Stock price prediction based on deep neural networks

2 authors, including:

Xuesong Yan
China University of Geosciences
32 PUBLICATIONS   **301** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   NSFC Key Project : Spatio-temporal perspective and intelligent management for urban water resources & environment system based on big geodata View project
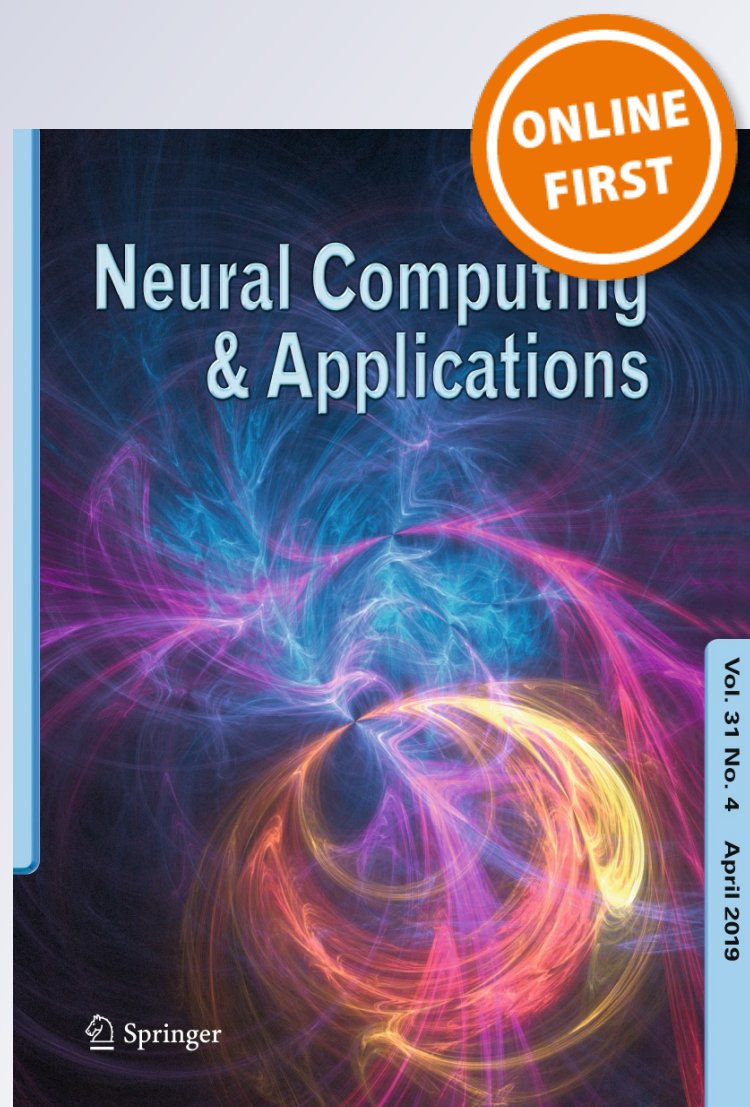
Stock price prediction based on deep neural networks

Pengfei Yu & Xuesong Yan

ONLINE FIRST

Neural Computing & Applications

Vol. 31 No. 4   April 2019

Springer

Springer

Springer

DEEP LEARNING FOR BIG DATA ANALYTICS

Stock price prediction based on deep neural networks

Pengfei Yu[1,2] · Xuesong Yan[1,2]

Abstract
Understanding the pattern of financial activities and predicting their development and changes are research hotspots in academic and financial circles. Because financial data contain complex, incomplete and fuzzy information, predicting their development trends is an extremely difficult challenge. Fluctuations in financial data depend on a myriad of correlated constantly changing factors. Therefore, predicting and analysing financial data are a nonlinear, time-dependent problem. Deep neural networks (DNNs) combine the advantages of deep learning (DL) and neural networks and can be used to solve nonlinear problems more satisfactorily compared to conventional machine learning algorithms. In this paper, financial product price data are treated as a one-dimensional series generated by the projection of a chaotic system composed of multiple factors into the time dimension, and the price series is reconstructed using the time series phase-space reconstruction (PSR) method. A DNN-based prediction model is designed based on the PSR method and a long- and short-term memory networks (LSTMs) for DL and used to predict stock prices. The proposed and some other prediction models are used to predict multiple stock indices for different periods. A comparison of the results shows that the proposed prediction model has higher prediction accuracy.

Keywords Financial data prediction · Neural networks · Deep learning · Phase-space reconstruction

1 Introduction

In recent years, financial activities have been increasingly growing in number with rapid economic development, and their variation trend has also become increasingly complex. Understanding the pattern of financial activities and predicting their development and changes are focuses of research in academic and financial circles. An approximate prediction of financial data using one or a series of methods can help understand the development of and changes in the financial market at the macroscopic level and provide a basis for investors and for-profit organizations to make trade decisions and plans at the microscopic level, thereby

allowing them to maximize profits. Because financial data contain complex, incomplete and fuzzy information, predicting their development trend is an extremely difficult challenge [1, 2].

Before the emergence of efficient machine learning algorithms, researchers in China and elsewhere generally used various statistical and econometric methods to build prediction models for research. Conventional statistical and econometric models require linear models and cannot be used to predict and analyse financial products before transforming nonlinear models to linear models. As an important branch of machine learning algorithms, neural networks (NNs) have the following advantages compared to conventional statistical methods: they are numeric, data-driven and adaptive. Therefore, NNs have a higher capacity to analyse inaccurate and noisy data and have been extensively used to predict time series. By design, the newly emerged deep learning (DL) algorithms can be trained with massive volumes of nonlinear data and used to construct deep NNs (DNNs) with multiple hidden layers and capture more abstract nonlinear relationships between data. Compared to conventional machine learning

✉ Xuesong Yan
yanxs@cug.edu.cn

Pengfei Yu
pfyu@cug.edu.cn

[1] School of Computer Science, China University of Geosciences, Wuhan 430074, Hubei, China

[2] Hubei Key Laboratory of Intelligent Geo-Information Processing, Wuhan 430074, Hubei, China

algorithms, these DL algorithms can be used to solve nonlinear problems more satisfactorily [3].

In 1988, White predicted the changes in daily stock returns of IBM using NNs [4]. Zhang combined an auto-regressive integrated moving average (ARIMA) model with an artificial NN (ANN) to predict time series and conducted a comparative study. The results showed that the ANN was more advantageous in analysing and processing nonlinear data [5]. In 2009, Vanstone and Finnie [6] proposed an NN-based empirical methodology for designing automated trade systems based on ANNs. Jasemi et al. [7] studied the information hidden in Japanese stock candle-stick charts using a multi-layer perception (MLP) model. Ticknor proposed a stock index price prediction model that uses a Bayesian network and determined its effectiveness based on the data for Microsoft Corp. and Goldman Sachs Group Inc. stocks [8]. Wang [9] proposed a stochastic time effective function NN and combined it with principal component analysis to predict the prices of multiple securities. Xiong et al. predicted the linear pattern of a time series using an ARIMA model, estimated the nonlinear residual of the time series using an NN model and combined the results to form the final prediction for the time series. They examined the effectiveness of the hybrid prediction model through simulations based on a RMB exchange rate series [10]. Based on the theory for the relationship between volume and price, Wu et al. [11] performed an empirical study on the relationship between volume and price in China's stock market using the exponential generalized auto-regressive conditional model and a backpropagation NN. Li [12] predicted error series using empirical mode decomposition in conjunction with a support vector machine (SVM).

The primary objective of DL is to obtain highly robust feature sets from complex real-world data and, on this basis, acquire highly correlated information. Deep nonlinear topologies of DNNs can be built by stacking the hidden layers of the used nonlinear activation functions and used to analyse and model data with complex structures. Compared to conventional shallow NNs, DNNs can better represent complex high-dimensional functions, such as highly varying functions [13]. In addition, the data processing capacity of computer chips has improved dramatically. In particular, graphic processing units are highly suitable for deep model training. Moreover, increasingly many research results have been achieved in the related fields, such as machine learning and information and signal processing. Furthermore, relevant techniques for storing large datasets have also improved. These allow DL algorithms to fit highly nonlinear functions better and undergo rapid popularization and development. Today, DL has achieved marked results in the classification, speech recognition and computer vision fields. Shen et al.

constructed a deep belief network using a continuous restricted Boltzmann machine and combined it with the conjugate gradient method to predict exchange rates. Through a comparison, they found that the DL network structure was superior to conventional feedforward NNs [14]. Ding et al. predicted the changes in stock index prices from an event-driven perspective. They constructed a prediction model using a convolutional NN and used it to measure the short- and long-term effects of events on the changes in stock prices. The accuracy of their method was 6% higher than that of a conventional method for the stocks in the Standard and Poor's (S&P) 500 [15]. Zhao et al. [16] constructed a prediction model using a de-noising auto-encoder and performed multiple sets of experiments based on datasets extended using the bagging method. They also used the prediction model to predict oil prices. Krauss et al. studied the application of several integrated methods, including DNNs, gradient boosted regression trees and random forests, in statistical arbitrage. In addition, they also proposed an equal-weighted integrated model based on multiple models and, with it, achieved excess earnings in the S&P 500 [17]. In recent years, there are some researches on the prediction of stock price by deep learning method [18–20].

2 Financial data prediction and DNNs

2.1 Financial data prediction

Financial data, particularly stock index data, often treated as a time series affected by multiple variable factors. These factors can generally be classified into two types: macroscopic variables, which affect the market on a long-term scale (e.g. economic policies or national gross domestic product), and microscopic variables, which affect the market on a microscopic scale (e.g. random events, irrational emotional fluctuations of investors and market rumours). By analysing macroscopic and microscopic factors, decisions can be made for the financial market, which is a dynamic system with constantly changing factors. However, because these factors often change, it can be considered that they are affected by an unknown system. It is nearly impossible to collect all the macroscopic and microscopic variables and determine the extents of their influence. Therefore, financial time series prediction is often viewed as one of the main difficult research areas in the literature relating to time series and machine learning.

The first step of financial time series analysis is to observe financial time series data. Commonly observed financial variables include price (stock price, stock index, exchange rate and futures price), return (stock return, stock index return, interest rate and futures return), fluctuation,

trade volume and companies' financial variables (bond issuance and hedging tools). Because the rate of return is unaffected by the scale of investment and, as a stable series, exhibits excellent statistical properties, it is often used as a measure of trading experiments.

Assuming P_t is the asset price at time t, a single-period simple gross return will be earned if one asset is held between time $t-1$ and time t when dividends and dividend returns are not considered in formula (1):

$$1+R_t = \frac{P_t}{P_{t-1}} \text{ or } P_t = P_t(1+R_t) \tag{1}$$

The corresponding single-period simple return is given as formula (2):

$$R_t = \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{2}$$

Compared to ordinary time series, financial time series also have some unique statistical properties, including the following:

Outliers	A financial time series can be easily affected by macroscopic factors (e.g. human and policy factors) and contains data that differ relatively significantly from the original data
Trend	A mid-/long-term upward or downward trend can be observed from a financial time series in any observation dimension
Mean reversion	A financial time series has a tendency to revert to its mean level
Fluctuation focus	Large fluctuations are often closely followed by large fluctuations, whereas small fluctuations are often closely followed by small fluctuations

2.2 DNNs

NNs are an important area of the machine learning field and are a type of data-driven algorithm. They are adaptive, have a relatively strong capacity to approximate nonlinear functions and can be used to satisfactorily predict nonlinear, time-varying data. However, NNs often rely heavily on the feature attributes of the input data. Sparse input features and missing data will likely result in under-fitted predictions. If the input features are highly correlated and there are more feature attributes than samples, there will be an extended model training process and a risk of over-fitting. ANNs are the precursor of DNNs, which currently are a focus of research in the machine learning field. DNNs differ from conventional ANNs in two main areas. (1) Most DNNs consist of multiple hidden layers or multi-stage nonlinear information processing processes. (2)

Continuous, deeper and more abstract hidden layers are constructed using supervised methods or non-supervised feature representation methods. By incorporating knowledge and advanced technologies of many fields, DL has evolved into an enormous branch in the technical field and system structure of machine learning. However, the basic feature of DL is still to build multi-layer models for processing nonlinear information. Based on the model structure, the employed technique and the scene of application, DL is approximately categorized into two types, namely unsupervised (semi-supervised) learning and supervised learning. Of supervised and unsupervised learning models, supervised DL models are more efficient in the training and testing processes, have more flexible frameworks and are more suitable for the end-to-end learning process of complex systems compared to conventional dichotomous models. In contrast, unsupervised DL models, particularly probabilistic production models, are easier to understand and can be more easily embedded with field knowledge, combined with other models and used to address uncertain problems. Nonetheless, it is often difficult to use unsupervised DL models to process the reasoning and learning of complex systems. Figure 1 shows the approximate categorization of DL model structures.

