Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 53-5 (2020) 692–697

A Tool based on ML-driven Graphical Model for Stock Price Prediction by Leading Indicators

Jimmy Ming-Tai Wu *, **Zhongcui Li** *, **Gautam Srivastava** **, **Jerry Chun-Wei Lin** ***

* *Shandong University of Science and Technology, China (e-mail: wmt@wmt35.idv.tw, 17685458562@163.com).*
** *Brandon University, Canada (e-mail: srivastavag@brandonu.ca)*
*** *Western Norway University of Applied Sciences, Norway (e-mail: jerrylin@ieee.org) Corresponding author*

Abstract: Stock prediction has become an emerging issue in recent decades and many studies have incorporated it with social systems to provide a better accuracy for the prediction results. Machine learning (ML) model is widely studied and developed to show better performance in data analytics and prediction, which can be also applied in the stock markets for the price prediction.To be better applied in the stock market for price predication, it is necessary to finalize a ML-driven toolbox that can be easily adopted into the stock market. In this paper, aiming at the task of time series (financial) feature extraction and prediction of price movements, a new convolutional novel neural network to improve the prediction accuracy of stock trading is proposed. The proposed model is called SSACNN, short form of stock sequence array convolutional neural network that 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 CNN framework. In our experimental results, 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: stock history, option and future of stocks, convolutional neural network, prediction.

1. INTRODUCTION

The financial market is a market co-existing with the commodity market as well as the labor market. Currently, many different financial time series forecasting methodologies are known to exist (Rani and Sikka, 2012; Zhong and Enke, 2017) 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. Various methods and data sources are used for stock market forecasting (Gunduz et al., 2017; Hagenau et al., 2013; Kim and Han, 2000; Kuo et al., 2013; Long et al., 2019). 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 (Kim and Han, 2000). Over time, traditional forecasting methods include statistical methods, linear discriminant analysis, quadratic discriminant analysis, random forests (Gholamian and Davoodi, 2018), logistic regression and evolutionary computation algorithms (Hu et al., 2019; Pan et al., 2019). Recently, genetic algorithm is a tool and technology that can extract features from the original financial data according to a set of variables (Chen et al., 2019; Chen and Yu, 2017) that has shown some success.

In recent years, neural networks play an increasingly important role in social life and are often used in image recognition, speech recognition, and text recognition. The basic principle of neural networks is to use a series of available neural network feature extractors to design a feature-network for specific samples, so as to achieve the desired effect and complete its own project. In (de Oliveira et al., 2013), the authors built a neural framework for the financial market. Based on the application of traditional neural networks to the financial field, deep learning is reflected in it in (Ding et al., 2015). As far as we know, Convolutional Neural Networks (CNN) have been applied in some studies of stock market prediction (Hoseinzade and Haratizadeh, 2019; Siripurapu, 2014; Gunduz et al., 2017). For example, a previous CNN work uses stock candlestick charts to be the input image and feeds into the input layer directly in(Siripurapu, 2014), or is to seek out a general framework for mapping the historical data of a market to its future fluctuations in (Hoseinzade and Haratizadeh, 2019), 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 (Di Persio and Honchar, 2016; Gunduz et al., 2017) made use of a CNN which was able to use technical indicators for each sample.

10.1016/j.ifacol.2021.04.219

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. Our main progresses can be summarized as follows:

(1) 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 paper proposes a two-dimensional tensor as input data in the proposed CNN network. It then uses a feature extraction method to train the system to predict the stock market.
(3) The algorithm proposed in this paper is compared with several previous algorithms, which proves that this algorithm can better avoid too much noise, save more useful information, and prevent over fitting phenomenon.

2. RELATED WORK

We can define the financial time series as the organization of values of random financial variables in a certain period. Rani and Sikka think time series clustering is one of the essential concepts of data mining. They are used to understand the mechanism of generating time series and predict the future value of a given time series (Rani and Sikka, 2012). However, the time and frequency in the stock are often different, so it is not easy to predict the stock. Through considerations for stock prices, Chen at al. present an improved methodology to give a more feasible stock portfolio application for investors. Furthermore, a group stock portfolio (GSP) that is sequence based was derived by the authors use to give sound investment advice (Chen and Yu, 2017). The authors also propose an optimization algorithm that uses many aspects of evolutionary computation (Chen and Hsieh, 2016).

