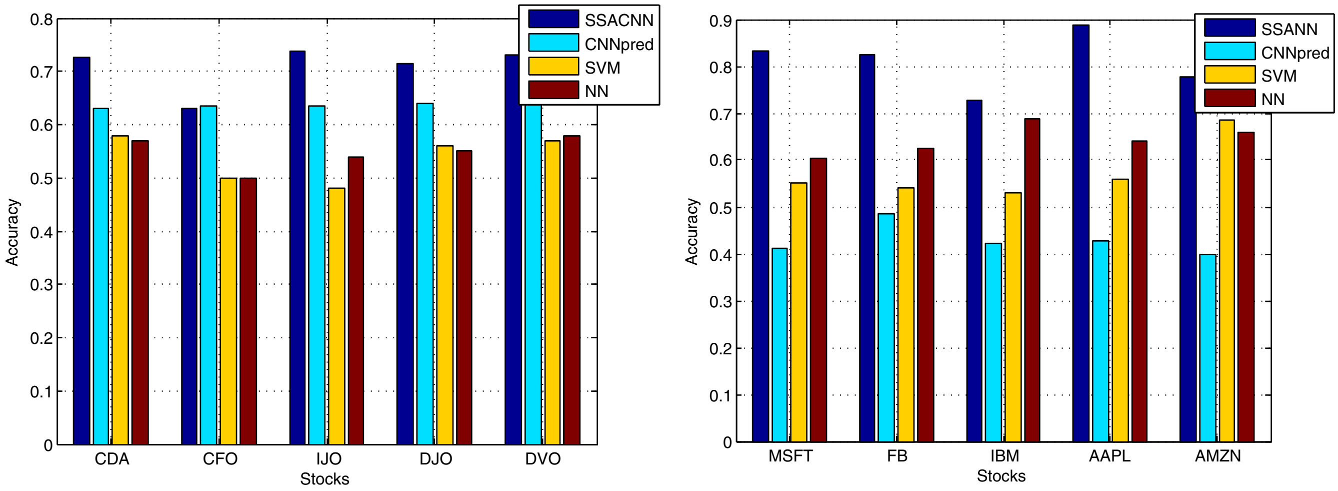


FIGURE 10 The prediction accuracy for all four model specifications (SSACNN, CNNpred, SVM, NN), using the futures data. CNN, convolutional neural network; NN, neural network; SSACNN, stock sequence array convolutional neural network; SVM, support vector machine [Color figure can be viewed at wileyonlinelibrary.com]