3 DNN-based stock price prediction method

3.1 Description of problem

A characteristic of the price time series for most financial products is that they can be treated as data series that change continuously within the trade range. However, because there are certain time intervals within the trade range, all financial time series cannot be treated as completely continuous data. In addition, generally, there is a relatively significant difference in trends between series observed in different intervals. Here, stock index data are taken as an example. A long-term upward trend is observed in a seasonal interval. However, in this long-term period, the mid- and short-term trends observed in daily or hourly intervals may fluctuate or even be downward during many stages. To reduce the number of factors affecting the experiments and ensure consistency in the comparison experiments and considering China's T+1 trading system, stock price data are selected as the main study object in the experiments conducted in this study and trading days are used as observation intervals. A discrete stock index price series is recorded: $y_1, y_2, \ldots, y_t, \ldots, y_n$. Thus, when predicting a financial time series, all the historical data are known at time t and used for searching for a nonlinear process that emulates price changes. A prediction model is constructed and used to predict y_t, as indicated in formula

Fig. 1 Schematic diagram of the categorization of DL model structures
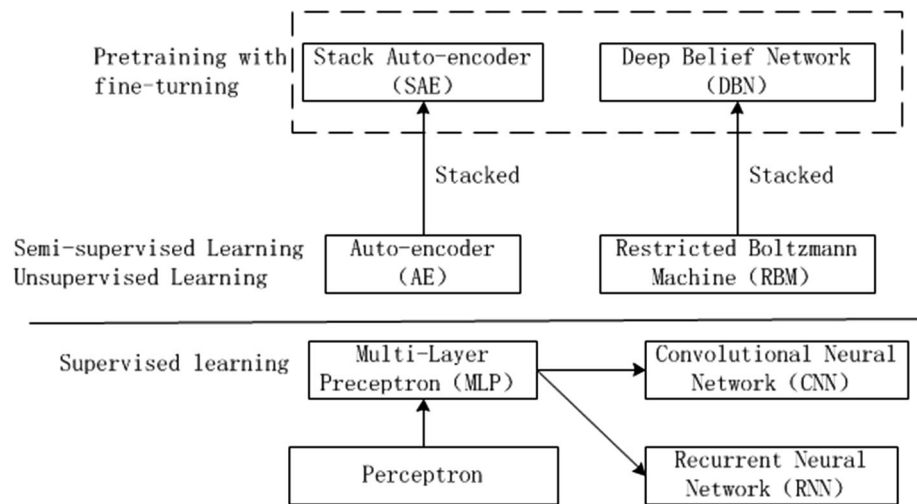
(3). The ultimate goal is to select a suitable model and optimal parameters based on the observed time series data to approximate the process f.

$$y_t - s(t) = f(y_{t-d}, y_{t-d-1}, \ldots, y_{t-d-n+1}) + w(t) \qquad (3)$$

where d is the duration of the delay, n is the time span that requires consideration and $w(t)$ is the noise in the data observed at time t.

To more clearly describe the analysis and prediction of stock index price series, the process of building a stock index price prediction model is abstracted into three stages, namely data analysis and processing, prediction model building and experimental result prediction and evaluation,

as shown in Fig. 2. The first stage involves pre-processing processes (e.g. data acquisition and de-noising), phase-space reconstruction (PSR) and data structuring processes (e.g. label defining and test set partitioning) and eventually produces an input dataset and an output dataset effectively applicable to the prediction model. At the second stage, a prediction model is built using various machine learning methods. In addition, reasonable model parameters are selected, and the model is adjusted and optimized. Furthermore, prediction is performed using the model based on the generated input and output data. At the third stage, evaluation metrics for prediction results are defined, and the results obtained using the prediction model are
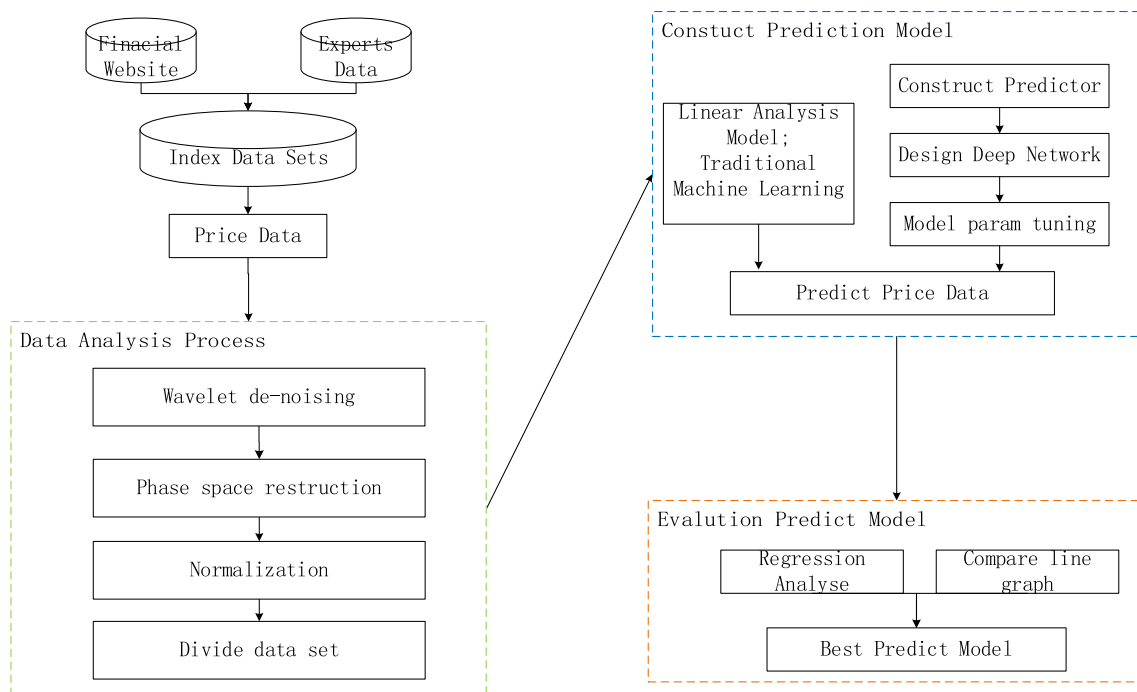

Fig. 2 Framework of the stock index price series prediction process

subjected to multi-dimensional analysis. The evaluation results are presented in the form of charts.

3.2 PSR of price data

It has been only several decades since most financial products that focus on stock indices were created. The number of time series data points generated with trading days as observation intervals is basically within several thousands and slightly smaller than those in the datasets in other machine learning fields. When the dataset contains insufficient data, a model trained using a machine learning method is often over-fitted. In this study, regarding the data volume, data are folded and replicated multiple times using the common sliding window method. Regarding the dimensionality, price data are extended dimensionally through PSR. At time t, a known time series $[y_t = x(t),\ t = 1, \ldots, n]$ that the undergone PSR, which can be expressed as formula (4):

$$X(n) = x(n),\ x(n + \tau), \ldots, x(n + (m - 1) \cdot \tau) \qquad (4)$$

The series consisting of the reconstructed data is given as formula (5):

$$\begin{cases} X(1) = x(1),\ x(1 + \tau), \ldots, x(1 + (m - 1) \cdot \tau) \\ X(2) = x(2),\ x(2 + \tau), \ldots, x(2 + (m - 1) \cdot \tau) \\ \qquad\qquad\qquad \vdots \\ X(M) = X(t - (m - 1) \cdot \tau) = [x(t - (m - 1) \cdot \tau), \\ x(t - (m - 2) \cdot \tau), \ldots, x(t + (m - m) \cdot \tau)] \end{cases} \qquad (5)$$

where τ is the delay interval, which allows the two observation data $x(t)$ and $x(t + \tau)$ at the endpoints of τ to be independent of one another to some extent but not completely uncorrelated and m is the embedding dimension of the generated series. Reasonably extending the embedding dimension can increase the analytical potential of a dynamic system structure.

In this study, τ and m are calculated using the mutual information method and an improved Cao false nearest neighbour method, respectively. PSR is performed after τ and m are determined. The reconstructed data are denoted by $X(t)$. The input and output of the supervised learning model are then defined as follows: the input is the k number of reconstructed data points before time t: $X(i), i = t - k, t - k + 1, \ldots, t - 1$., and the output is the reconstructed data at time t: $X(t)$. In the last step of the experiment, the data of the last dimension $x(t - (m - 1)\tau)$ of the predicted reconstructed data $X(t)$ are used as the predicted price series, which is subsequently evaluated by comparison with the actual price data.

3.3 DNN-based prediction model

Recurrent NNs (RNNs) [21] achieve explicit modelling of time through self-connection of the hidden layers and record long-term information by improving the nodes in the hidden layers. RNNs have achieved marked results in natural language processing and audio frequency analysis. In conventional RNNs, there are links between the nodes in the hidden layers. Owing to these recurrent feedback links, network models have a memory ability. Thus, RNNs can model information on a time scale. The duration of information transfer can be treated as the model depth. However, earlier RNNs are unable to model information with a long time span and can lead to a vanishing gradient problem when used to build large time scale models. By optimizing the nodes, deep RNNs are able to efficiently model on a time scale and prevent the occurrence of a vanishing gradient problem.

RNN models that use long- and short-term memory (LSTM) nodes are effective and expandable when used to address a number of problems involving series data. These models are for general use and effective in capturing long-term time-dependent information and have achieved exceptional results in handwriting recognition, concise natural language translation and audio frequency data analysis. An LSTM [22] node consists of so-called dynamic gate structures, including input, forget and output gate structures, as shown in Fig. 3.

In addition, an LSTM node structure also contains a recurrent logic structure, which is referred to as nerve cell and used to record node information.

