Expert Systems with Applications 159 (2020) 113609

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM

Yong'an Zhang, Binbin Yan *, Memon Aasma

School of Economics and Management, Beijing University of Technology, Beijing, China

ARTICLE INFO

Article history:
Received 17 October 2019
Revised 15 April 2020
Accepted 26 May 2020
Available online 30 May 2020

Keywords:
Deep learning
Long short-term memory
Complementary ensemble empirical mode decomposition
Financial time series
Stock market forecasting
Principal component analysis

ABSTRACT

Deep learning is well-known for extracting high-level abstract features from a large amount of raw data without relying on prior knowledge, which is potentially attractive in forecasting financial time series. Long short-term memory (LSTM) networks are deemed as state-of-the-art techniques in sequence learning, which are less commonly applied to financial time series predictions, yet inherently suitable for this domain. We propose a novel methodology of deep learning prediction, and based on this, construct a deep learning hybrid prediction model for stock markets—CEEMD-PCA-LSTM. In this model, complementary ensemble empirical mode decomposition (CEEMD), as a sequence smoothing and decomposition module, can decompose the fluctuations or trends of different scales of time series step by step, generating a series of intrinsic mode functions (IMFs) with different characteristic scales. Then, with retaining the most of information on raw data, PCA reduces dimension of the decomposed IMFs component, eliminating the redundant information and improving prediction response speed. After that, high-level abstract features are separately fed into LSTM networks to predict closing price of the next trading day for each component. Finally, synthesizing the predicted values of individual components is utilized to obtain a final predicted value. The empirical results of six representative stock indices from three types of markets indicate that our proposed model outperforms benchmark models in terms of predictive accuracy, *i.e.*, lower test error and higher directional symmetry. Leveraging key research findings, we perform trading simulations to validate that the proposed model outperforms benchmark models in both absolute profitability performance and risk-adjusted profitability performance. Furthermore, model robustness test unveils the more stable robustness compared to benchmark models.

1. Introduction

Forecasting the future price or return of underlying assets in financial markets is of paramount importance for reducing the risks in decision-making by determining the future movement of assets appropriately. The traditional forecasting of financial time series is typically based solely on economics and finance. In the recent years, it has evolved rapidly to adopt comprehensive ensembles of different disciplines to predict financial markets, which makes financial markets forecasting a unique promising financial research field.

Financial market is a noisy, non-parametric dynamic system (Huang & Tsai, 2009; Cavalcante, Brasileiro, & Souza et al., 2016), which usually has a series of complex and non-linear features. Analysis and forecasting of financial data have always been a challenging issue in the financial domain. Existing analytical prediction methods show certain "discomfort" in varying degrees. For instance, traditional metrology approaches or equations with parameters are inappropriate for analyzing complex, high-dimensional, noisy financial sequences. In addition, conventional artificial networks with nonlinear characteristics are not yet capable of modeling such complex data accurately (Langkvist, Karlsson, & Loutfi, 2014). Meanwhile the performance of traditional machine learning methods depends on artificial features design to a large extent (Arel, Rose, & Karnowski, 2010), which causes certain interference to the predictive performance. Meanwhile, these methods have faced issues in the application process, such as over-fitting and slow convergence. Their departure from the basic settings of traditional financial metrology developed a gap, alienating the financial researchers in terms of application (Dixon, Klabjan, & Bang, 2017).

On the other hand, artificial intelligence (AI) has been a research hotspot to narrow such a gap, which is largely attributed to its

* Corresponding author at: School of Economics and Management, Beijing University of Technology, Beijing 100124, China.