(A) Taiwanese stocks

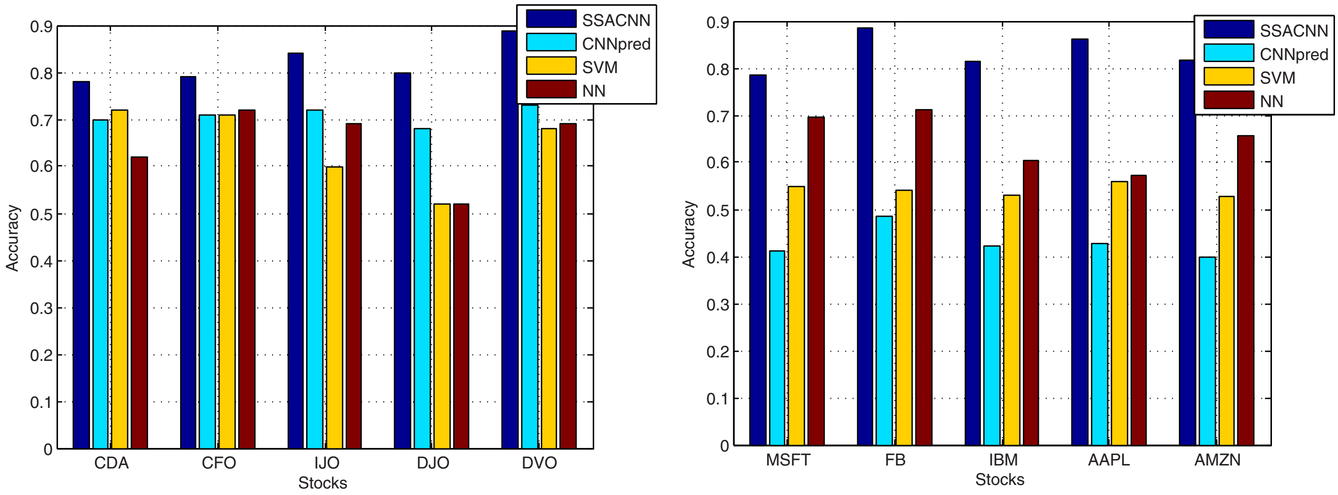
(B) American stocks

FIGURE 11 The prediction accuracy for all four model specifications (SSACNN, CNNpred, SVM, NN), using the option data. CNN, convolutional neural network; NN, neural network; SSACNN, stock sequence array convolutional neural network; SVM, support vector machine [Color figure can be viewed at wileyonlinelibrary.com]

compared. Figures 10 and 11 is the prediction results of the set of experiments. From Figures 10 to 11, it can be seen that the best result is obtained by the proposed framework. The proposed SSACNN significantly performs better than all the other methods (i.e., CNNpred, SVM, and NN). It has proved that these two leading indicators imply the trend of stocks. Therefore, compared with setting the individual forecast historical data as the input data, the potential of the combination data of the historical prices and leading indicators are expected better than using purely historical data. The following parts show that the basic analysis for the combination data.

In this part, the experiments combine historical data, the datasets of options, and futures to comparing with the other method (CNNpred, SVM, NN). Figure 12 is the prediction results of the set of experiments. It can clearly show that the historical data and futures options to achieve better prediction accuracy. What is more, the accuracy of all algorithms is improved obviously. It further explained that the fundamental analysis is more, the accuracy is higher. Moreover, whether the algorithm proposed in this article predicts historical data, future, and option separately, or combines the three, the prediction accuracy is higher than other algorithms.

As can be seen from Figures 9 to 12, it showed that the proposed algorithm is better than the other algorithms in the different financial markets. It can also be clearly found that the more input indicators and information (historical prices, future, and option), the accuracy can be higher obtained. The proposed algorithm, thus, is proved that it has the ability to grab the critical information from these leading indicators and historical prices to predict the trend of the target stock.



(A) Taiwanese stocks

(B) American stocks

FIGURE 12 The prediction accuracy for all four model specifications (SSACNN, CNNpred, SVM, NN), using the option, future, and history data. CNN, convolutional neural network; NN, neural network; SSACNN, stock sequence array convolutional neural network; SVM, support vector machine [Color figure can be viewed at wileyonlinelibrary.com]

5 | CONCLUSION

This article presents a novel CNN that can be used in stock market prediction. Many different techniques are used in SSACNN for feature extraction to ensure the extracted knowledge has a high utility. Using multiple filters, propose classification occurred to assist in classification-based market prediction. Through verified experimental results, the work of SSACNN outperformed both SVM-based approaches as well as other CNN-based work in terms of prediction accuracy. The proposed algorithm named SSACNN that using the CNN for feature extraction on financial time series, and based on classification for prediction. By multifilters feature, characterizes were extracted and used for classification-based market prediction, and a framework using a CNN was verified to be better than the statistical methods and traditional CNN on the prediction task. The best prediction result from SSACNN has better performance than SVM and the traditional CNN. In the proposed SSACNN, the data is directly integrated into a matrix to avoid too much dispersion of the data and reduce the useless information. It also further refer to some leading index to enhance the performance of predicting the trend of stocks. In total, the effectiveness of the stock price prediction is improved effectively in this framework.