An LSTM node is described below:

1. Input data weight: W_i, W_f, W_c, W_o.
2. Recurrent data weight: R_i, R_f, R_c, R_o.
3. Peephole weight: $V \in \mathbb{R}^N$.
4. Offset: $b_i, b_f, b_c, b_o \in \mathbb{R}^N$.
5. $x_t,\ y_t$ are the input and output of the node at time t, respectively.
6. f_t is the output of the forget gate at time t, which is expressed as formula (6):

$$f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f) \qquad (6)$$

7. i_t is the output of the input gate at time t, which is expressed as formula (7):

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i) \qquad (7)$$

8. $\tilde{C}_t,\ C_t$ is the input and cell structure of the node at time t, respectively, which are expressed as formulas (8) and (9):

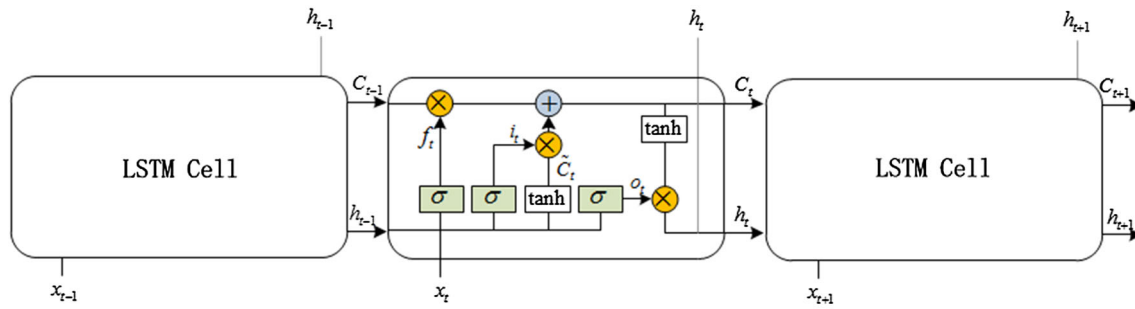$$\tilde{C} = \tanh(W_c x_t + R_c h_{t-1} + b_c) \qquad (8)$$

Fig. 3 Structural diagram of an LSTM neural node

$$C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1} \tag{9}$$

9. O_t is the output of the output gate, which is expressed as formula (10):

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o) \tag{10}$$

10. The final output h_t of the node is expressed as formula (11):

$$h = o_t \odot \tanh(c_t) \tag{11}$$

In an LSTM nerve cell model, the input gate determines how much information can be added to the nerve cell, and the output gate determines how much information can be output after processing. Evidently, the forget gate determines how much of the output of the previous moment will be retained for calculation in the subsequent moment. Due to their unique node structure, when used to model series data, LSTM NNs are able to capture mid- and long-term data and will not cause a time scale gradient vanishing problem, unlike earlier RNNs.

The model established in this study is a multi-hidden layer LSTM network structure consisting of LSTM nodes with an m-dimensional vector that has undergone PSR at time t, $X(t)$, as the input and the reconstructed vector at time $t + 1$, $X(t + 1)$, as the output. Figure 4 shows the established DNN-based model.

The input and output layers each contain an m number of nodes, corresponding to the reconstructed actual time series data at time t and the reconstructed predicted time series data at time $t+1$, respectively. The hidden layers consist of multiple LSTM node layers and are connected with one another the same manner as the hidden layers of RNNs mentioned earlier. Hidden layer architectures are mostly designed based on experience in previous studies. Hence, to reduce the number of comparison parameters in the experiment, all the hidden layers of the model are set to contain the same number of nodes. Thus, the model contains L hidden layers, each of which contains K LSTM nodes. An optimal combination of L and K is selected by traversing the grid in the experiment.

In addition, the final output is the reconstructed data on day $t+1$, which differs from normal regression experiments, which have only one output. Therefore, the last loss function of the model is corrected: the mean of the squares of the differences between any two dimensions is used as the loss function. Formula (12) describes the loss function of each training sample:

$$\text{Loss}\,(X(t+1), f(X(t))) = \frac{1}{m}\sum_{d=0}^{m}(x'(t+d\cdot\tau) - x(t+d\cdot\tau))^2 \tag{12}$$

3.4 Evaluation metrics for prediction results

As mentioned previously, the final model established in the experiment is an m-dimensional vector that has undergone PSR. To compare with other time series analysis methods and prediction models and ensure that the experimental results are normative, the final experimental results are calculated using the price data. In other words, the data of the last dimension of the reconstructed vector, $x'(t + (m - 1) \cdot \tau)$, are used as the final predicted values.

To effectively evaluate the established prediction model, the predicted price data are analysed using four metrics from prediction accuracy and error perspectives based on the actual price data.

Directional accuracy (DA) is expressed as formula (13):

$$DA = \frac{1}{n}\sum_{i=1}^{n} a_t, \; when \; a_t = \begin{cases} 1, & if \; (\text{trend}' = \text{trend}) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where trend' is the predicted trend and trend is the actual trend. Because noise is removed from the stock index data using the wavelet de-noising technique, the obtained price series contains identical values for consecutive days. Thus, the trend at time t is defined as formula (14):

Fig. 4 Structural diagram of the DNN-based LSTM price prediction model

$X'(t+1)$

$x'(t+1) \quad x'(t+1+\tau) \quad \cdots \quad x'(t+1+(m-1)\tau)$

Output

LSTM $\leftrightarrow$ LSTM $\leftrightarrow$ LSTM $\quad\cdots\quad$ LSTM

Hidden

K nodes

L layers

LSTM $\leftrightarrow$ LSTM $\leftrightarrow$ LSTM $\quad\cdots\quad$ LSTM

Input

$x(t) \quad x(t+\tau) \quad \cdots \quad x(t+(m-1)\tau)$

$X(t)$
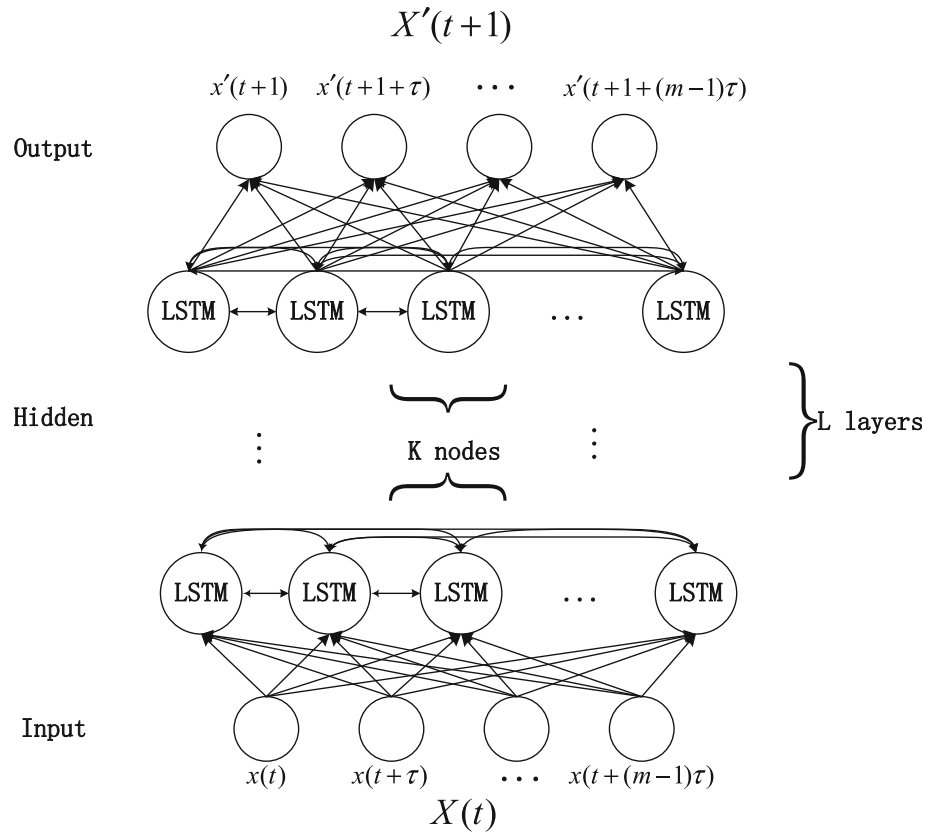
$$trend_t = \begin{cases} 0, & y_t - y_{t-1} < 0 \\ 1, & y_t - y_{t-1} > 0 \\ trend_{t-1}, & y_t - y_{t-1} = 0 \end{cases} \qquad (14)$$

Mean root square error (MRSE) is expressed as formula (15):

$$MRSE = \sqrt{\frac{1}{n}\sum_{t=1}^{t}(y_t - y'_t)^2} \qquad (15)$$

Mean absolute per cent error (MAPE) is expressed as formula (16):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - y'_t}{y_t}\right| \qquad (16)$$

Pearson correlation coefficient (CORR) is expressed as formula (17):

$$R = \frac{\sum_{t=1}^{n}(y_t - \bar{y}_t)(y'_t - \bar{y}'_t)}{\sqrt{\sum_{t=1}^{n}(y_t - \bar{y}_t)^2(y'_t - \bar{y}'_t)^2}} \qquad (17)$$

where y_t is the actual price and y'_t is the predicted price.

The prediction accuracy of the model is used as the main analytical metric for evaluating its prediction capacity, which is a direct estimate of the trend in the financial data. The higher DA is, the higher the prediction capacity of the model is. MRSE, MAPE and CORR are used to evaluate

the closeness of the predicted price data to the actual price data. The lower MRSE and MAPE are and the closer CORR is to 1, the more reliable the prediction results produced by the model are.

4 Experimental results and analysis

4.1 Description of the experimental data

Stock market index data were selected as experimental data based on the following several factors: high accessibility, stable market performance and relatively high analytical significance. Data for six stock indices for various market environments, namely the S&P 500, the Dow Jones industrial average (DJIA), the Nikkei 225 (N 225), the Hang Seng index (HSI), the China Securities index 300 (CSI 300) and the ChiNext index, were obtained from Yahoo Finance (finance.yahoo.com), TuShare financial data interface (tushare.org) and relevant organizations. The analytical data are the closing prices for each trading day. Table 1 presents a description of the selected data.

In this study, three sets of experiments were performed on the time series data for the six stock indices with an aim to establish a reasonable model structure and the effectiveness of the proposed PSR-DNN-combined approach in

Table 1 Statistics of closing prices of each stock index selected for the experiment

Index	S&P500	DJIA	Nikkei 225	HSI	CSI300	ChiNext
Begin date	20080102	20080102	20080104	20080102	20080102	20100602
End date	20171229	20171229	20171229	20171229	20171229	20171229
Mean	1613.20	14474.31	13548.91	22008.99	3020.05	1484.27
Median	1461.12	13552.44	13345.03	22340.84	3025.96	1359.32
Max	2690.16	24837.51	22939.18	30003.49	5731.76	3982.25
Min	676.53	6547.05	7054.98	11015.84	1627.76	593.66
Std.	482.49	3905.66	4171.40	3086.08	685.12	652.63
Samples	2518	2518	2449	2465	2434	1844

analysing and predicting financial time series was examined.

4.2 PSR analysis of various stock indices

Before establishing a prediction model, τ and m were first determined for each stock index using the PSR method, as shown in Figs. 5, 6, 7, 8, 9 and 10.

As shown in the mutual information delay variation chart and m discriminant chart for each stock index, there was no significant difference in τ and m between the stock indices. For mutual information, its first and second minimum values were reached at τ of 3 and 6, respectively, for all the stock indices. For m, its value basically approached and reached the extreme value at the fifth and sixth times, respectively, and the values at the fifth and sixth times were basically close.

Based on the above description of the experiment, the PSR parameters for all the stock index price data were set as follows: $\tau = 3$, and $m = 5$. The reconstructed series of a known price series $[y_t = x(t), t = 1, 2, \ldots, n]$ is as follows:

$$X(t) = x(t), x(t + 3), \ldots, x(t + 5.3),$$
$$t = 1, 2, \ldots, n - 6 \tag{18}$$

To demonstrate the effectiveness of the PSR method, a comparison experiment was performed on the subsequently adjusted and optimized LSTM prediction model. Table 2 summarizes the experimental results (the mean of each evaluation metric for each stock index obtained from multiple sets of experiments based on the data for a 6-year period).

The experimental results show a relatively significant difference in prediction accuracy for the S&P 500 and the ChiNext index: the prediction accuracy for these two indices was, on average, over 3% higher when the PSR method was used than when the PSR method was not used. For other stock indices, the prediction accuracy was approximately 1% higher when the PSR method was used than when the PSR method was not used. Overall, although the use of the PSR method will decrease the goodness of fit

between the predicted and actual values, more attention is paid to increasing the prediction accuracy in practical application. The results also show that the MRSE and R were within their respective acceptable ranges. Thus, it can be concluded that the use of the PSR method can help improve the model's prediction performance.

