

Received May 29, 2018, accepted July 3, 2018, date of publication July 25, 2018, date of current version September 21, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2859809

# Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?

LIN CHEN<sup>1</sup>, ZHILIN QIAO<sup>2</sup>, MINGGANG WANG<sup>3</sup>, CHAO WANG<sup>4</sup>,  
RUIJIN DU<sup>5</sup>, AND HARRY EUGENE STANLEY<sup>6</sup>

<sup>1</sup>School of Management, Northwestern Polytechnical University, Xi'an 710072, China

<sup>2</sup>School of Economics and Finance, Xi'an Jiaotong University, Xi'an 710061, China

<sup>3</sup>School of Mathematical Science, Nanjing Normal University, Nanjing 210042, China

<sup>4</sup>College of Economics and Management, Beijing University of Technology, Beijing 100124, China

<sup>5</sup>Energy Development and Environmental Protection Strategy Research Center, Jiangsu University, Zhenjiang 212013, China

<sup>6</sup>Center for Polymer Studies and Department of Physics, Boston University, Boston, MA 02215, USA

Corresponding author: Zhilin Qiao (clinne@126.com)

This work was supported by the National Social Science Fund of China under Grant 17BGL143.

**ABSTRACT** Unpredictable stock market factors make it difficult to predict stock index futures. Although efforts to develop an effective prediction method have a long history, recent developments in artificial intelligence and the use of artificial neural networks have increased our success in nonlinear approximation. When we study financial markets, we can now extract features from a big data environment without prior predictive information. We here propose to further improve this predictive performance using a combination of a deep-learning-based stock index futures prediction model, an autoencoder, and a restricted Boltzmann machine. We use high-frequency data to examine the predictive performance of deep learning, and we compare three traditional artificial neural networks: 1) the back propagation neural network; 2) the extreme learning machine; and 3) the radial basis function neural network. We use all of the 1-min high-frequency transaction data of the CSI 300 futures contract (IF1704) in our empirical analysis, and we test three groups of different volume samples to validate our observations. We find that the deep learning method of predicting stock index futures outperforms the back propagation, the extreme learning machine, and the radial basis function neural network in its fitting degree and directional predictive accuracy. We also find that increasing the amount of data increases predictive performance. This indicates that deep learning captures the nonlinear features of transaction data and can serve as a powerful stock index futures prediction tool for financial market investors.

**INDEX TERMS** Prediction methods, artificial neural networks, stock markets, deep learning.

## I. INTRODUCTION

Stock market prediction is a classic topic in both financial circles and academia. Extreme stock market fluctuations, e.g., the global stock market turmoil in February 2018, damage financial markets and the global economy. We thus need a more effective way of predicting market fluctuations. Among the many predictive efforts over the last few decades [1]–[3], some have had success using quantitative methods [4]–[9], such as autoregressive integrated moving average (ARIMA) models, artificial neural networks, support vector machine, and neuro-fuzzy based systems, but because of the nonlinear characteristics of stock market behavior, financial economists continue to debate these methodologies [10]–[13].

Recently “deep learning” (DL) has attracted a great deal of attention in many research fields. DL is a new area of machine learning that has improved the ability of computers

in areas of image recognition and classification, natural language processing, speech recognition, and social network filtering [14]–[16]. In some cases the results are comparable to or even superior to those of human experts [17], [18]. The structure of DL is a multi-layer neural network that uses a cascade of multiple layers of nonlinear processing units to extract and transform various features. Its learning can be either supervised or unsupervised, and it forms a hierarchy of concepts by utilizing multiple representation levels that correspond to different levels of abstraction [19]. The use of DL has improved computational power and big data processing, and it allows more sophisticated algorithms. Previous studies indicate that DL more efficiently solves nonlinear problems than traditional methods [19]–[21]. Irrespective of the level of complication or the presence of linear and nonlinear big data financial market factors, DL can extract abstract features

and identify hidden relationships in financial markets without making econometric assumptions [22]. Traditional financial economic methods and other quantitative techniques cannot do this. Over against the limitations of existing models DL can process high frequency big data, analyze the financial market, and predict stock returns [23], [24]. When setting the parameters of artificial neural networks, the learning rate, epochs, goals, and number of artificial neurons, all must be taken into account to achieve desirable results [25], [26]. DL can extract abstract features without subjective interferences. We thus do not need to add influencing factors or control variables when we use a large time-series dataset to predict financial market behavior. Some other methods have been applied to time-series analysis [27]–[29]. For example, Gao *et al.* [28] propose a new wavelet multiresolution complex network for analyzing multivariate time series, which is capable of analyzing the information in the dynamical and topological fields and is successfully applied in many research fields. But DL can provide accurate financial market time-series forecasting, and the accuracy increases as the size of the database increases [30]. Thus DL is perfect for time-series prediction in the financial market.

We here use deep learning to predict stock market behavior, and we compare the performance of this approach with the performance of traditional back propagation (BP) network, extreme learning machines (ELM), radial basis function (RBF). Many studies have used traditional predictive methods and low frequency data to predict stock prices and returns, but the market inefficiencies caused by high frequency microstructure noise may provide additional profit opportunities. When traditional predictive methods are used to examine high frequency data they are also subject to overfitting and low accuracy.

We thus use a DL-based prediction model and high frequency time-series data from 20 February 2017 to 20 April 2017 in the Chinese stock market to predict stock index futures and compare the predictive performance with that from other methods. In Sec. 2 we describe deep learning and other artificial intelligence algorithms. In Sec. 3 we describe the data source and the data characteristics and propose the criteria for evaluating the performance of our model. Sec. 4 provides our empirical results, and Sec. 5 provides the conclusions.