E-mail addresses: bjutzhya@bjut.edu.cn (Y. Zhang), maywind23@163.com (B. Yan).

https://doi.org/10.1016/j.eswa.2020.113609
0957-4174/

spectacular success in natural language processing, image classification, and various time series tasks (Hinton & Salakhutdinov, 2006; Sarikaya, Hinton, & Deoras, 2014; Leong, Hew, Ooi, Lee, & Hew, 2019; Ramirez, Melin, & Prado-Arechiga, 2019). Underlying this process is one of the developments in feature learning framework, referred to as deep learning (LeCun, Bengio, & Hinton, 2015) whose basic structure is described as a multi-layer neural network with powerful computing power, availability of big data sets and more complex and enhanced algorithms (Hinton, Osindero, & Teh, 2006; Krizhevsky, Sutskever, & Hinton, 2017; Salakhutdinov & Hinton, 2009; Deng & Yu, 2014; Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). The deep learning method, proposed by Hinton and Salakhutdinov (2006) and developed into various structures including stack autoencoders, deep belief networks, deep convolutional neural networks and recurrent neural networks, provides a novel perspective for financial markets forecasting. Deep learning considers information as human brain processing data, and strengthens traditional neural networks through a series of hidden layers to improve predictive power (Fehrera & Feuerriegela, 2015). The application of deep learning in finance not only relieves the issues of analysis and prediction in a certain extent, but also brings about the changes of empirical analysis paradigm in finance as well. Therefore, with the advent of the intelligent era and the sharply increasing demand for forecasting financial time series, deep learning has become a promising frontier of application in the financial domain.

Deep learning is widely used in financial market forecasting, such as volatility forecasting, price forecasting and trend (*i.e.* directional movements) forecasting for various financial assets, *i.e.*, stocks, futures, bonds (Makinen, Kanniainen, Gabbouj, & Iosifidis, 2019; Sim, Kim, & Ahn, 2019; Singh & Srivastava, 2017; Zhang, Zohren, & Roberts, 2019). The research focuses on how the various structures of deep learning are designed and adjusted to yield better predictive outcomes. Most scholars evaluate the prediction performances of deep learning in comparison to traditional models, and draw a conclusion that the deep learning method can enhance the predictive accuracy. Xiong, Nicholas, and Shen (2016) incorporated the domestic trends of Google, which can represent public sentiments and macroeconomic factors, with a Long Short-Term Memory model to examine the impact of these factors on the volatility of S&P 500 from October 19, 2004 to July 24, 2015. Shen, Chao, and Zhao (2015) improved the classic deep belief network by using continuous Restricted Boltzmann Machine to build up deep belief network. They revealed that the model is applicable for continuous data, and can be applied to predict the weekly data of GBP/USD, INR/USD and BRL/USD exchange rates. Similarly, Minh, Sadeghi-Niaraki, Huy, Min, and Moon (2018) presented a novel framework that combined a two-stream gated recurrent unit network with sentiment dictionary trained on financial news and Harvard IV-4. The experimental results confirmed a promising predictive power. Wen, Li, Zhang, and Chen (2019) utilized sequence reconstruction method by leveraging motifs to denoise financial temporal series, along with convolutional neural network to capture the spatial features, to predict the stock market trend. The empirical results indicated that the proposed model outperformed other benchmark models with 4%-7% accuracy improvement. Kim and Won (2018) integrated the LSTM network with various GARCH-type models to predict stock price volatility on KOSPI200 index. The results suggested that the proposed model improved predictive performance over benchmark models. Chong, Han, and Park (2017) combined the deep learning networks with three unsupervised feature extraction methods—PCA, autoencoder, and restricted Boltzmann machine—to predict future market behavior. Consequently, applying deep learning for forecasting financial market in part heightened the predictive performance, but not yet bound to obtain high accuracy outcome. However, it has the application advantages over traditional methods that do not inherently produced on financial market data. Therefore, deep learning models in accordance with prediction and analysis of financial time series, need to take into account the financial markets' environment characteristics when it comes to specific financial issues. Additionally, deep learning, as a novel attempt, had been deployed to other perspectives in finance, including precise amelioration of algorithms or strategies for quantitative trading (Chatzis, Siakoulis, Petropoulos, Stavroulakis, & Vlachogiannakis, 2018; Day & Lee, 2016), stock market crisis events forecasting (Huck, 2019) and financial sentiment analysis (Sezer & Ozbayoglu, 2018).