4.3 Analysis of the DNN structure

The three main parameters of the established DNN-based prediction model, namely the number of hidden layers, the number of LSTM nodes in the hidden layers and the activation function of the LSTM output, were determined through an experiment. Other hyperparameters were similarly set based on the grid searching method and the relevant literature [18, 19] as follows: time step, 10; initial learning rate, 0.001; the initial weights follow the normal distribution pattern; probability for preservation of nodes, 0.4; and number of iterations, 5000. To facilitate better observation and comparison of the experimental results, PSR, de-noising and normalization were first performed in the model parameter experiment. In addition, relatively fine experimental data were selected: 500 sets of data for the S&P 500 for a nearly 2-year (January 4, 2010 to February 3, 2012) period.

First, the number of LSTM nodes in the hidden layers was analysed through an experiment under the following conditions: number of hidden layers, 3, and activation function of the output, hyperbolic tangent (tanh) function. Table 3 and Fig. 11 present the experimental results for the test set.

The experimental results demonstrate that there was no significant difference between the graphs plotted by the established single-LSTM hidden layer models with various numbers of nodes. The data in Table 3 indicate that the model prediction accuracy changed with the number of nodes in the hidden layers. Relatively excellent results were obtained when the number of nodes in the hidden layers was set to 20, 32 or 48. Thus, the number of nodes in the hidden layers of the subsequent experimental model was set to 32.
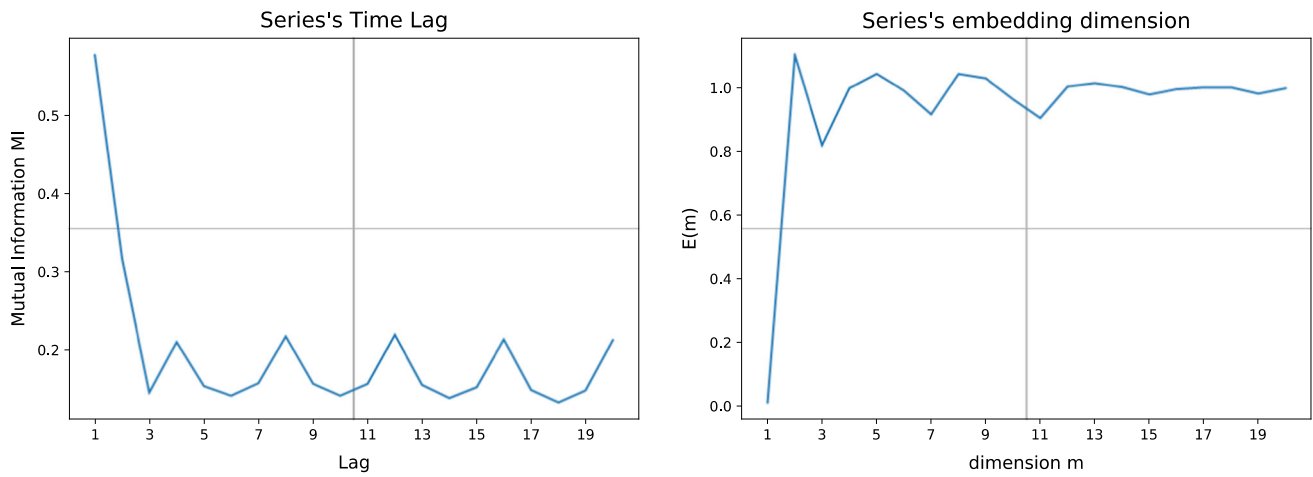
Fig. 5 τ and m discriminant charts for the S&P 500
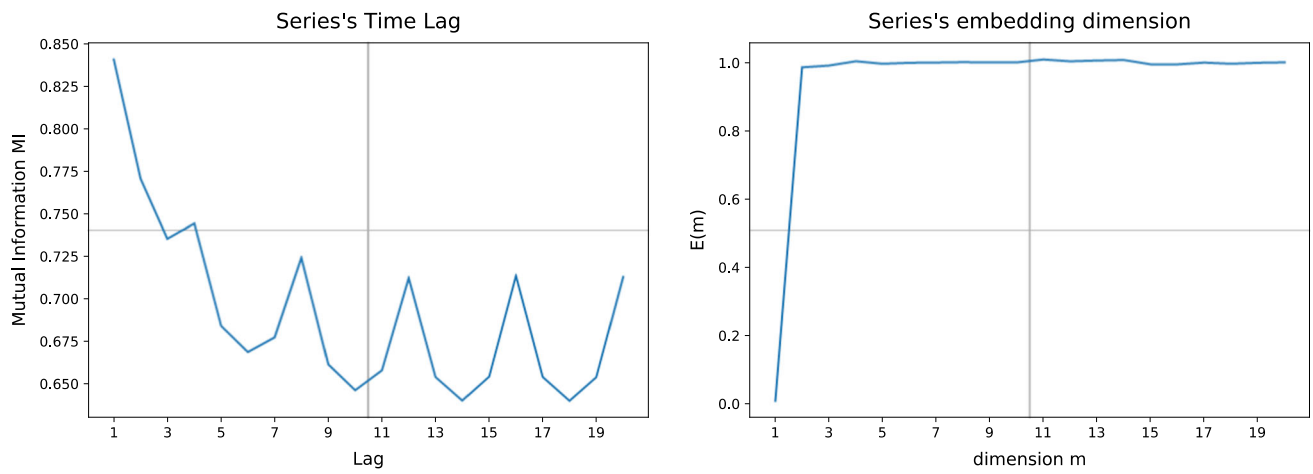
Fig. 6 τ and m discriminant charts for the DJIA
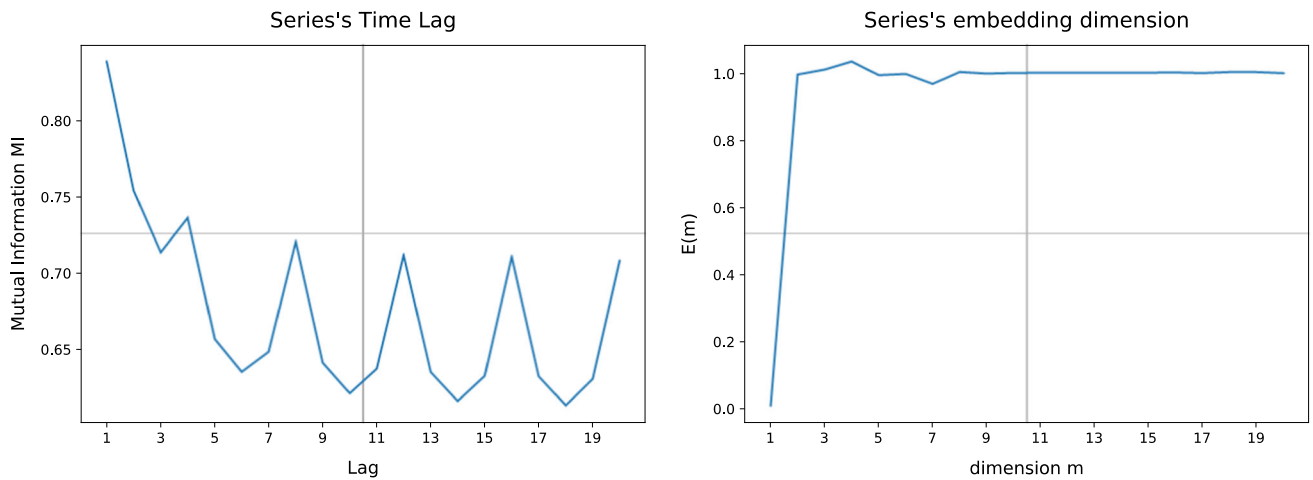
Fig. 7 τ and m discriminant charts for the N 225
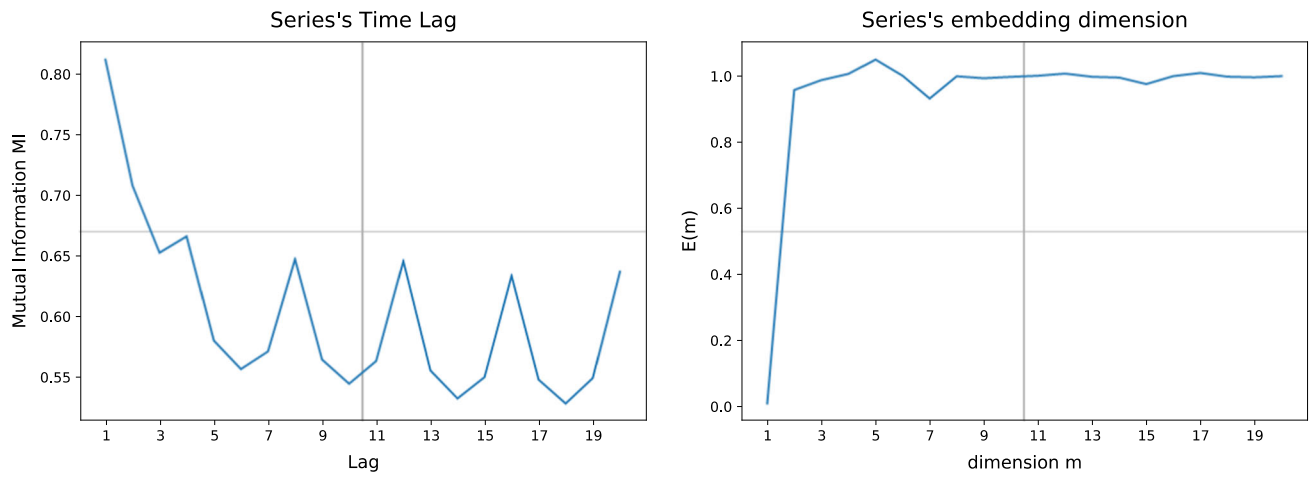
Fig. 8 τ and m discriminant charts for the HSI
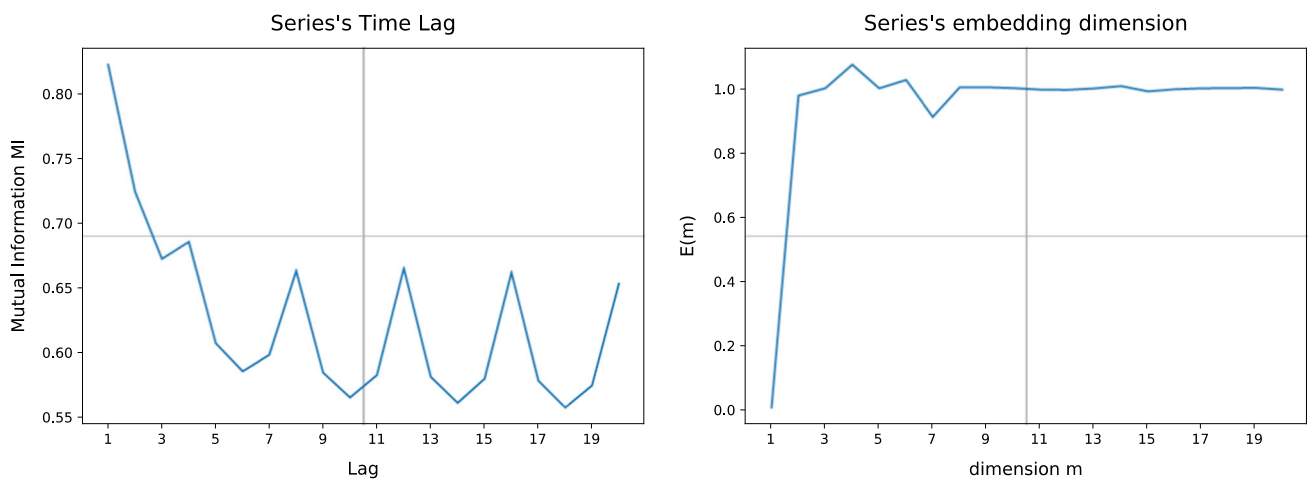