## II. DEEP LEARNING AND OTHER ARTIFICIAL INTELLIGENCE ALGORITHMS

### A. DEEP LEARNING

Deep learning is a machine learning technology that bypasses manual extraction and extracts features mechanistically. Deep learning simulates the cerebral cortex and abstract data or signals, layer by layer, and models the image recognition of the cerebral cortex. Deep learning first extracts low-level features from original signals. It then extracts high-level features, first from low-level features and then from higher-level features. In an image recognition system,

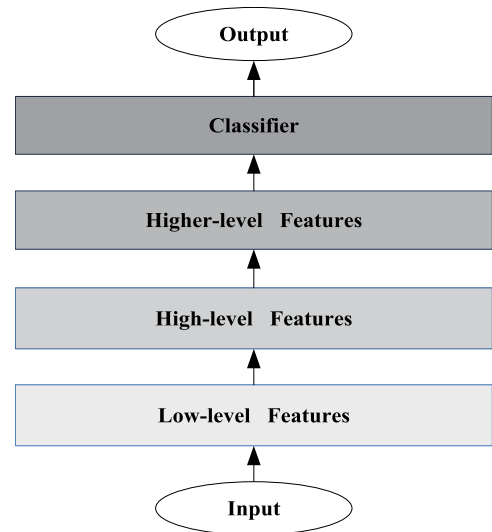


FIGURE 1. Illustration of deep learning hierarchy.

the original signals are pixels. The low-level features are the edges of objects. The high-level features are the contours, and the highest-level feature is the image. Using the characteristics of high-level classification, deep learning outputs forecast results [15]. Figure 1 shows the deep learning hierarchy.

Deep learning abstracts and transfers features of data through different layers. Here  $f$  is the activation function. Given input  $X$  and predictable output  $\hat{Y}$ , the prediction function of deep learning is

$$\begin{aligned}
 M_1 &= f_1(W_0X + b_0), \\
 M_2 &= f_2(W_1M_1 + b_1), \\
 &\dots \\
 M_L &= f_L(W_{L-1}M_{L-1} + b_{L-1}), \\
 \hat{Y}(X) &= W_L M_L + b_L.
 \end{aligned}
 \tag{1}$$

Here  $W$  indicates the weight matrices,  $b$  the biases, and  $L$  the number of layers. The training process of deep learning has two stages. The first is unsupervised learning from bottom to top. The second is supervised learning from top to bottom. The widely used data representation methods in deep learning are two nonlinear transformations, (i) the autoencoder (AE) and (ii) the restricted Boltzmann machine (RBM).

#### 1) AUTOENCODER

The autoencoder learns a representation (an encoding) for a dataset, typically in order to reduce the dimensionality. It compresses the input into a latent-space representation and then reconstructs the output from this representation (see Fig.2).

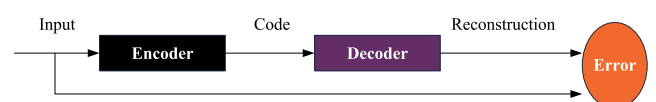


FIGURE 2. Principle of autoencoder.

An autoencoder consists of an encoder and a decoder [14], defined as transitions  $\phi$  and  $\varphi$ ,

$$\begin{aligned}\phi &: \chi \rightarrow \kappa \\ \varphi &: \kappa \rightarrow \chi \\ \phi, \varphi &= \arg \min_{\phi, \varphi} \|X - (\varphi \circ \phi)X\|^2.\end{aligned}\quad (2)$$

In the encoder stage,  $x_l \in \mathbb{R}^p = \chi$  is the input of layer  $l$  mapped to  $y_l \in \mathbb{R}^q = \kappa$  with the function  $y_l = f_l(W_l x_l + b_l)$ . A sigmoid function is typically used to activate function  $f_l$ , which we also use here. In the decoder stage,  $y_l$  is mapped to the reproduction of the same shape as  $x_l$  using the function  $x'_l = f'_l(W'_l y_l + b'_l)$ . Autoencoders are trained to minimize reconstruction errors,

$$E(X, X') = \|X - X'\|^2 = \|X - f'(W'(f(WX + b)) + b')\|^2.\quad (3)$$

We here use autoencoders to set initial weights and thresholds to reduce the error and more rapidly reach the desired range.

## 2) RESTRICTED BOLTZMANN MACHINE

The restricted Boltzmann machine (RBM) is a generative stochastic artificial neural network that can learn a probability distribution. It is a restricted variant of a Boltzmann machine in which neurons must form a bipartite graph. Depending on the task, the training of an RBM can be either supervised or unsupervised. A standard RBM has a hidden binary-value and visible units. Here  $X$ ,  $Y$  are the visible (input) and hidden layers, respectively. The energy of a configuration (a pair of boolean vectors)  $(X, Y)$  is defined

$$G(X, Y) = -\alpha^T X - \beta^T Y - X^T W Y, \quad (4)$$

where the parameter  $\alpha$ ,  $\beta$ , and  $W$  are the weights of visible units, hidden units, and those associated with the connection between hidden and visible units, respectively. The probability distributions over hidden or visible vectors in general Boltzmann machines are defined in terms of the energy function [31]:

$$P(X, Y) = \frac{1}{Z} e^{-G(X, Y)}, \quad (5)$$

where  $Z$  is a partition function defined as the sum of  $e^{-G(X, Y)}$  over all possible configurations, and the marginal probability of a visible vector is the sum over all possible hidden layer configurations,

$$P(X) = \frac{1}{Z} \sum_Y e^{-G(X, Y)}. \quad (6)$$

With the restriction that RBM has the shape of a bipartite graph with no intra-layer connections, the hidden unit activations are mutually independent given the visible unit activations, and vice versa. In this paper, we use RBM to pre-train the network layer by layer, and then fine-tune it with feedback method.

## B. ARTIFICIAL INTELLIGENCE ALGORITHMS

Artificial intelligence algorithms have been used to predict economic and market behavior [32]–[34]. To compare predictive performances, we focus on three popular artificial neural networks, (i) the back propagation neural network, (ii) the extreme learning machine, and (iii) the radial basis function neural network.