The effectiveness of deep learning methodologies on stock market prediction was proved, there are still some promising directions in the future. For example, the LSTM model might have more potential to get better performance than the CNN model. In forecast-based trading, this method has advantages in terms of simulating profitability, and stability is to be improved. In this article, the proposed framework predicts the trend of the stock price in the future. However, it does not provide the actual trading indicators to the users. It still relies on users' experiences to make the transaction decision in stock markets. In the future, we will also involve an expert system based on the proposed model to provide the clear signals or transaction rules in stock trading.

ACKNOWLEDGMENT

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada through a Discovery Grant held by Dr. Gautam Srivastava.

AUTHOR CONTRIBUTIONS

Conceptualization, M.T.W. and Z.C.L.; software, M.T.W. and Z.C.L.; Formal analysis, M.T.W., Z.C.L., M.H.T, G.S and J.C.W.L; Methodology, M.T.W., Z.C.L. and G.S,; Writing-original draft, M.T.W. and Z.C.L.; Writing-review and editing, M.T.W., Z.C.L., M.H.T, G.S and J.C.W.L.

ORCID

Gautam Srivastava https://orcid.org/0000-0001-9851-4103
Jerry Chun-Wei Lin https://orcid.org/0000-0001-8768-9709

REFERENCES

1. Rani S, Sikka G. Recent techniques of clustering of time series data: a survey. *Int J Comput Appl*. 2012;52(15):1-9.
2. Zhong X, Enke D. Forecasting daily stock market return using dimensionality reduction. *Expert Syst Appl*. 2017;67:126-139.
3. Tsai HH, Wu ME, Wu WH. The information content of implied volatility skew: evidence on Taiwan stock index options. *Data Sci Pattern Recognit*. 2017;1(1):48-53.
4. Gunduz H, Yaslan Y, Cataltepe Z. Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowl-Based Syst*. 2017;137:138-148.
5. Hagenau M, Liebmann M, Neumann D. Automated news reading: stock price prediction based on financial news using context-capturing features. *Decis Support Syst*. 2013;55(3):685-697.
6. Kim KJ, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst Appl*. 2000;19(2):125-132.
7. Kuo SY, Kuo C, Chou YH. Dynamic stock trading system based on quantum-inspired tabu search algorithm. Paper presented at: Proceedings of the IEEE Congress on Evolutionary Computation. Cancun, Mexico: IEEE; 2013:1029-1036.
8. Long W, Lu Z, Cui L. Deep learning-based feature engineering for stock price movement prediction. *Knowl-Based Syst*. 2019;164:163-173.
9. Khaidem L, Saha S, Dey SR. Predicting the direction of stock market prices using random forest; 2016. arXiv Preprint arXiv:1605.00003.
10. Hu P, Pan JS, Chu SC, Chai QW, Liu T, Li ZC. New hybrid algorithms for prediction of daily load of power network. *Appl Sci*. 2019;9(21):4514.
11. Pan JS, Hu P, Chu SC. Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power. *Processes*. 2019;7(11):845.
12. Chen CH, Hsieh CY. Actionable stock portfolio mining by using genetic algorithms. *J Inf Sci Eng*. 2016;32(6):1657-1678.
13. Chen CH, Lu CY, Lin CB. An intelligence approach for group stock portfolio optimization with a trading mechanism. *Knowl Inf Syst*. 2019;62:1-30.
14. Chen CH, Yu CH. A series-based group stock portfolio optimization approach using the grouping genetic algorithm with symbolic aggregate approximations. *Knowl-Based Syst*. 2017;125:146-163.
15. Lin JCW, Shao Y, Fournier-Viger P, Hamido F. BILU-NEMH: a BILU neural-encoded mention hypergraph for mention extraction. *Inf Sci*. 2019;496:53-64.
16. Lin JCW, Shao Y, Zhou Y, Pirouz M, Chen HCA. Bi-LSTM mention hypergraph model with encoding schema for mention extraction. *Eng Appl Artif Intell*. 2019;85:175-181.
17. Lin JCW, Shao Y, Zhang J, Yun U. Enhanced sequence labeling based on latent variable conditional random fields. *Neurocomputing*. 2020;403:431-440.
18. Djenouri Y, Srivastava G, Lin JCW. Fast and accurate convolution neural network for detecting manufacturing data. *IEEE Trans Ind Inform*. 2020;1-1.
19. Huang KW, Lin CC, Lee YM, Wu ZX. A deep learning and image recognition system for image recognition. *Data Sci Pattern Recognit*. 2019;3(2):1-11.
20. Gardner MW, Dorling S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmos Environ*. 1998;32(14-15):2627-2636.
21. Baba N, Kozaki M. An intelligent forecasting system of stock price using neural networks. Paper presented at: Proceedings of the International Joint Conference on Neural Networks. Baltimore, MD, USA: IEEE; 1992;1:371-377.
22. Oliveira dFA, Nobre CN, Zarate LE. 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*. 2013;40(18):7596-7606.
23. Ding X, Zhang Y, Liu T, Duan J. Deep learning for event-driven stock prediction. Paper presented at: Proceedings of the International Joint Conference on Artificial Intelligence; 2015; ACM, New York, NY.
24. Chen K, Zhou Y, Dai F. A LSTM-based method for stock returns prediction: a case study of China stock market. Paper presented at: Proceedings of the International Conference on Big Data (Big Data). Santa Clara, CA, USA: IEEE; 2015:2823-2824.
25. Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res*. 2018;270(2):654-669.
26. Yong BX, Rahim MRA, Abdullah AS. A stock market trading system using deep neural network. Paper presented at: Proceedings of the Asian Simulation Conference; 2017:356-364; Springer, New York, NY.
27. Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS One*. 2017;12(7):e0180944.
28. Cai X, Hu S, Lin X. Feature extraction using restricted Boltzmann machine for stock price prediction. Paper presented at: Proceedings of the International Conference on Computer Science and Automation Engineering (CSAE). Zhangjiajie, China: IEEE; 2012;3:80-83.
29. Zhu C, Yin J, Li Q. A stock decision support system based on DBNs. *J Comput Inf Syst*. 2014;10(2):883-893.
30. Di Persio L, Honchar O. Artificial neural networks architectures for stock price prediction: comparisons and applications. *Int J Circ Syst Signal Process*. 2016;10:403-413.