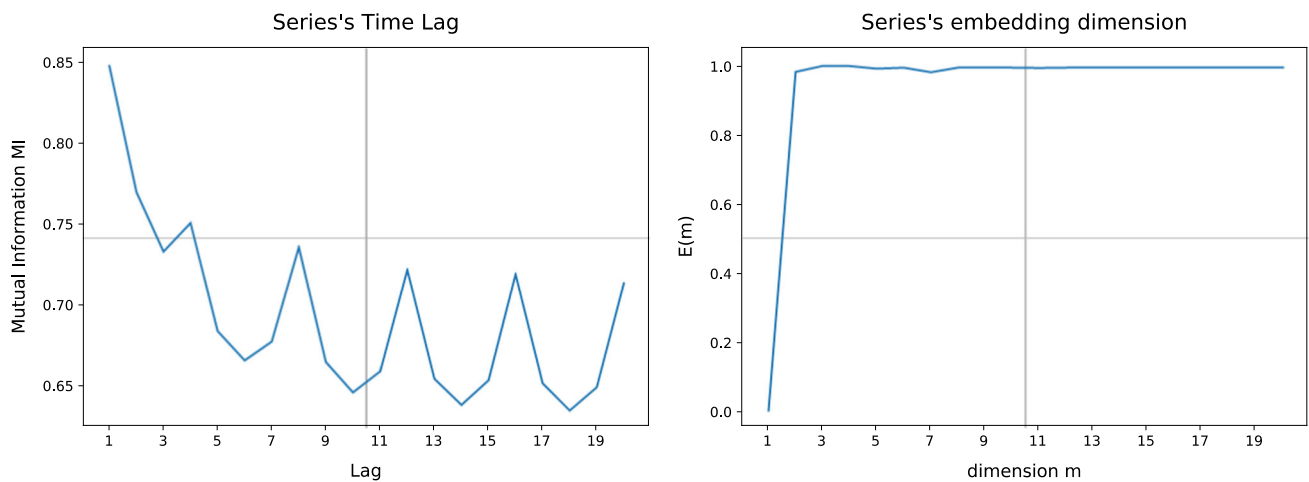**Fig. 9** τ and m discriminant charts for the CSI 300

Fig. 10 τ and m discriminant charts for the ChiNext index

Table 2 Comparison of experimental results obtained when the PSR method was used and was not used

Index	Using PSR	DA (%)	MRSE	MPAE (%)	R
S&P500	Yes	59.50	6.47*1e−2	0.87	0.93
	No	55.17	3.38*1e−2	0.63	0.95
DJIA	Yes	57.71	6.84*1e−2	0.49	0.97
	No	56.45	5.23*1e−2	0.67	0.97
Nikkei225	Yes	55.92	1.283*1e−1	1.95	0.92
	No	53.35	6.56*1e−2	0.81	0.97
HSI	Yes	57.36	4.86*1e−2	0.71	0.97
	No	56.55	4.311e−2	0.59	0.97
CSI300	Yes	56.67	3.46*1e−2	0.80	0.96
	No	54.94	3.46*1e−2	0.75	0.95
ChiNext	Yes	57.23	2.129*1e−1	3.98	0.93
	No	54.42	1.115*1e−1	2.52	0.94

Table 3 Results of the comparison experiment on the number of nodes in the hidden layers of the model

Number of node	DA	MRSE	MPAE	R
15	58.0%	3.96*1e−2	1.02*1e−3	0.969
18	60.0%	4.04*1e−2	2.06*1e−3	0.968
20	61.9%	4.00*1e−2	2.98*1e−3	0.969
24	61.2%	3.88*1e−2	1.22*1e−3	0.969
28	57.4%	3.98*1e−2	1.86*1e−3	0.968
32	63.8%	4.21*1e−2	2.81*1e−3	0.968
36	59.4%	3.93*1e−2	1.98*1e−3	0.969
40	58.1%	4.25*1e−2	3.67*1e−3	0.967
48	62.6%	4.25*1e−2	3.67*1e−3	0.968
56	61.9%	4.78*1e−2	5.2*1e−3	0.964

After the number of LSTM nodes in the hidden layers was determined, the number of hidden layers was determined through an experiment. Models established in most of the relevant studies (including those on RNNs) do not contain too many hidden layers. Thus, the experimental range for the number of hidden layers was set to [1, 8]. Table 4 and Fig. 12 present the experimental results.

The experimental results show that as the number of hidden layers increased, data fluctuations occurred in greater intervals and became smoother. There was a relatively large difference between the predicted and actual values. Considering the very small number of sample sets used as training data, it is prudent to establish a model with not too many layers to prevent over-fitting. As demonstrated by the experimental results, the best prediction results were achieved with three hidden layers.

After obtaining a deep RNN network with three hidden layers, each of which contains 32 nodes, the activation function of the final output gate of each LSTM node was determined through an experiment. The tanh and linear rectification (ReLU) functions were selected. Table 5 and Fig. 13 present the experimental results.

Clearly, the fitted predicted values for the intervals with relatively significant fluctuations obtained when the ReLU function was selected as the activation function of the output gate of each LSTM node were inferior to those obtained when the tanh function was selected as the activation function.

4.4 Performance analysis of the prediction model

To determine the effectiveness of the established model in predicting financial data, multiple prediction models were used to predict the data for the six stock indices for various market environments. Table 6 summarizes the description of the comparison experimental models. To facilitate analysis and evaluation, the experimental results were consolidated by year and are presented in Figs. 14, 15, 16, 17, 18 and 19 and Tables 7, 8, 9, 10, 11 and 12.

To facilitate observation, the figures show the data for 2016, including the actual data and the data predicted by various prediction models. Each dataset contains approximately 235 data points. Although there are some slight differences between the curves produced by the models for different stock indices, common findings are described below:

Regarding the goodness of fit to the series, the deep LSTM network model was superior to the SVM and deep MLP models.

The sections of the curves produced by the deep LSTM network model and the deep MLP model for small fluctuations in the stock indices were softer, whereas the corresponding results predicted by the support vector regression (SVR) and ARIMA methods were basically identical to those of the previous day. Tables 7, 8, 9, 10, 11 and 12 summarize the experimental data in more detail.

The following can be found from the experimental data:

(1) The maximum prediction accuracy of the LSTM model reached 62.87% (for the S&P 500 for 2012).

(2) There were some differences in prediction results between different stock indices and between different periods. On average, the predication accuracy for the S&P 500 was the highest, whereas the prediction accuracy for the ChiNext index was the lowest.

(3) Based on MRSE, MAPE and CORR, the goodness of fit of the results produced by the deep LSTM network model was slightly lower than that of the
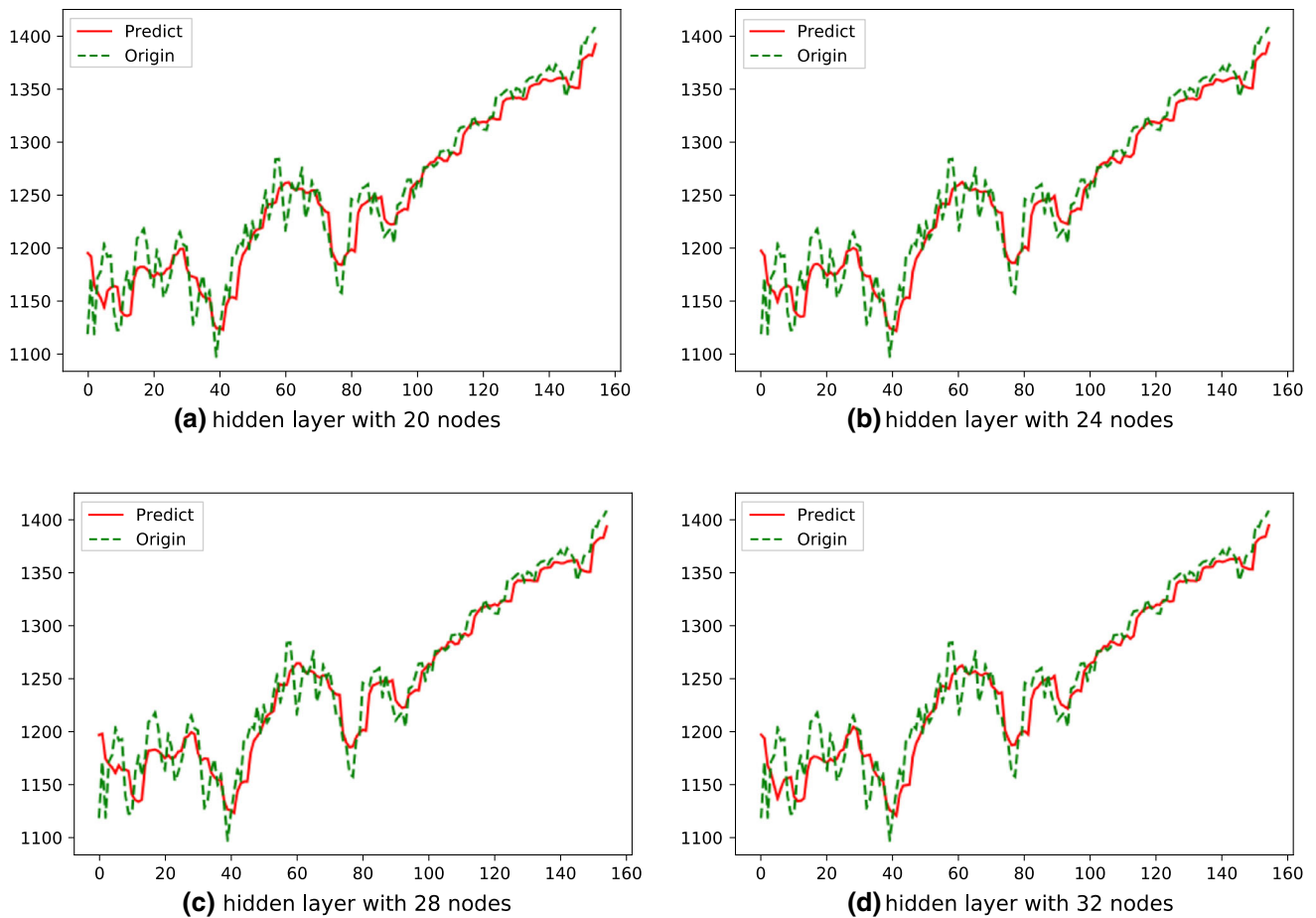
Fig. 11 Comparison of partial results of the experiment on the number of nodes in the hidden layers

Table 4 Results of the comparison experiment on the number of hidden layers

Number of layer	DA	MRSE	MPAE	R
1	57.4%	4.69*1e−2	3.09*1e−3	0.952
2	60.0%	4.08*1e−2	2.06*1e−3	0.968
3	63.8%	4.35*1e−2	2.11*1e−3	0.964
4	58.9%	7.48*1e−2	6.21*1e−3	0.929
5	56.7%	9.18*1e−2	1.03*1e−2	0.925
6	54.2%	1.08*1e−1	8.81*1e−3	0.882
7	53.2%	1.26*1e−2	8.36*1e−3	0.821
8	52.2%	1.71*1e−2	4.98*1e−3	0.704

results produced by the ARIMA method but higher than those of the results produced by the SVR and MLP models.

(4) Based on DA, the prediction accuracy of the LSTM model for the price series (average: 56.85%) was

much higher than that of the ARIMA model and 3–5% higher than those of the SVR and MLP models.