### 1) BACK PROPAGATION NEURAL NETWORK

The back propagation (BP) neural network is an artificial intelligence algorithm widely used in prediction, in particular for advanced multiple regression analysis. It better generates complex and non-linear responses than a standard regression analysis [35]. The formula for the BP algorithm is

$$W(n) = W(n-1) - \Delta W(n), \quad (7)$$

where

$$\Delta W(n) = \gamma \frac{\partial E}{\partial W}(n-1) + \theta \Delta W(n-1). \quad (8)$$

Here  $\gamma$  is the learning rate,  $E$  the gradient of error function, and  $\theta \Delta W(n-1)$  the quantity of incremental weight. A BP network uses the gradient method, and the learning and inertial factors are determined by experience. This affects the convergence in a BP network. A BP network rapidly converges to a local minimum, but because it learns the convergent velocity more slowly, it has relatively few applications.

### 2) EXTREME LEARNING MACHINE

The extreme learning machine (ELM) is a feedforward neural network for classification, regression, clustering, and feature learning with a single layer or multi layers of hidden nodes in which the parameters of hidden nodes need not be tuned. For a set of training samples  $\{(X_j, Z_j)\}_{j=1}^S$  with  $S$  samples and  $V$  classes, the single hidden layer feedforward neural network with  $d$  hidden nodes and activation function  $f$  is

$$Y_j = \sum_{i=1}^d \eta_i f_i(X_j) = \sum_{i=1}^d \eta_i f(W_i X_j + e_i), \quad j = 1, 2, \dots, S, \quad (9)$$

where  $X_j = [x_{j1}, x_{j2}, \dots, x_{js}]^T$  is the input,  $Z_j = [z_{j1}, z_{j2}, \dots, z_{jv}]^T$  the corresponding output,  $W_i = [w_{i1}, w_{i2}, \dots, w_{is}]^T$  the connecting weights of hidden neuron  $i$  to input neurons,  $e_i$  the bias of hidden node  $i$ ,  $\eta_i = [\eta_{i1}, \eta_{i2}, \dots, \eta_{iv}]^T$  the connecting weights of hidden neuron  $i$  to the output neurons, and  $Y_j$  the actual network output. Usually the hidden parameters  $\{W_i, e_i\}$  are randomly generated during training without tuning. The ELM solves the compact model using error minimization

$$\min_{\eta} \|D\eta - Z\|_F \quad (10)$$

with

$$D(W_1, W_2, \dots, W_d, e_1, e_2, \dots, e_d) = \begin{bmatrix} f(W_1X_1 + e_1) & \dots & f(W_hX_1 + e_d) \\ \vdots & \dots & \vdots \\ f(W_1X_s + e_1) & \dots & f(W_hX_s + e_d) \end{bmatrix},$$

$$\eta = \begin{bmatrix} \eta_1^T \\ \vdots \\ \eta_d^T \end{bmatrix}, \quad Z = \begin{bmatrix} Z_1^T \\ \vdots \\ Z_s^T \end{bmatrix},$$

where  $D$  is the hidden layer output matrix, and  $\eta$  the output weight matrix. ELM randomly selects the hidden node parameters, and only the output weight parameters needed to be determined. Despite these advantages, ELM cannot be used when the time series is noisy.

### 3) RADIAL BASIS FUNCTION NEURAL NETWORK

The radial basis function (RBF) network is an artificial neural network that uses radial basis functions for activation. The network output is a linear radial basis function combination of inputs and neuron parameters. RBF networks have many uses, including function approximation, time series prediction, classification, and system control [36]. Radial basis functions are often used to construct function approximations,

$$y(x) = \sum_{i=1}^N \mu_i v(\|x - x_i\|), \quad (11)$$

where  $y(x)$  is the sum of  $N$  radial basis functions, and  $\mu_i$  the weight estimated using the linear least square matrix method. The Gaussian function is a commonly used radial basis function, and is here used by  $v(\|x - x_i\|)$ . Geometrically the RBF network divides the input space into hypersphere subspaces, which can cause such algorithmic problems as overfitting, overtraining, the small-sample effect, and singularities.

## III. DATA AND PERFORMANCE EVALUATION CRITERIA

### A. DATA

China Securities Index Co., Ltd, a joint venture company of Shanghai and Shenzhen Stock Exchanges, received permission from the China Securities Regulatory Commission to release the China securities indices. The CSI 300 is a representative index of these indices. It is a capitalization-weighted stock market index designed to represent the performance of the top 300 stocks listed in the Shanghai and Shenzhen stock exchanges. As an early stock index futures contract issued in China, the CSI 300 futures contract has enriched the existing index system in China's securities markets and achieved great success. The CSI 300 index futures began trading on China Financial Futures Exchange (CFFEX) with the commodity ticker symbol IF on 16 April 2010. The notional value of one contract is RMB 300 times the value of the CSI 300. The CSI 300 index futures provides an indicator when observing the fluctuation of the stock market. It facilitates companies and individual investors to better understand stock market

developments and allows the accumulation of experience when trading index investment products. Thus we here select the CSI 300 futures contract transaction data for empirical analysis.

The CSI 300 index futures uses the T+0 trading system for short-term transactions. They list the price fluctuations and performance of the Chinese A-share market, which serve as benchmarks for derivative innovations and indexing. Because our goal here is to predict the short-term price of stock index futures and to provide guidance in transactions, the CSI 300 index futures is the appropriate one for analysis. We collect all transaction data of the CSI 300 futures contract (IF1704) from 20 February 2017 to 20 April 2017, which we acquire from the high frequency RESSET database, a top Chinese financial research data resource. We use one-minute high frequency data that includes opening price, highest price, lowest price, closing price, trade volume, and opening interest to test the performance of deep learning and other artificial intelligence algorithms. We collect 10,000 6-variable groups from 20 February 2017 to 20 April 2017 for a total of 60,000 data.