31. Hoseinzade E, Haratizadeh S. CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Syst Appl*. 2019;129:273-285.
32. Siripurapu A. Convolutional networks for stock trading. *Stanford Univ Dep Comput Sci*. 2014;1-6.
33. Taylor MP, Allen H. The use of technical analysis in the foreign exchange market. *J Int Money Financ*. 1992;11(3):304-314.
34. Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. Paper presented at: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. Vancouver, BC, Canada: IEEE; 2013:6645-6649.
35. Williams RJ, Zipser D. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput*. 1989;1(2):270-280.
36. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278-2324.
37. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780.
38. Pang X, Zhou Y, Wang P, Lin W, Chang V. An innovative neural network approach for stock market prediction. *J Supercomput*. 2018;76:1-21.
39. Nelson DM, Pereira AC, Oliveira DRA. Stock market's price movement prediction with LSTM neural networks. Paper presented at: Proceedings of the International Joint Conference on Neural Networks. Anchorage, AK, USA: IEEE; 2017:1419-1426.
40. Chou YH, Kuo SY, Chen CY, Chao HC. A rule-based dynamic decision-making stock trading system based on quantum-inspired tabu search algorithm. *IEEE Access*. 2014;2:883-896.

How to cite this article: Wu JM-T, Li Z, Srivastava G, Tasi M-H, Lin JC-W. A graph-based convolutional neural network stock price prediction with leading indicators. *Softw Pract Exper*. 2020;1–17. https://doi.org/10.1002/spe.2915