The prediction model was established in this study based only on the historical price data without considering macroscopic factors (e.g. relevant policies, market system and investor psychology). The S&P 500 is an index for a developed market in which investors are more rational and policies are sounder; consequently, its price fluctuations are more in line with the market rules. The average prediction accuracy for the ChiNext index was the lowest. However, of all the stock indices, the ChiNext index is the youngest (inaugurated in October 2009). ChiNext is a second-tier stock market, which can be seen as a low-entry-requirement, high-risk and stringently regulated stock market. As a result, the ChiNext index is more easily affected by macroscopic factors, such as policy control, industrial changes and investor psychology. These factors cannot be determined based on price data alone. Nevertheless, the DNN-based prediction model was still able to achieve relatively high prediction accuracy for some time intervals.
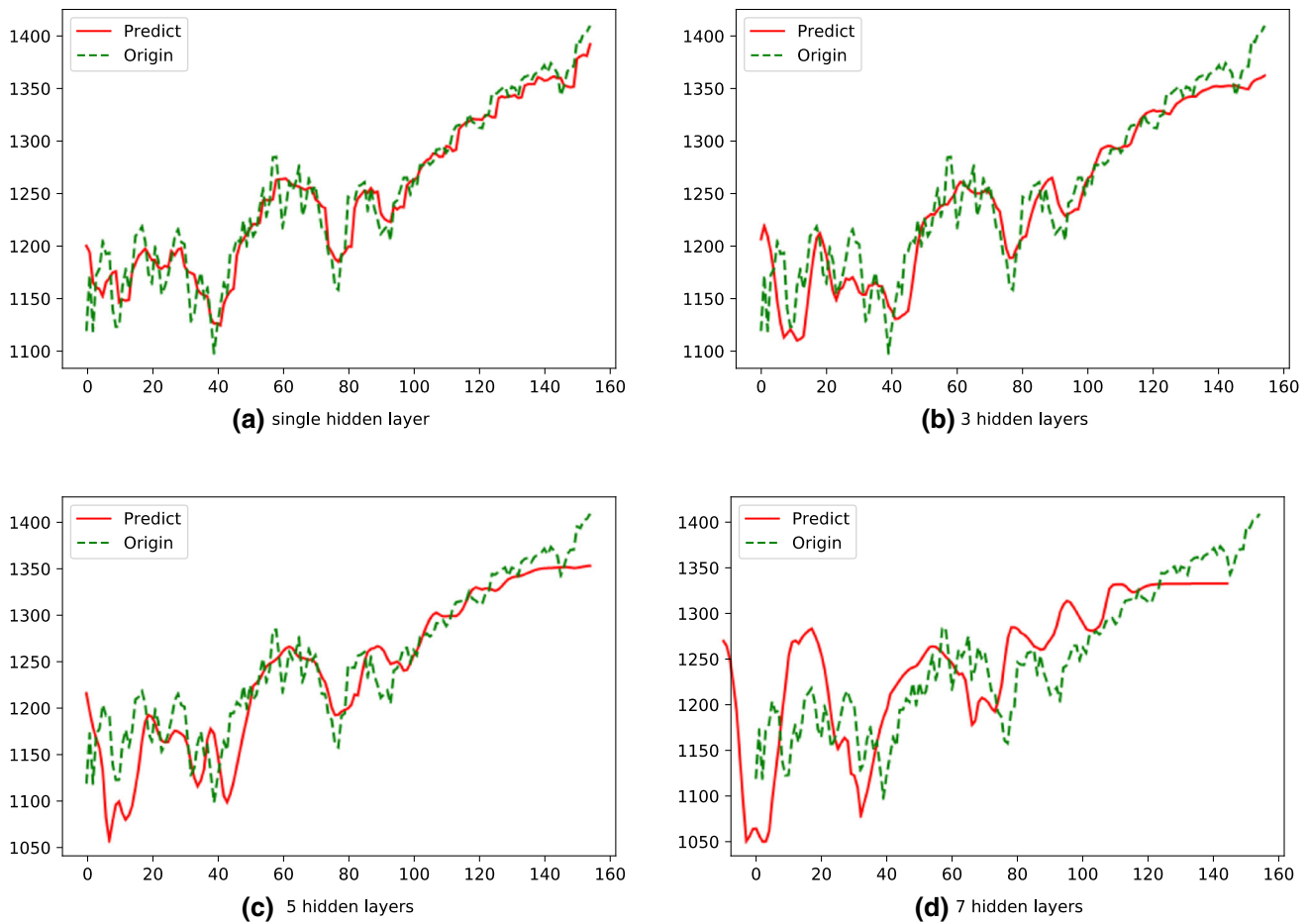
Fig. 12 Comparison of the results of the experiment on the number of hidden layers

Table 5 Experimental results obtained when different activation functions of the output gate were selected

Activate function	DA	MRSE	MPAE	R
Tanh	61.2%	4.35*1e−2	2.11*1e−3	0.964
Relu	58.0%	1.34*1e−1	1.06*1e−2	0.856

Thus, the errors in the above-mentioned experimental results are within the acceptable range and are in basic agreement with the estimation. The deep LSTM network prediction model that uses the PSR method for price series analysis can relatively accurately predict financial data.
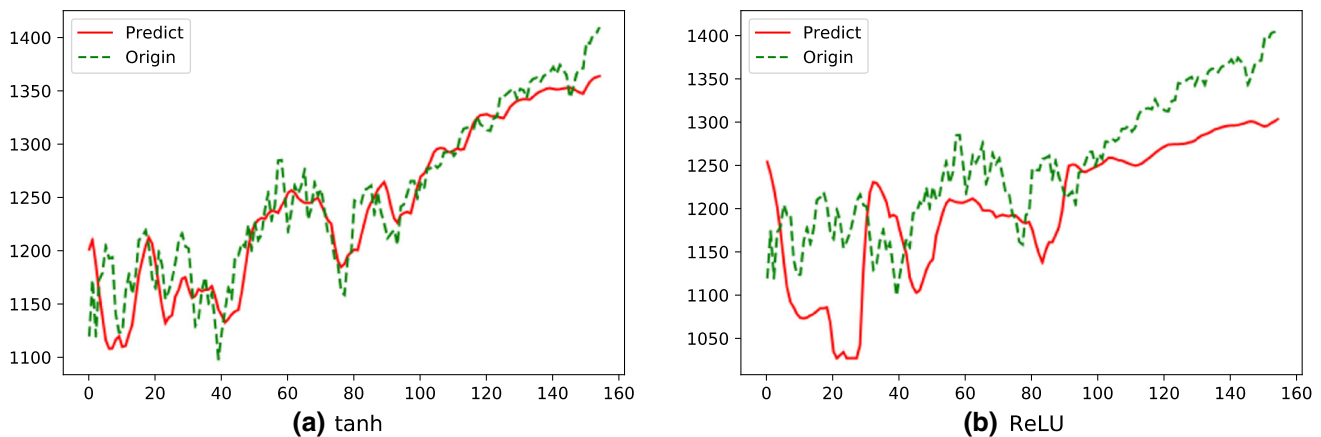
Fig. 13 Comparison of the experimental results for the functions of the output gate of each LSTM node

Table 6 Description of the prediction models selected for comparison

Model name	PSR	Pre-processing	Hyper parameters
Deep long–short-term memory network (LSTMs)	Delay: 3 Embedded dimension: 5	Wavelet de-noising, normalization	Hidden layers: 3 Nodes of each layer: 32 Time steps: 5, Dropout: 0.6
Deep multi-layer perception (MLP)	Delay: 3 Embedded dimension: 5	Wavelet de-noising, normalization	Hidden layers: 7层 Nodes of layers: [64, 64, 32, 32, 16, 16, 16] Dropout: 0.4
Support vector regression (SVR)	No	Normalization	Cost: 100. Gamma: 10.0
Auto-regressive integrated moving average (ARIMA)	No	No	Auto-regressive p: 6 Integrated d: 0 Moving average q: 2
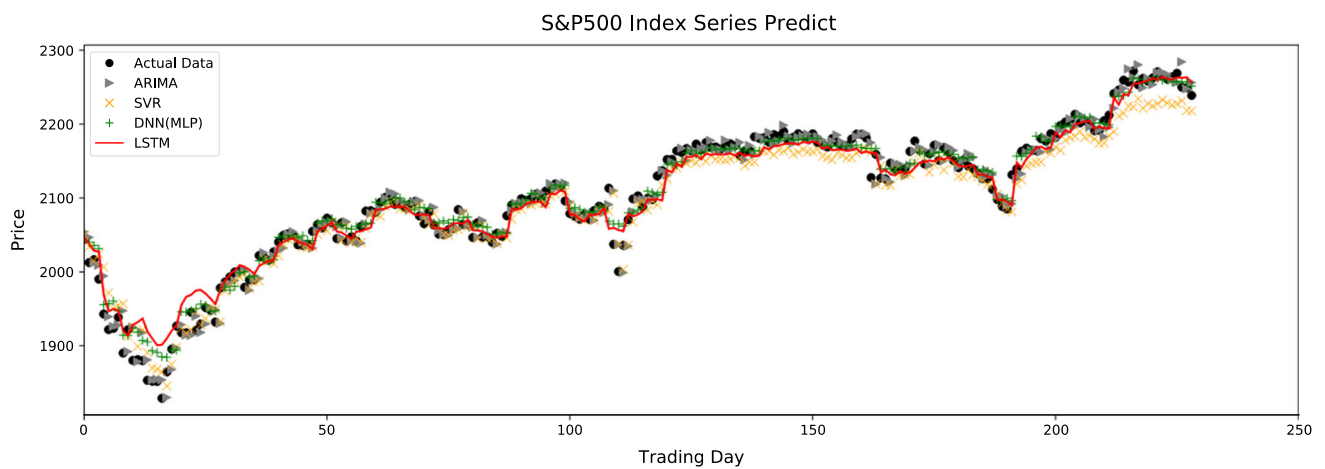
Fig. 14 Comparison of the results produced by various prediction models for the S&P 500
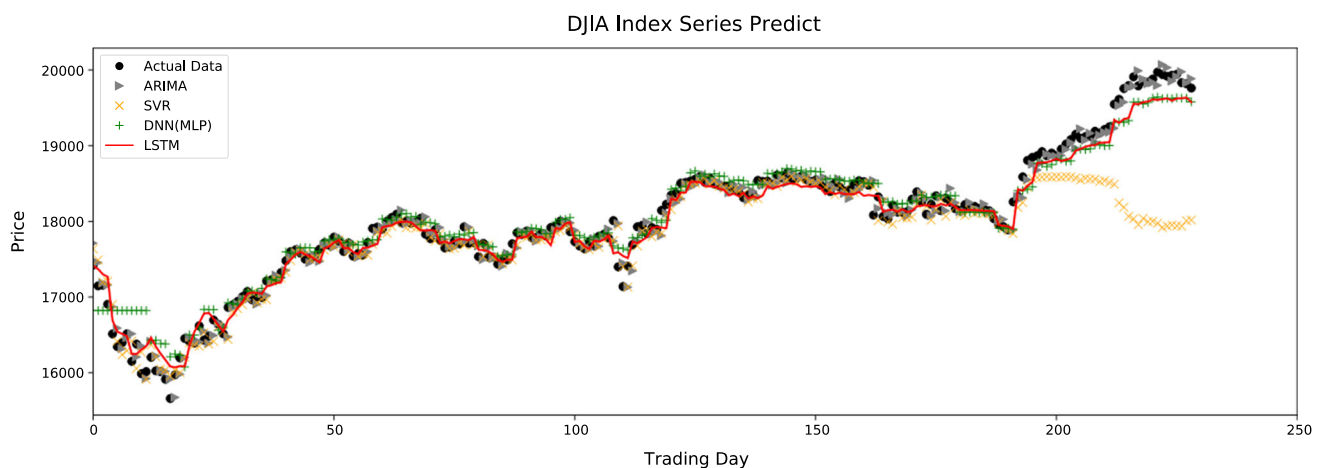
Fig. 15 Comparison of the results produced by various prediction models for the DJIA
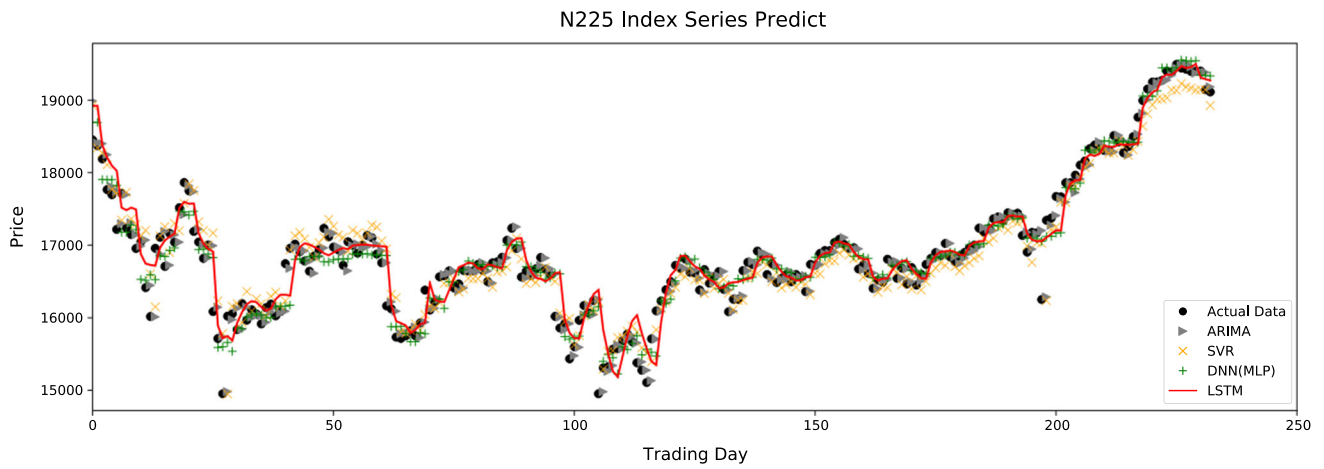
Fig. 16 Comparison of the results produced by various prediction models for the N 225