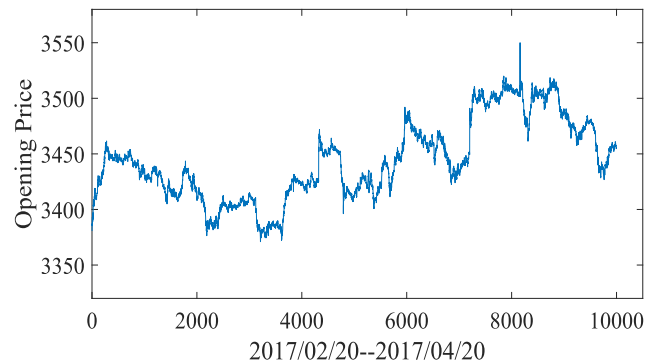


FIGURE 3. Opening price per minute of the CSI 300 futures contract (IF1704).

Figure 3 shows the minute-by-minute trend of the opening price of the CSI 300 futures contract (IF1704) from 20 February 2017 to 20 April 2017. Note that there is a big fluctuation during this period, and that thus there may be arbitrage opportunities if this indication is accurate.

When using the deep learning algorithm the preparatory data processing includes decreasing the difference between the threshold and the actual data. Usually the sample data are normalized prior to their being input into the neural network,

$$x'_t = \frac{x_t - x_{min}}{x_{max} - x_{min}}, \quad (12)$$

where  $x'_t$  is the data after normalization, and  $x_{min}$  and  $x_{max}$  the minimum and maximum data of input  $x_t$ , respectively. After processing we anti-normalize the output,

$$\hat{y}_t = y'_t(y_{max} + y_{min}) + y_{min}, \quad (13)$$

where  $\hat{y}_t$  is the predicted data after anti-normalization, and  $y_{min}$  and  $y_{max}$  the minimum and maximum data, respectively, of output  $y'_t$ .



**B. PERFORMANCE EVALUATION**

We use three criteria to evaluate predictive accuracy, (i) the root-mean-square error (RMSE), (ii) the mean absolute percentage error (MAPE), and (iii) the directional predictive accuracy. The RMSE is defined

$$RMSE = \sqrt{\frac{\sum_{t=1}^N (\hat{y}_t - y_t)^2}{N}}, \tag{14}$$

where  $y_t$  and  $\hat{y}_t$  are actual value and predicted value at time  $t$ , respectively, and  $N$  the data size of the tested set. RMSE expresses the standard deviation of the difference between the predicted and actual values.

The MAPE, also known as the mean absolute percentage deviation (MAPD), expresses the accuracy as a percentage, and is defined

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{y_t - \hat{y}_t}{y_t} \right|. \tag{15}$$

MAPE measures the mean absolute relative error of each prediction model. RMSE and MAPE have been widely used to evaluate the predictive accuracy [37], [38]. The smaller the values of RMSE and MAPE, the higher the accuracy of the model.

Furthermore, we are also interested in tendencies in predictive accuracy, we measure the directional predictive accuracy,

$$DA = \frac{1}{N} \sum_{t=1}^N D_t, D_t = \begin{cases} 1, & (y_{t+1} - y_t)(\hat{y}_{t+1} - \hat{y}_t) \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The closer  $DA$  is to 1, the higher the directional predictive accuracy. The closer  $DA$  is to 0, the lower the directional predictive accuracy.

**IV. RESULTS OF EMPIRICAL ANALYSIS**

To eliminate random influences within each simulation we use the average value of 10 simulation results to evaluate the performance of each method. To determine the performance of each neural networks when there are different amounts of data, we divide the database into small, medium and large datasets (see Fig.4). In each small dataset, there are 2000 groups with six variables in each group, yielding a total of 12000 data. There are five small-scale datasets. In each medium dataset, there are 5000 groups with six variables in each group, yielding 30000 data in each medium dataset. There are two medium-scale datasets. Similarly, in each large dataset, there are 10000 groups with 60000 data. We use the first 90 percent of data in each dataset as training samples and the remaining 10 percent as testing samples. We run all neural networks using the Matlab R2017b software package and a Lenovo laptop computer with a Core(TM) i5-5200U 2.20GHz CPU and 8GB of random-access memory (RAM).

In DL the number of hidden layer nodes strongly affects the prediction. There is no uniform way of determining hidden layer nodes [39], and the experience of model creator is key

**A small-scale dataset**

	A	B	C	D	E	F	G	H
1	Date	Time	Opening Price	Highest Price	Lowest Price	Closing Price	Trade Volume	Opening Interest
2	2017 02 20	09 31	3380.6	3383.6	3380.6	3383.6	3	3
3	2017 02 20	09 32	3387.4	3387.4	3387.4	3387.4	1	4
4	2017 02 20	09 33	3387.4	3387.4	3385	3385.4	5	9
5	2017 02 20	09 34	3385.4	3385.4	3385.4	3385.4	4	13
6	2017 02 20	09 35	3388	3388	3386	3386.8	4	17

**A medium-scale dataset**

	A	B	C	D	E	F	G	H
1	Date	Time	Opening Price	Highest Price	Lowest Price	Closing Price	Trade Volume	Opening Interest
2	2017 02 20	09 31	3380.6	3383.6	3380.6	3383.6	3	3
3	2017 02 20	09 32	3387.4	3387.4	3387.4	3387.4	1	4
4	2017 02 20	09 33	3387.4	3387.4	3385	3385.4	5	9
5	2017 02 20	09 34	3385.4	3385.4	3385.4	3385.4	4	13
6	2017 02 20	09 35	3388	3388	3386	3386.8	4	17

**A large-scale dataset**

	A	B	C	D	E	F	G	H
1	Date	Time	Opening Price	Highest Price	Lowest Price	Closing Price	Trade Volume	Opening Interest
2	2017 02 20	09 31	3380.6	3383.6	3380.6	3383.6	3	3
3	2017 02 20	09 32	3387.4	3387.4	3387.4	3387.4	1	4
4	2017 02 20	09 33	3387.4	3387.4	3385	3385.4	5	9
5	2017 02 20	09 34	3385.4	3385.4	3385.4	3385.4	4	13
6	2017 02 20	09 35	3388	3388	3386	3386.8	4	17

**FIGURE 4. Demonstration of different scale datasets.**

during the layer-by-layer experiment. The number of hidden layer nodes strongly correlates with the input and output layer nodes. In general, the greater the number of nodes in the input and output layers, the greater the number of nodes needed in the hidden layers to achieve efficient feature learning. Currently, the following methods can be used to determine the initial number of hidden layer nodes,

$$L = \sqrt{m + n} + \alpha, \tag{17}$$

$$L = \log_2 m, \tag{18}$$

and

$$L = \sqrt{mn}. \tag{19}$$

Here  $m$  is the number of input layer nodes,  $n$  the number of output layers nodes, and  $\alpha$  a constant between 1-10. The number of nodes in the hidden layer produced using these methods is only approximate, and often must be corrected during training and learning. Gradually increasing and decreasing the number of hidden layer nodes is a common method of reducing errors to a usable range. We use Eq. (17) to calculate the number of hidden layer nodes and find 10 nodes in the first hidden layer and 4 nodes in the second hidden layer.