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 (Taylor and Allen, 1992) is a postmortem analysis, which uses historical data to predict the future and data, graphics, and statistical methods to explain problems. At the same time, features extracted by designed indicators are based on presumed patterns so that some information may be lost in this approach.

In recent years, neural network plays an increasingly important role in social life. In deep learning, it provides a lot of units, for example, recurrent unit (Williams and Zipser, 1989), convolutional unit (LeCun et al., 1998), long-short memory term unit (Hochreiter and Schmidhuber, 1997) with different characteristics for feature extraction on samples. Pang et al. introduced an LSTM based method used to predict stock market action (Graves et al., 2013; Pang et al., 2018). 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 (Siripurapu, 2014), 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 (Hoseinzade and Haratizadeh, 2019). Nevertheless, the input images contain too much noise, useless information, and are prone to over-fitting. Recurrent neural networks (RNN) is also applied in stock price prediction. In (Nelson et al., 2017), it uses the long-term and short-term memory neural network (LSTM) to deal with time series and applies it to stock forecasting. 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 support vector machine classification is very sensitive to the quality of the training set. Therefore, Zhong proposed an SVM based model and avoided to use complex financial data to ignore this problem (Zhong and Enke, 2017).

3. STOCK SEQUENCE ARRAY CONVOLUTIONAL NEURAL NETWORK

3.1 Data Sets

Before making stock forecasts, an index sequence y_1, y_2, y_3, …, y_t generated is set as input data. It includes the historical data of prices and two leading indexes, future and option of a stock. In the experiments, some stocks from the American stock market and Taiwanese stock market will be applied. The so-called leading indicator of stock is the statistics of 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 paper 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 paper shows the relevant information of five stocks in Taiwan. Its attributes include the historical data of stock and the attribute of the stock's future and option that can be seen in Table 1, Table 2 and Table 3.

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	3,635	3,825	3,880	3,635	…

Table 2. The future 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	3,660	3,815	3,865	3,635	…

In Table 1 and Table 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 future (present price, opening price, closing price, the highest price, and

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	...

etc.). In Table 3, Because 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, and etc. 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 will normalize all of the input values. The proposed function is given in Eq. 1:

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

where X_t is the indexes vector for time t (*open*, *high*, *low*, *close*...), $\dot{X}_t$ is the indexes vector after the normalize process. *mean*, *max* and *min* are the average value, maximal value and minimal value of the indexes 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 Eq. 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 Convolutional Neural Network

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 future transactions. Conducting two-way trading can make money even in the situation as the market is not good. On the other hand, options are similar to futures. 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 the option, 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 un-liquidated 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 paper, input images are going to be produced by (collect) the indexes vector information for stocks in 30 days. The x-axis indicates the dates of continuous periods for input images. The y-axis means the indexes of historical data-sets for stocks in these dates for input images. In the experiments, the paper predefined the width of a sliding window is 30 days in the sequences of stock indexes. Each window can generate an input image and you can move a=one date from the current window to get the next image. Finally, the method can get the sequence of the input images. It can be expressed as $y_1, y_2, /dots, y_m$. Two adjacent images mean that their sliding windows are placed in different ways by one day. The labeling function is described in Eq. 2.

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

There, 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.

Advanced SSACNN Optimization Framework In the past, a tradition method is using trading strategies to get trading signals, which be produced by fundamental or indicators(Chou et al., 2014; Kuo et al., 2013; Chen et al., 2019). With the develop of deep learning neural network, the convolutional neural network (CNN) is proposed which a the most famous algorithm (Di Persio and Honchar, 2016; Hoseinzade and Haratizadeh, 2019; Siripurapu, 2014; Gunduz et al., 2017). It mainly includes several convolutional layers, pooling layers, and full connection layers. It has been proved to have the ability to identify images.

The convolutional layer is usually used to make the convolution operation on the data-set. Actually, 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.