Fig. 17 Comparison of the results produced by various prediction models for the HSI

Fig. 18 Comparison of the results produced by various prediction models for the CSI 300

Fig. 19 Comparison of the results produced by various prediction models for the ChiNext index

Table 7 Comparison of the results produced by the prediction models for the S&P 500

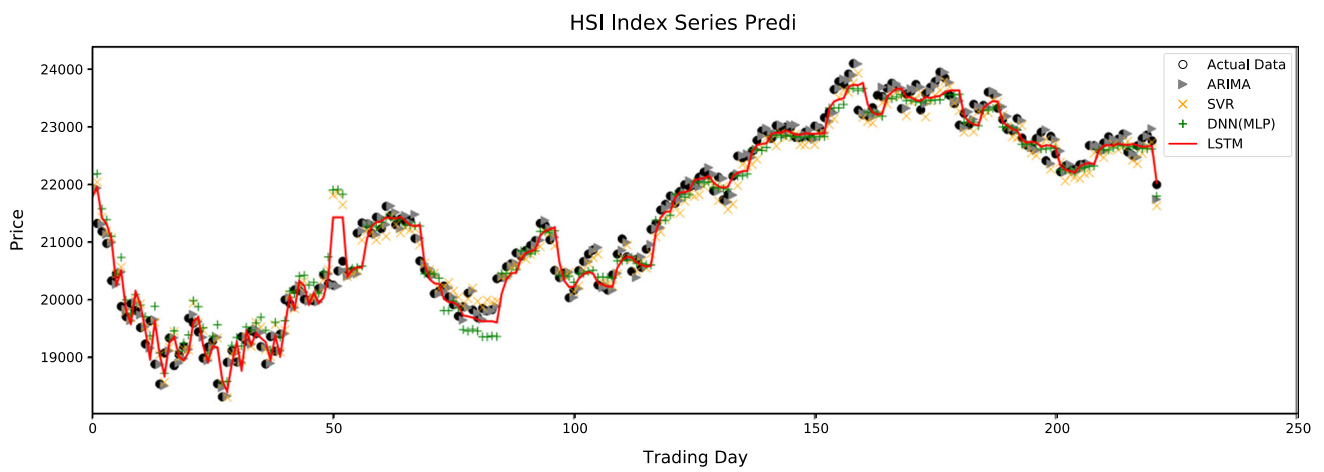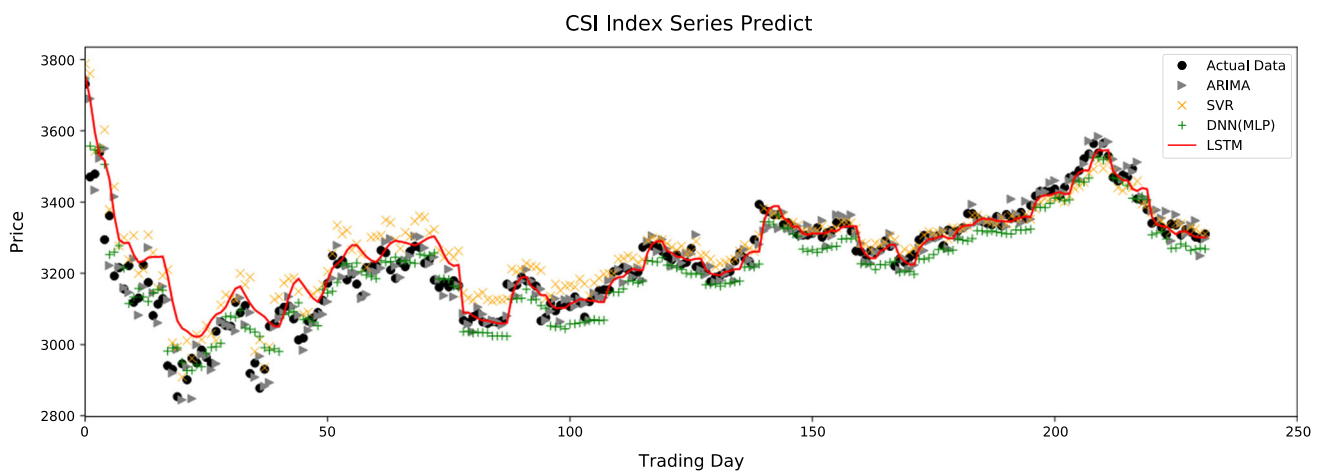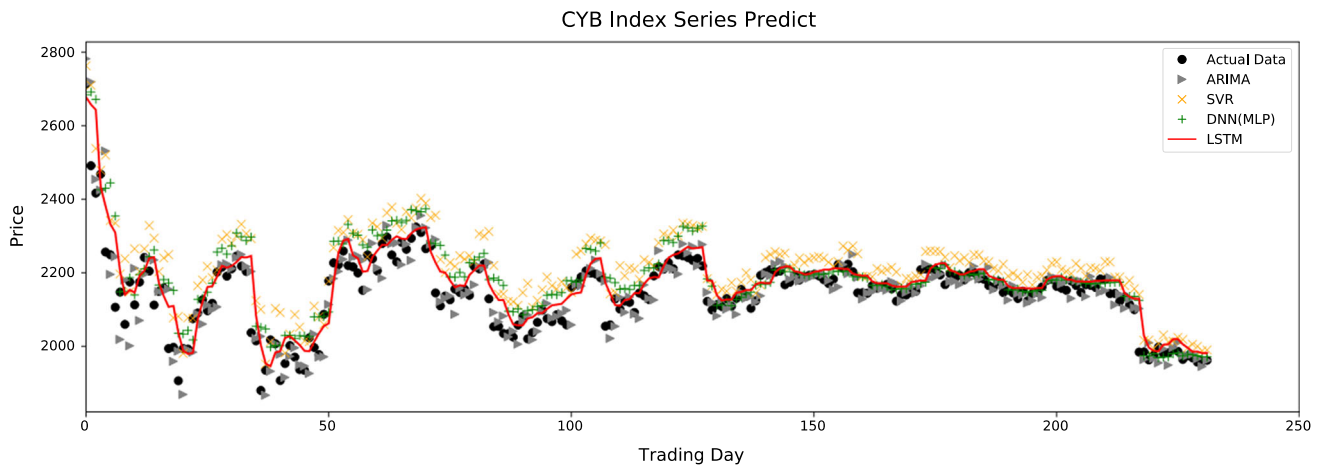| Years	2010	2011	2012	2013	2014	2015	2016	2017	Average
Evaluating indicator	Direction accuracy (DA) %								
ARIMA	51.77	52.33	49.19	50.41	54.35	48.39	51.54	51.36	51.16
SVR	56.08	55.46	57.44	53.24	52.18	53.06	56.72	55.08	54.90
MLP	59.02	55.81	59.09	55.08	53.28	52.87	53.39	47.06	54.45
LSTM	62.76	57.08	62.87	60.18	49.37	55.65	62.39	54.31	58.07
Evaluating indicator	Mean root square error (RMSE) 1.0*1e−2								
ARIMA	5.38	5.12	4.19	3.56	3.47	4.60	5.73	4.22	4.53
SVR	6.93	7.88	6.07	14.20	8.86	5.44	6.99	17.66	10.50
MLP	2.87	4.46	3.20	16.87	3.70	3.97	5.68	3.07	5.48
LSTM	2.55	3.84	5.91	8.23	15.41	11.88	8.29	7.22	7.92
Evaluating indicator	Mean absolute error per cent (MAPE) %								
ARIMA	2.14	0.07	0.03	0.14	0.08	0.01	0.03	0.11	0.33
SVR	1.07	1.73	1.10	3.08	1.73	0.73	0.29	3.56	1.66
MLP	1.09	0.05	0.15	3.11	0.24	0.25	0.32	0.23	0.68
LSTM	0.60	0.02	0.81	1.41	4.02	2.44	0.30	0.79	1.30
Evaluating indicator	Correlation coefficient (CORR)								
ARIMA	0.95	0.95	0.96	0.99	0.97	0.92	0.98	0.96	0.96
SVR	0.96	0.92	0.97	0.96	0.92	0.91	0.99	0.95	0.95
MLP	0.98	0.95	0.97	0.87	0.98	0.92	0.99	0.98	0.96
LSTM	0.97	0.96	0.94	0.98	0.86	0.88	0.98	0.97	0.94

5 Conclusions

In this paper, a prediction model is proposed for financial price data, which are non-stationary and relatively noisy time series, through three process steps, namely time series data processing, network model building and result evaluation and analysis. The PSR method for time series analysis is combined with a DNN-based LSTM network model. Data de-noising and normalization are performed at the pre-processing stage. In addition, data structuring is also achieved by partitioning using a time window.