**A. PREDICTIVE PERFORMANCE OF SMALL-SCALE DATASETS**

Each small dataset contains 12000 data. According to the total sample volume, 5 small-scale datasets are obtained for analysis. The first 10800 data (90%) are training data and the remaining 1200 (10%) test data. Figure 5 shows the testing sample values and the CSI 300 futures contract predictions for different periods. Table 1 compares the predictive performances of the DL, BP, ELM and RBF methods. In each dataset we record the average value of 10 simulations. The last line in each period gives the running time for each method, and the last row of Table 1 shows the mean value of each method.

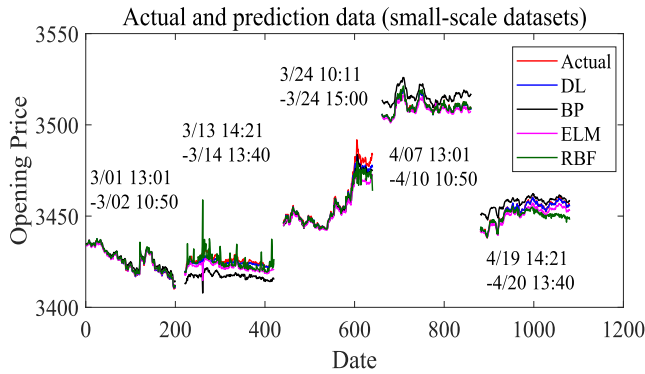


FIGURE 5. Actual and prediction data of the CSI 300 futures contract (small-scale datasets).

TABLE 1. The error and running time of prediction (small-scale datasets).

Testing Period	Criteria	DL	BP	ELM	RBF
3/01 13:01- 3/02 10:50	RMSE	0.4167	1.6583	0.8595	0.7208
	MAPE	0.0001	0.0004	0.0002	0.0002
	DA	0.7150	0.4800	0.5850	0.6750
	Time(s)	10.0472	5.3171	0.3734	18.6100
3/13 14:21- 3/14 13:40	RMSE	1.6441	8.5737	3.7178	3.7802
	MAPE	0.0004	0.0025	0.0010	0.0008
	DA	0.4850	0.4450	0.4500	0.4600
	Time(s)	14.4155	1.7276	0.1273	19.0071
3/24 10:11- 3/24 15:00	RMSE	2.1212	3.4200	5.4519	4.2656
	MAPE	0.0004	0.0007	0.0011	0.0006
	DA	0.7650	0.5050	0.4950	0.7350
	Time(s)	19.1702	1.6232	0.1061	16.1977
4/07 13:01- 4/10 10:50	RMSE	0.5213	6.0358	1.8921	0.7040
	MAPE	0.0001	0.0017	0.0005	0.0001
	DA	0.8200	0.4200	0.5300	0.7800
	Time(s)	24.4831	1.6459	0.0907	18.1132
4/19 14:21- 4/20 13:40	RMSE	0.5151	4.9356	2.5872	4.4237
	MAPE	0.0001	0.0012	0.0007	0.0010
	DA	0.7900	0.4750	0.4900	0.5250
	Time(s)	29.9869	1.7319	0.0927	16.5090
Mean	RMSE	1.0437	4.9247	2.9017	2.7788
	MAPE	0.0002	0.0013	0.0007	0.0005
	DA	0.7150	0.4650	0.5100	0.6350
	Time(s)	19.6206	2.4091	0.1581	17.6874

Table 1 shows that the DL prediction of the opening price per minute of the CSI 300 futures contract outperforms the BP, ELM, RBF methods in both accuracy and direction. The mean RMSE value of DL is 1.0437, lower than the mean RMSE values of BP (4.9247), ELM (2.9017), and RBF (2.7788), and the mean MAPE value of DL (0.0002) is also lower than those of BP (0.0013), ELM (0.0007), and RBF (0.0005). Thus the DL method has the lowest error when predicting the price of the CSI 300 index futures. The DL directional prediction of 0.7150 is also more accurate than the mean DA values of BP (0.4650), ELM (0.5100), and RBF (0.6350). The running time for small datasets does not

indicate that DL is superior to the other methods. DL usually has more than one hidden layer, thus it requires more time to train the network. These results indicate that the predictive performance of DL is better than that of BP, ELM, and RBF, implying that DL may be a useful tool when forecasting stock market behavior.

**B. PREDICTIVE PERFORMANCE OF MEDIUM-SCALE DATASETS**

When a dataset contains 30000 data, we divide it into two medium datasets. We use the first 27000 data (90%) to train the network and the remaining 3000 (10%) to test the predictive performance of each method. Figure 6 shows the sample values and forecasts of the CSI 300 futures contract during two periods. Table 2 compares the predictive performances of the DL, BP, ELM, and RBF methods. In the last row are the average values of 10 simulation results for each dataset and a summary of the mean value for each method.