Among deep learning structures, in particular, long short-term memory (LSTM) networks, a representative type of recurrent neural networks (RNN), are suitable for modeling temporal patterns, which is widely utilized in a various of tasks regarding time series issues (Cortez, Carrera, Kim, & Jung, 2018; Jurgovsky et al., 2018; Kim & Cho, 2018; Petersen, Rodrigues, & Pereira, 2019). Experiments results showed LSTM to be superior in performance in comparison to conventional RNN, in large part to surmounting difficulties of gradient vanishing (or explosion). Furthermore, learning long-term dependencies effectively through memory cells and gates facilitates the wide application of LSTM in an array of studies on financial time series modeling. In prior studies based on artificial neural networks, technical indicators of capturing temporal characteristics were calculated and fed into networks as input features (Gocken, Ozcalici, Boru, & Dosdogru, 2016; Kristjanpoller & Hernandez, 2017; M'ng & Mehralizadeh, 2016; Patel, Shah, Thakkar, & Kotecha, 2015; Qiu & Song, 2016). On the contrary, LSTM does not require to select features manually, such as technical indicators; instead, given any observations as the input, it can automatically detect the best patterns that are suitable for dependent data (LeCun et al., 2015). To bridge over the difficulties of overfitting, Baek and Kim (2018) presented ModAugNet framework composed of two modules: an overfitting prevention LSTM module and a prediction LSTM module. It was observed that the proposed model mounted the predictive accuracy in comparison to benchmark model. Fischer and Krauss (2018) utilized LSTM network to forecast the directional movements for constituent stocks of S&P 500. The empirical results demonstrated that LSTM networks outperformed benchmark—memory-free classification models (*i.e.*, a random forest (RAF), a deep neural net (DNN), and a logistic regression classifier (LOG)) and unveiled lower exposure to systematic risk. Bao, Yue, and Rao (2017) adopted ensemble model, LSTM included, to predict stock price, which effectively improved performance of ensemble models. In addition, different types of index data can be fed into networks to expand the information set available. Liu (2019) deployed LSTM RNNs to forecast volatility of stock indices (S&P 500 and AAPL) and achieved a better predictive performance compared to Support Vector Machines (SVM) and GARCH model, which confirmed that LSTM network can extract useful and abstract features from big raw data. Chung and Shin (2018) presented a hybrid model integrating LSTM network with genetic algorithm (GA) and utilized GA to determine the parameters of LSTM network, including time window size and topology. The experimental results confirmed the excellent predictive performance of their proposed model. Li and Tam (2017) proposed a hybrid model that incorporated LSTM with real-time wavelet denoising functions to forecast the stock indices, where a sliding window mechanism was adopted for wavelet denoising. The experimental results unveiled that their proposed model outperformed the LSTM model without wavelet denoising module.

In addition, it is widely accepted that financial market is a nonlinear and non-stationary complex dynamic system, which needs a denoising approach in the phase of preprocessing to financial time series before which are fed into deep learning framework. Generally speaking, wavelet transforms (WT) is extensively regarded as the main denoising approach for financial market as reported by (Hsieh, Hsiao, & Yeh, 2011; Bao et al., 2017). Nevertheless, it needs to specify wavelet basis function artificially, coupled with some drawbacks: ringing near discontinuities, shift variance, the lack of directionality of decomposition functions and some others (Sović & Seršić, 2012). Instead, Huang, Shen, Long, Wu, Shih, Zheng, and Liu (1998) proposed the empirical mode decomposition (EMD), a method that is effective for dealing with non-linear and non-stationary time series. Any complex signal given can be decomposed into a limited number of intrinsic mode functions (IMF) by EMD, each IMF representing the component of original signal characteristic scale. These components form a complete and nearly orthogonal basis for the original signal and EMD is adaptive and highly efficient, without the issue of pre-selecting basis functions like wavelet transform. Wang and Wang (2017) incorporated EMD with stochastic time strength neural network (STNN) to forecast the stock price fluctuation, and empirical results demonstrated a better performance than benchmark models, with a new evaluated method (q-order multiscale complexity invariant distance) applied. Bisoi, Dash, and Parida (2019) proposed a hybrid model that integrated Robust Kernel based Extreme Learning Machine (RKELM) with VMD, *i.e.*, a variant of EMD that evaded a limitation of EMD, which was experimentally proved to be superior in both one-step-ahead stock price prediction and daily trend prediction. Awajan, Ismail, and Al Wadi (2018) presented a stock market forecasting model that combined EMD with Holt-Winter. The experimental results exhibited that EMD-HW bagging outperformed fourteen selected methods. Furthermore, several quite efficient optimization algorithms which incorporated EMD or its variants with neural networks had proved efficiency that in turn boosted predictive performances as well (Awajan & Ismail, 2017; Li, 2017; Wei, 2016; Xiu & Chen, 2017; Yang et al., 2018; Zhang, Lin, & Shang, 2017).