Here the input layer $L-1$ is a $M \times M$ matrix and use the $K \times K$ convolutional filter. So, the layer of input L is calculated by Eq. 3. Nowadays, Relu is better than other activation functions, because it can solve nonlinear problems better. It is shown in Eq. 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: Eq. 5

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	...

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

In the Eq. 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. $\imath$ is a activation function and $bias^{L-1}$ is represent the bias of *L-1*.

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

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

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 over-fitting problem. When a model of training makes too fit to the training data, over-fitting is a case that arises. Using pooling layer can help to reduce the risk of over-fitting. 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 over-fitting. 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 layers is defined by Eq. 6.

$$V_{a}^{b} = \imath\left(\sum_{K} V_{K}^{b-1} w_{K,a}^{b-1} + bias^{b-1}\right) \tag{6}$$

In Eq. 6, V_a^b is the value of layer b in neuron a, $\imath$ 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 of the CNN, the algorithm first transforms the data into an image which use this feature of convolutional neural network by Gunduz and Siripurapu et al. (Gunduz et al., 2017; Siripurapu, 2014). Except pooling, this paper also uses other technical operations, including dropout and norm that were used in deep neural networks. 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 sub-network as the target network of this update. It can be imagined that if the whole network has n parameters, then the number of available sub-networks is 2^n. Moreover, when n is large, the sub-networks 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". In this paper, 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, pool layer, dropout layer, norm layer and initially loop this sequence three times (Here, it define convolutional layer, pool layer, dropout layer, norm layer as a layer), Finally, input to 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 Eq. 2. The specific framework is shown in Figure 1.

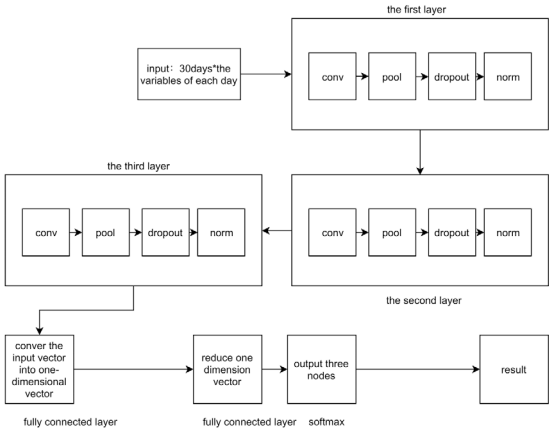

Fig. 1. Proposed SSACNN framework

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 one day. This experiment is carried out under the windows system of TensorFlow by using Python language. We designed different parameter settings for the proposed algorithm, which is shown in Table 5.

In order to further assess the performance of the proposed algorithm (SSACNN) on the market of stock prediction, the experiments combine historical data, the data-sets of options, and futures to comparing with the other method (SVM, CNNpred, CNN-corr, NN). Figs. 2 and 3 are the prediction results of the set of experiments. It can clearly

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

Parameters	Levels
Epochs	5000,10000...30000
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

show 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 explained that the fundamental 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.

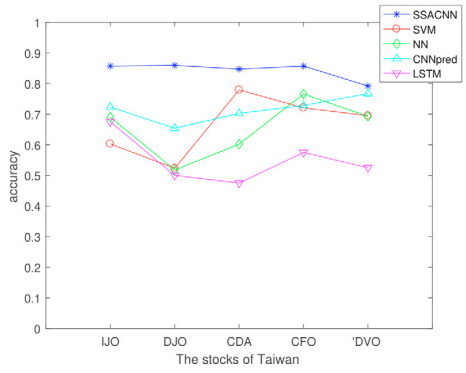

Fig. 2. Predication accuracy for Taiwanese stocks by four (SVM, CNNpred, CNN-corr, SSACNN, NN) models using the option, future, and history data

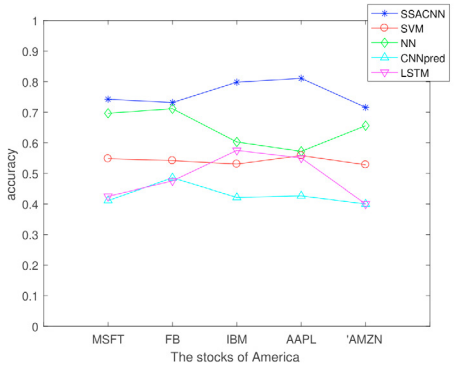

Fig. 3. Predication accuracy for American stocks by four (SVM, CNNpred, CNN-corr, SSACNN, NN) models using the option, future, and history data

As can be seen from Figs. 2-3, it is evident that the proposed algorithm is better than the other algorithm in the different financial markets. So, The proposed algorithm is thought that it is effective.