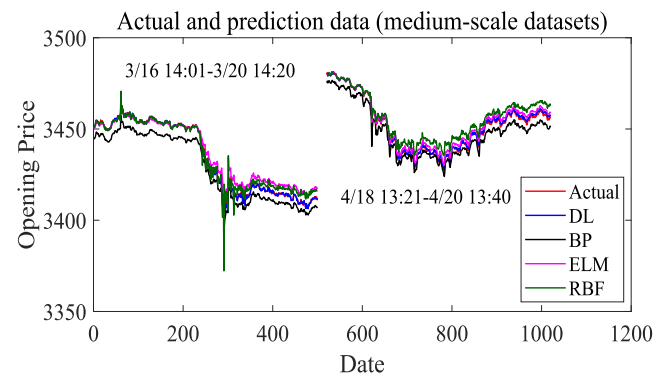


FIGURE 6. Actual and prediction data of the CSI 300 futures contract (medium-scale datasets).

TABLE 2. The error and running time of prediction (medium-scale datasets).

Testing Period	Criteria	DL	BP	ELM	RBF
3/16 14:01- 3/20 14:20	RMSE	0.7324	5.9677	4.2976	2.9929
	MAPE	0.0001	0.0017	0.0010	0.0006
	DA	0.7440	0.4780	0.4700	0.6040
	Time(s)	13.0792	4.8034	0.3846	229.3378
4/18 13:21- 4/20 13:40	RMSE	0.8531	4.2729	2.1513	4.8682
	MAPE	0.0002	0.0012	0.0005	0.0012
	DA	0.7180	0.4700	0.5420	0.5400
	Time(s)	9.4341	1.5968	0.1121	208.4432
Mean	RMSE	0.7928	5.1203	3.2245	3.9305
	MAPE	0.0002	0.0014	0.0007	0.0009
	DA	0.7310	0.4740	0.5060	0.5720
	Time(s)	11.2566	3.2001	0.2482	218.8905

Table 2 shows that the DL predictions are superior to those of BP, ELM, and RBF in both accuracy and direction. The DL has a mean RMSE value of 0.7928, much lower than the mean RMSE values of BP (5.1203), ELM (3.2245), and RBF (3.9305). The mean MAPE value of DL is 0.0002,

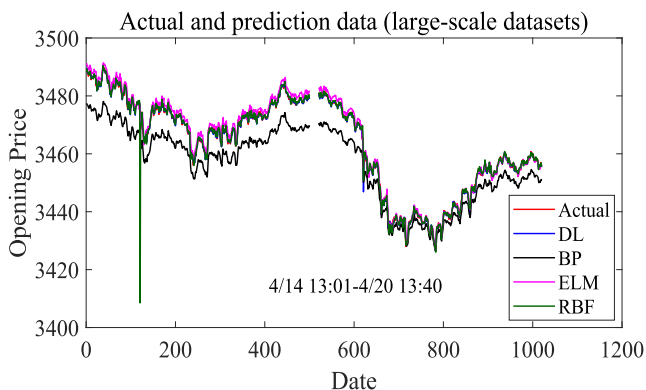
**TABLE 3. The error and running time of prediction (large-scale datasets).**

Testing Period	Criteria	DL	BP	ELM	RBF
4/14 13:01-	RMSE	0.6423	7.6147	1.7071	2.1135
4/20 13:40	MAPE	0.0001	0.0020	0.0004	0.0002
	DA	0.8120	0.4790	0.5460	0.7870
	Time(s)	26.8537	5.5083	0.4637	3681.4000

also lower than those of the BP (0.0014), ELM (0.0007), and RBF (0.0009) methods, indicating that the error of DL when predicting the price of the CSI 300 index futures is minimal. The mean DA value of DL is 0.7310 for directional prediction, much higher and more accurate than those of BP (0.4740), ELM (0.5060), and RBF (0.5720). When increasing the size of the sample data, DL also shows better performance. In contrast to the average RBF running time of 218.8905, DL requires on average 11.2566 seconds to complete the training and testing with a usable predictive accuracy. Thus the high predictive performance of DL is also confirmed in medium datasets.

**C. PREDICTIVE PERFORMANCE OF LARGE-SCALE DATASETS**

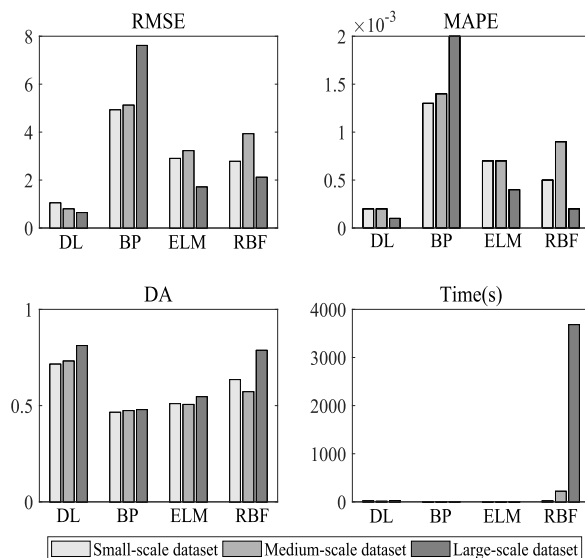
Our large-scale dataset contains 60000 data. We use the first 54000 data (90%) to train the network and the remaining 6000 (10%) to test the predictive accuracy of the methods. We have one large-scale dataset in the current database. Figure 7 shows the sample values and the forecasts of the CSI 300 futures contract in the predictive period. Table 3 shows the predictive performances of the DL, BP, ELM, and RBF methods. The average value of 10 simulation results is also provided.



**FIGURE 7. Actual and prediction data of the CSI 300 futures contract (large-scale datasets).**

Table 3 shows that the DL predictions of value and direction are more accurate than those of the BP, ELM, and RBF methods. The RMSE value of DL (0.6423) is lower than those of BP (7.6147), ELM (1.7071), and RBF (2.1135). The MAPE value of DL (0.0001) is also lower than those of BP (0.0020), ELM (0.0004), and RBF (0.0002). The error is smallest when using DL to predict the price of the CSI 300 index futures. The DA value of DL (0.8120) is also higher

than those of BP (0.4790), ELM (0.5460), and RBF (0.7870), indicating that the deep learning method has the highest directional predictive accuracy. The running time of DL is also superior to that of RBF. Although the DA value of RBF reaches 0.7870, it requires 3681.4000 seconds to process this scale of data. In contrast, DL requires only 26.8537 seconds to process the same data, and the accuracy of its prediction is higher. Thus the predictive performance of DL is better than the others for large-scale dataset.