Subsequently, a DNN-based model is designed and optimized by selecting an optimum activation function and optimization method. Ultimately, four metrics are used to evaluate the effectiveness of the model. Finally, various prediction methods, including the conventional ARIMA linear analytical method, the conventional SVR machine learning method, a deep MLP model, a deep LSTM model involving no PSR process and the deep LSTM combined with PSR, are compared. The comparison of the prediction results for various stock indices for various periods demonstrates that the established DNN-based prediction

Table 8 Comparison of the results produced by the prediction models for the DJIA

Years	2010	2011	2012	2013	2014	2015	2016	2017	Average
Evaluating indicator	Direction accuracy (DA) %								
ARIMA	50.81	52.46	52.44	52.46	49.56	49.77	53.44	47.54	51.06
SVR	51.61	53.11	47.56	52.05	51.41	53.44	46.31	52.28	50.97
MLP	57.38	55.51	59.09	51.69	50.41	54.92	57.20	51.26	54.68
LSTM	61.51	57.52	56.12	58.41	57.32	56.07	61.06	54.31	57.79
Evaluating indicator	Mean root square error (RMSE) 1.0*1e−2								
ARIMA	3.78	4.93	3.89	3.58	3.68	5.15	5.24	4.52	4.35
SVR	7.65	4.21	9.88	10.01	5.21	7.82	8.34	11.54	8.08
MLP	2.35	3.86	3.44	2.82	4.64	6.05	7.31	15.60	5.76
LSTM	2.00	3.59	4.27	5.29	3.39	3.75	5.95	18.77	5.88
Evaluating indicator	Mean absolute error per cent (MAPE) %								
ARIMA	0.98	0.12	0.05	0.09	0.02	0.02	0.08	0.06	0.18
SVR	1.10	0.82	1.57	2.54	0.36	1.21	1.01	1.35	1.25
MLP	0.58	0.55	0.38	0.32	0.24	0.52	0.22	1.94	0.59
LSTM	0.26	0.08	0.63	0.76	0.26	0.08	0.24	2.28	0.57
Evaluating indicator	Correlation coefficient (CORR)								
ARIMA	0.95	0.94	0.94	0.98	0.97	0.95	0.99	0.96	0.96
SVR	0.95	0.92	0.92	0.88	0.91	0.93	0.89	0.94	0.92
MLP	0.97	0.96	0.96	0.99	0.97	0.93	0.98	0.98	0.97
LSTM	0.98	0.96	0.95	0.99	0.98	0.96	0.99	0.98	0.97

Table 9 Comparison of the results produced by the prediction models for the N 225

Year	2010	2011	2012	2013	2014	2015	2016	2017	Average
Evaluating indicator	Direction accuracy (DA) %								
ARIMA	51.46	46.41	52.05	51.47	49.58	52.08	50.62	46.20	49.98
SVR	46.44	51.90	51.23	47.30	53.08	52.15	51.87	47.83	50.23
MLP	53.19	53.71	58.33	56.12	52.44	53.41	49.37	50.28	53.36
LSTM	56.28	57.32	54.66	54.50	59.09	56.32	51.93	54.24	55.54
Evaluating indicator	Mean root square error (RMSE) 1.0*1e−2								
ARIMA	2.77	5.53	3.25	5.36	3.23	4.48	4.24	2.54	3.93
SVR	3.56	19.84	3.79	19.62	5.21	7.82	4.56	4.43	8.60
MLP	2.40	35.76	3.70	7.92	4.64	6.05	4.66	2.25	8.42
LSTM	2.04	7.87	3.66	13.23	6.43	4.91	3.99	2.63	5.60
Evaluating indicator	Mean absolute error per cent (MAPE) %								
ARIMA	0.49	0.07	0.07	0.04	0.13	0.11	0.04	0.07	0.13
SVR	1.04	2.64	0.24	3.06	0.36	1.21	0.04	1.10	1.21
MLP	0.62	6.69	0.09	1.10	0.24	0.52	0.31	0.04	1.20
LSTM	0.11	0.92	0.23	2.39	1.18	1.08	0.29	0.43	0.83
Evaluating indicator	Correlation coefficient (CORR)								
ARIMA	0.98	0.98	0.98	0.99	0.98	0.97	0.95	0.95	0.97
SVR	0.97	0.78	0.98	0.89	0.91	0.93	0.95	0.95	0.92
MLP	0.98	0.59	0.98	0.99	0.97	0.93	0.94	0.95	0.92
LSTM	0.98	0.97	0.98	0.90	0.97	0.96	0.96	0.96	0.96

Table 10 Comparison of the results produced by the prediction models for the HSI

Year	2010	2011	2012	2013	2014	2015	2016	2017	Average
Evaluating indicator	Direction accuracy (DA) %								
ARIMA	49.80	48.32	46.69	50.83	52.30	49.79	49.79	47.06	49.32
SVR	52.48	48.74	46.28	51.67	49.37	55.44	53.24	51.76	51.12
MLP	54.36	58.70	55.88	52.54	60.61	53.14	50.22	56.89	55.29
LSTM	59.49	59.46	59.40	50.00	54.71	49.79	61.54	59.88	56.78
Evaluating indicator	Mean root square error (RMSE) 1.0*1e−2								
ARIMA	1.94	4.12	2.60	2.93	3.67	4.75	2.94	1.82	3.10
SVR	7.86	8.52	4.04	4.30	8.22	6.07	14.20	4.62	7.23
MLP	1.80	4.61	2.37	4.01	3.71	4.83	14.62	5.55	7.23
LSTM	3.75	5.53	2.29	3.07	5.04	6.68	13.77	1.85	5.25
Evaluating indicator	Mean absolute error per cent (MAPE) %								
ARIMA	0.03	0.10	0.07	0.02	0.01	0.03	0.00	0.11	0.05
SVR	1.74	1.37	1.12	0.61	0.96	1.10	3.08	1.58	1.45
MLP	0.29	0.02	0.06	0.71	0.07	0.10	2.53	1.86	0.71
LSTM	0.59	0.23	0.01	0.08	0.33	0.44	1.95	0.26	0.48
Evaluating indicator	Correlation coefficient (CORR)								
ARIMA	0.98	0.99	0.98	0.97	0.97	0.99	0.99	0.99	0.98
SVR	0.93	0.93	0.97	0.94	0.92	0.97	0.96	0.99	0.95
MLP	0.98	0.99	0.98	0.97	0.98	0.99	0.81	0.99	0.96
LSTM	0.97	0.98	0.98	0.97	0.97	0.99	0.82	0.99	0.96

Table 11 Comparison of the results produced by the prediction models for the CSI 300

Year	2010	2011	2012	2013	2014	2015	2016	2017	Average
Evaluating indicator	Direction accuracy (DA) %								
ARIMA	48.74	50.00	49.79	55.98	48.52	51.67	47.50	51.28	50.44
SVR	49.16	46.19	48.12	55.13	47.68	53.24	47.08	53.85	50.06
MLP	54.27	53.95	54.47	53.04	56.77	54.24	47.03	53.72	53.44
LSTM	58.70	57.27	55.84	54.87	54.75	56.44	50.86	60.33	56.13
Evaluating indicator	Mean root square error (RMSE) 1.0*1e−2								
ARIMA	2.27	3.00	2.94	4.41	5.77	15.93	1.62	1.08	4.63
SVR	3.57	9.54	17.47	4.78	15.75	4.62	2.22	2.68	7.58
MLP	2.28	6.89	11.22	3.90	7.42	12.08	1.54	2.47	7.23
LSTM	1.94	5.77	8.11	3.92	18.23	5.34	2.22	1.84	5.92
Evaluating indicator	Mean absolute error per cent (MAPE) %								
ARIMA	0.12	0.18	0.02	0.04	0.20	0.04	0.11	0.07	0.10
SVR	1.50	0.64	1.36	0.62	1.46	1.13	1.26	1.42	1.17
MLP	0.65	0.55	3.31	0.14	0.26	1.78	0.67	1.03	1.05
LSTM	0.21	0.81	1.63	0.04	2.18	0.79	1.25	0.52	0.93
Evaluating indicator	Correlation coefficient (CORR)								
ARIMA	0.98	0.99	0.98	0.97	0.99	0.98	0.94	0.99	0.98
SVR	0.93	0.93	0.92	0.97	0.88	0.97	0.91	0.98	0.94
MLP	0.98	0.97	0.86	0.97	0.99	0.98	0.94	0.99	0.96
LSTM	0.99	0.98	0.90	0.97	0.86	0.95	0.90	0.99	0.94

Table 12 Comparison of the results produced by the prediction models for the ChiNext index

Year	2012	2013	2014	2015	2016	2017	Average
Evaluating indicator	Direction accuracy (DA) %						
ARIMA	53.30	50.85	54.36	50.00	41.67	50.72	50.15
SVR	56.13	48.72	52.05	51.41	41.67	46.38	49.39
MLP	55.56	47.39	51.90	52.63	54.24	46.97	51.45
LSTM	55.00	52.44	51.93	61.07	54.98	58.73	55.69
Evaluating indicator	Mean root square error (RMSE) $1.0*1e{-}2$						
ARIMA	1.92	4.48	4.06	11.54	1.97	0.84	4.14
SVR	5.96	21.32	9.21	5.21	3.19	1.49	7.73
MLP	2.96	6.15	2.62	10.99	2.51	1.86	4.52
LSTM	2.72	12.47	3.99	2.56	1.92	1.25	4.15
Evaluating indicator	Mean absolute error per cent (MAPE) %						
ARIMA	0.01	0.27	0.08	0.37	0.17	0.06	0.16
SVR	2.44	6.84	1.84	0.36	3.30	1.71	2.75
MLP	0.93	1.15	0.24	0.71	1.98	1.32	1.06
LSTM	0.97	3.39	0.29	0.45	1.24	1.32	1.28
Evaluating indicator	Correlation coefficient (CORR)						
ARIMA	0.95	0.99	0.95	0.99	0.87	0.84	0.93
SVR	0.66	0.89	0.90	0.91	0.88	0.87	0.85
MLP	0.92	0.99	0.97	0.99	0.87	0.65	0.90
LSTM	0.93	0.97	0.69	0.97	0.90	0.86	0.89

model displays a higher prediction capacity than the other models.

Compliance with ethical standards

Conflict of interest No conflicts of interest of this work.

References

1. Zhang Li, Wang Fulin, Bing Xu, Chi Wenyu, Wang Qiongya, Sun Ting (2018) Prediction of stock prices based on LM-BP neural network and the estimation of overfitting point by RDCI. Neural Comput Appl 30(5):1425–1444
2. Oliveira ALI, Meira SRL (2006) Detecting novelties in time series through neural networks forecasting with robust confidence intervals. Neurocomputing 70(1):79–92
3. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521 (7553):436
4. White H (1988) Economic prediction using neural networks: the case of IBM daily stock returns 451–458
5. Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159–175
6. Vanstone B, Finnie G (2009) An empirical methodology for developing stockmarket trading systems using artificial neural networks. Expert Syst Appl 36(3):6668–6680
7. Jasemi M, Kimiagari AM, Memariani A (2011) A modern neural network model to do stock market timing on the basis of the ancient investment technique of Japanese Candlestick. Expert Syst Appl 38(4):3884–3890
8. Ticknor JL (2013) A Bayesian regularized artificial neural network for stock market forecasting. Expert Syst Appl 40 (14):5501–5506
9. Wang J (2015) Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks. Neurocomputing 156:68–78
10. Xiong Z (2011) Research on RMB exchange rate forecasting model based on combining ARIMA with Neural networks. J Quant Tech Econ 28(06):64–76 **(in Chinese)**
11. Wu Q, Wang C, Tang Y (2013) Empirical research on volume-price relationship based on GARCH models and BP neural network. J Sichuan Univ Nat Sci Edn 50(04):703–708 **(in Chinese)**
12. Li X, Zhang Z (2014) Support vector machine method for financial time series prediction based on simultaneous error prediction. J Tianjin Univ Sci Technol 47(01):86–94 **(in Chinese)**
13. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
14. Shen F, Chao J, Zhao J (2015) Forecasting exchange rate using deep belief networks and conjugate gradient method. Neurocomputing 167:243–253
15. Ding X, Zhang Y, Liu T, et al (2015) Deep learning for event-driven stock prediction. In Twenty-fourth international joint conference on artificial intelligence, pp 2327–2333
16. Zhao Y, Li J, Yu L (2017) A deep learning ensemble approach for crude oil price forecasting. Energy Econ 66:9–16
17. Krauss C, Do XA, Huck N (2017) Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S&P 500. Eur J Oper Res 259(2):689–702

18. Song Y, Lee JW, Lee J (2018) A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction. Appl Intell 49:1–15

19. Minh DL, Sadeghi-Niaraki A, Huy HD et al (2018) Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. IEEE Access 6:1–1

20. Göçken M, Özçalici M, Boru A, Dosdogru AT (2019) Stock price prediction using hybrid soft computing models incorporating parameter tuning and input variable selection. Neural Comput Appl 31(2):577–592

21. Graves A, Jaitly N (2014) Towards end-to-end speech recognition with recurrent neural networks. In: International conference on machine learning, pp 1764–1772

22. Greff K, Srivastava RK, Koutnik J et al (2016) LSTM: a search space Odyssey. IEEE Trans Neural Netw 28:1–11