5. CONCLUSION

In this paper, the proposed algorithm named stock sequence array convolutional neural network (SSACNN) that using the convolutional neural network for feature extraction on financial time series, and based on classification for prediction. By multi-filters feature, characterizes were extracted and used for classification-based market prediction, and a framework using a convolutional neural network was verified to be better than the statistical methods and traditional convolutional neural network on the prediction task. The best prediction result from SSACNN has better performance than SVM and the traditional convolutional neural network. 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.

REFERENCES

Chen, C.H. and Hsieh, C.Y. (2016). Actionable stock portfolio mining by using genetic algorithms. *J. Inf. Sci. Eng.*, 32(6), 1657–1678.

Chen, C.H., Lu, C.Y., and Lin, C.B. (2019). An intelligence approach for group stock portfolio optimization with a trading mechanism. *Knowledge and Information Systems*, 1–30.

Chen, C.H. and Yu, C.H. (2017). A series-based group stock portfolio optimization approach using the grouping genetic algorithm with symbolic aggregate approximations. *Knowledge-Based Systems*, 125, 146–163.

Chou, Y.H., Kuo, S.Y., Chen, C.Y., and Chao, H.C. (2014). A rule-based dynamic decision-making stock trading system based on quantum-inspired tabu search algorithm. *IEEE Access*, 2, 883–896.

de Oliveira, F.A., Nobre, C.N., and Zarate, L.E. (2013). Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index–case study of petr4, petrobras, brazil. *Expert Systems with Applications*, 40(18), 7596–7606.

Di Persio, L. and Honchar, O. (2016). Artificial neural networks architectures for stock price prediction: Comparisons and applications. *International journal of circuits, systems and signal processing*, 10, 403–413.

Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.

Gholamian, E. and Davoodi, S.M.R. (2018). Predicting the direction of stock market prices using random forest.

Graves, A., Mohamed, A.r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, 6645–6649. IEEE.

Gunduz, H., Yaslan, Y., and Cataltepe, Z. (2017). Intraday prediction of borsa istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems*, 137, 138–148.

Hagenau, M., Liebmann, M., and Neumann, D. (2013). Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems*, 55(3), 685–697.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Hoseinzade, E. and Haratizadeh, S. (2019). Cnnpred: Cnn-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129, 273–285.

Hu, P., Pan, J.S., Chu, S.C., Chai, Q.W., Liu, T., and Li, Z.C. (2019). New hybrid algorithms for prediction of daily load of power network. *Applied Sciences*, 9(21), 4514.

Kim, K.j. and Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with Applications*, 19(2), 125–132.

Kuo, S.Y., Kuo, C., and Chou, Y.H. (2013). Dynamic stock trading system based on quantum-inspired tabu search algorithm. In *2013 IEEE Congress on Evolutionary Computation*, 1029–1036. IEEE.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Long, W., Lu, Z., and Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, 164, 163–173.

Nelson, D.M., Pereira, A.C., and de Oliveira, R.A. (2017). Stock market's price movement prediction with lstm neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 1419–1426. IEEE.

Pan, J.S., Hu, P., and Chu, S.C. (2019). Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power. *Processes*, 7(11), 845.

Pang, X., Zhou, Y., Wang, P., Lin, W., and Chang, V. (2018). An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, 1–21.

Rani, S. and Sikka, G. (2012). Recent techniques of clustering of time series data: a survey. *International Journal of Computer Applications*, 52(15).

Siripurapu, A. (2014). Convolutional networks for stock trading. *Stanford Univ Dep Comput Sci*.

Taylor, M.P. and Allen, H. (1992). The use of technical analysis in the foreign exchange market. *Journal of international Money and Finance*, 11(3), 304–314.

Williams, R.J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2), 270–280.

Zhong, X. and Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, 67, 126–139.