**FIGURE 8. Predictive performance comparison of different scale datasets.**

Figure 8 shows the empirical results for different size datasets and compares their average predictive performances. DL exhibits the minimum RMSE and MAPE errors for small, medium, and large datasets and its directional predictions are the most accurate. The predictive accuracy of DL increases as the sample volume increases. Although the DL running time consumed is more than that of BP and ELM, it is much less than that of RBF, and although RBF is more accurate, it requires so much more running time and this advantage is obviated.

An empirical analysis shows that DL can be trained to a given accuracy, and that it fits actual data well. We construct a non-linear map of the transaction data to the closing price of the following minute. The error reaches an ideal value after a few iterations and satisfies the experimental requirements. DL also has the highest fitting degree of directional prediction, and its accuracy increases when the size of the sample data increase. The fitting degrees of BP and ELM are less accurate, and the higher accuracy RBF is undone by its excessively long running time.

We follow the study of Chong et al., add trade volume to the input, and test high frequency (one-minute) stock market data. We find that DL is more effective when applied to high frequency data, consistent with the assumption of Chong et al. [22]. Thus DL is an effective method, particularly for large high frequency data. It improves predictive

accuracy, reduces investment blindness, and lowers investment risk.

## V. CONCLUSION

Predicting stock market behavior is challenging. Nonlinear relationship among transaction data and unpredictable factors in market fluctuations make predictions difficult. Deep learning is a machine learning method suitable for solving nonlinear approximations that has been successfully applied in many fields.

To achieve better stock market predictive performance, we have used a deep architecture-based model. We use one-minute high frequency transaction data from the CSI 300 futures contract (IF1704) in the Chinese stock market and carry out an empirical analysis. To show the effect of sample volume on network training and predicting, we divide the sample into three scale datasets and compare the stock price prediction of deep learning with three traditional artificial neural networks (BP, ELM, RBF). We compare their predictive fitting degree and directional predictive accuracy and find that the predictive performance of deep learning is superior to that of BP, ELM and RBF. We also find that increasing sample volume significantly increases the predictive performance of deep learning. We thus find that sample volume strongly affects stock prediction, and that deep learning performs well when applied to large data. Furthermore, deep learning does not need prior predictive information to extract features from large datasets, and this increases its usefulness in predicting stock market behavior. This result provides additional evidence that DL is an effective method of predicting stock price. Our deep learning prediction model expands our ability to analyze financial market behavior. Because there are complex relationships among stock futures prices and such factors as the economy, politics, the environment, and culture, future research could apply complex network theory to key input variables that influence stock prices and returns. That would allow the construction of a deep learning network that would facilitate better predictive performance.

## REFERENCES

- [1] M. H. Pesaran and A. Timmermann, "Predictability of stock returns: Robustness and economic significance," *J. Finance*, vol. 50, no. 4, pp. 1201–1228, Sep. 1995.
- [2] A. Ang and G. Bekaert, "Stock return predictability: Is it there?" *Rev. Financial Stud.*, vol. 20, no. 3, pp. 651–707, Jul. 2006.
- [3] D. E. Rapach, J. K. Strauss, and G. Zhou, "International stock return predictability: What is the role of the United States?" *J. Finance*, vol. 68, no. 4, pp. 1633–1662, Aug. 2013.
- [4] M. Qi, "Nonlinear predictability of stock returns using financial and economic variables," *J. Bus. Econ. Stat.*, vol. 17, no. 4, pp. 419–429, Oct. 1999.
- [5] M. Aiolfi and C. A. Favero, "Model uncertainty, thick modelling and the predictability of stock returns," *J. Forecasting*, vol. 24, no. 4, pp. 233–254, Jul. 2005.
- [6] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.
- [7] G. S. Atsalakis and K. P. Valavanis, "Forecasting stock market short-term trends using a neuro-fuzzy based methodology," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10696–10707, Sep. 2009.
- [8] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFin text system," *ACM Trans. Inf. Syst.*, vol. 27, no. 2, p. 12, Feb. 2009.
- [9] A. A. Adebisi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," *J. Appl. Math.*, vol. 2014, Mar. 2014, Art. no. 614342.
- [10] M. J. Hinich and D. M. Patterson, "Evidence of nonlinearity in daily stock returns," *J. Bus. Econ. Stat.*, vol. 3, no. 1, pp. 69–77, Jan. 1985.
- [11] D. G. McMillan, "Nonlinear predictability of stock market returns: Evidence from nonparametric and threshold models," *Int. Rev. Econ. Finance*, vol. 10, no. 4, pp. 353–368, Dec. 2001.
- [12] A. Kanas, "Nonlinearity in the stock price–dividend relation," *J. Int. Money Finance*, vol. 24, no. 4, pp. 583–606, Jun. 2005.
- [13] C. A. Bonilla, R. Romero-Meza, and M. J. Hinich, "Episodic nonlinearity in Latin American stock market indices," *Appl. Econ. Lett.*, vol. 13, no. 3, pp. 195–199, Feb. 2006.
- [14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
- [15] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
- [16] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [17] C. Dan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. CVPR*, 2012, pp. 3642–3649.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [19] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, Jun. 2014.
- [20] X.-W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.
- [21] X. Jin, J. Shao, X. Zhang, W. An, and R. Malekian, "Modeling of nonlinear system based on deep learning framework," *Nonlinear Dyn.*, vol. 84, no. 3, pp. 1327–1340, May 2016.
- [22] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Syst. Appl.*, vol. 83, pp. 187–205, Oct. 2017.
- [23] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proc. IJCAI*, 2015, pp. 2327–2333.
- [24] J. B. Heaton, N. Polson, and J. Witte, "Deep learning for finance: Deep portfolios," *Appl. Stochastic Models Bus. Ind.*, vol. 33, no. 1, pp. 3–12, Jan. 2017.
- [25] Z. Gao, W. Dang, C. Mu, Y. Yang, S. Li, and C. Grebogi, "A novel multiplex network-based sensor information fusion model and its application to industrial multiphase flow system," *IEEE Trans. Ind. Informat.*, to be published.
- [26] M. Wang and L. Tian, "From time series to complex networks: The phase space coarse graining," *Phys. A, Stat. Mech. Appl.*, vol. 461, pp. 456–468, Nov. 2016.
- [27] Z. K. Gao, Q. Cai, Y. X. Yang, N. Dong, and S. S. Zhang, "Visibility graph from adaptive optimal kernel time-frequency representation for classification of epileptiform EEG," *Int. J. Neural Syst.*, vol. 27, no. 4, p. 1750005, Jun. 2017.
- [28] Z.-K. Gao, S. Li, W.-D. Dang, Y.-X. Yang, Y. Do, and C. Grebogi, "Wavelet multiresolution complex network for analyzing multivariate nonlinear time series," *Int. J. Bifurcation Chaos*, vol. 27, no. 8, p. 1750123, Jul. 2017.
- [29] M. Wang et al., "Exact results of the limited penetrable horizontal visibility graph associated to random time series and its application," *Sci. Rep.*, vol. 8, no. 1, Mar. 2018, Art. no. 5130.
- [30] K. Chourmouziadis and P. D. Chatzoglou, "An intelligent short term stock trading fuzzy system for assisting investors in portfolio management," *Expert Syst. Appl.*, vol. 43, pp. 298–311, Jan. 2016.
- [31] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 599–619.
- [32] M. Wang et al., "Fluctuation behavior analysis of international crude oil and gasoline price based on complex network perspective," *Appl. Energy*, vol. 175, pp. 109–127, Aug. 2016.
- [33] M. Wang, L. Tian, and P. Zhou, "A novel approach for oil price forecasting based on data fluctuation network," *Energy Econ.*, vol. 71, pp. 201–212, Mar. 2018.



[34] M. Wang et al., "A novel hybrid method of forecasting crude oil prices using complex network science and artificial intelligence algorithms," *Appl. Energy*, vol. 220, pp. 480–495, Jun. 2018.

[35] W. Jin, Z. J. Li, L. S. Wei, and H. Zhen, "The improvements of BP neural network learning algorithm," in *Proc. WCCC ICSP*, 2000, pp. 1647–1649.

[36] F. Schwenker, H. A. Kestler, and G. Palm, "Three learning phases for radial-basis-function networks," *Neural Netw.*, vol. 14, nos. 4–5, pp. 439–458, May 2001.

[37] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, Oct. 2006.

[38] S. Prestwich, R. Rossi, S. A. Tarim, and B. Hnich, "Mean-based error measures for intermittent demand forecasting," *Int. J. Prod. Res.*, vol. 52, no. 22, pp. 6782–6791, Nov. 2014.

[39] D. Stathakis, "How many hidden layers and nodes?" *Int. J. Remote Sens.*, vol. 30, no. 8, pp. 2133–2147, Apr. 2009.



**LIN CHEN** received the B.S. degree in accounting and the Ph.D. degree in management from Xi'an Jiaotong University in 2006. From 2011 to 2012, she was a Visiting Scholar with the University of Michigan. She is currently a Professor with the School of Management, Northwestern Polytechnical University, China. She has authored one book and more than 20 articles. Her research interests include performance evaluation, corporate governance, econometrics, and big data analytics in financial markets.



experimental economics, behavioral finance, and complexity science in economics and management.

**ZHILIN QIAO** received the Ph.D. degree in management from Xi'an Jiaotong University, China, in 2006. From 2006 to 2008, he did research at the Antai College of Economics and Management, Shanghai Jiao Tong University. From 2011 to 2012, he was a Visiting Scholar at the University of Michigan. He is currently an Associate Professor with the School of Economics and Finance, Xi'an Jiaotong University. He has authored over 20 academic articles. His research interests include

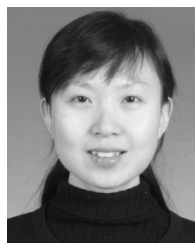


**MINGGANG WANG** received the Ph.D. degree in science from Nanjing Normal University in 2015. Since 2017, he has been a Visiting Scholar at Boston University, USA. He is currently an Associate Professor with the School of Mathematics, Nanjing Normal University, China. He has authored more than 20 articles in scientific reports, applied energy, and energy economics. His research interests include complexity science in economics and management, econometrics, time series analysis, data mining, and big data analysis.



and engineering at BJTU. His research has been published in *Computers & Industrial Engineering* and *Engineering Computations*.

**CHAO WANG** received the Ph.D. degree from Beijing Jiaotong University (BJTU) in 2015 with joint training by Purdue University in 2013 and 2014. He is currently an Associate Professor with the School of Economics and Management, Beijing University of Technology, China. He is also a Post-Doctoral Fellow at Boston University, USA. His research focuses on metaheuristics in optimization problem and econophysics. He conducted his doctoral study in management science



**RUIJIN DU** has been a Visiting Scholar at Boston University, USA, since 2017. She is currently an Associate Professor with the Faculty of Science, Jiangsu University, China. Her current interests include robustness of complex networks, modeling and analysis of complex system, data mining, and big data analysis. She has published more than 20 academic papers in *Applied Energy*, *Physical Review E*, *Scientific Reports*, *Chaos*, *Europhysics Letters*, and *Physica A*.



**HARRY EUGENE STANLEY** received the Ph.D. degree in physics from Harvard University in 1967. He is currently an American Physicist and a University Professor with Boston University, USA. He has made fundamental contributions to complex systems and is one of the founding fathers of econophysics. His current research interests include complexity science and econometrics. He was elected to the U.S. National Academy of Sciences in 2004.

...