However, there is a major issue with the EMD method: modal aliasing, *i.e.*, signals of different scales or frequencies appear in the same IMF component, or signals of the same scale or frequency are decomposed into a plurality of different IMF components. In response to this problem, Wu et al. (2009) proposed Ensemble Empirical Mode Decomposition (EEMD), an improved variant of EMD that adds different white noise to the original signal for multiple times, then conducts EMD decomposition, and averages the results of multiple decompositions to obtain the final IMF components. Based on EEMD, Yeh, Shieh, and Huang (2010) proposed Complementary Ensemble Empirical Mode Decomposition (CEEMD), a more sophisticated method that adds a pair of opposite white noise signals to the original signal and conducts EMD decomposition separately. Consequently, CEEMD reduced the reconstruction error caused by white noise while ensuring that the decomposition effect is at least equivalent to that of EEMD.

In light of the above-mentioned literatures, we find that the existing literatures have limitations. Firstly, it is a novel attempt to introduce deep learning into the research of financial market prediction, nevertheless lacking of widely accepted sophisticated research paradigm that incorporates deep learning with nonlinear and non-stationary signal processing methods. Secondly, WT can deal with financial time series to some extent, but it needs to set wavelet basis function manually. Apart from these, it is obvious that hybrid prediction model can outperform individual model of the ensemble in predictive accuracy, when the individual model is sufficiently diverse and accurate enough, as reported by Krogh and Vedelsby (1995). As such, to improve hybrid methods with various optimal algorithms and advanced network frameworks is the development trend of deep learning for financial time series.

In this paper, we present a novel deep learning theoretical framework for financial time series forecasting, on which constructing a CEEMD-PCA-LSTM model is based. The model comprehensively integrates the advantages of CEEMD, PCA, LSTM such that it can effectively improve the predictive accuracy of stock indices. Firstly, as the data-preprocessing module, CEEMD is utilized to decompose the financial time series to eliminate noise. Secondly, PCA, as the dimensionality reduction module, is adopted to extract abstract and deep features and to significantly save training time of deep learning. Thirdly, as the forecasting module, LTSM is applied to generate a one-step-ahead predictive output for IMF components. Finally, as the predictive synthesis module, all the predictive components are synthesized to obtain the final predicted result. In order to validate proposed model, we select six representative indices from three different developed stock markets to evaluate predictive accuracy. For instance, as is the most representative developed stock market, the New York Stock Exchange has a mature and sophisticated operation system, with Dow Jones Index and S&P 500 Index chosen as target index for developed markets. Then we evaluate the performance of proposed model from two dimensions—predictive accuracy and profitability performance (*i.e.* absolute return performance and risk-adjusted return performance). First, the test error indicators, RMSE, MAE and NMSE, mean the deviation of predicted stock indices from observed ones. In addition, Directional Symmetry (DS) is chosen to assess the consistency between predicted and observed in terms of directional movement of return. Second, we conduct trading simulation to evaluate absolute profitability performance and risk-adjusted profitability performance. To better capture the performance of CEEMD-PCA-LSTM, we deploy other four models as the comparisons against our proposed model. The reference models embody RNN, LSTM, EMD-PCA-LSTM, EEMD-PCA-LSTM, where the former two models are utilized to test the effectiveness and superiority of hybrid models and latter two models to test those of CEEMD. Finally, the model robustness test is to check the robustness of model as its parameter varies.

The remainder of this paper is organized as follows. Section 2 presents the deep learning forecasting methodology with theoretical framework composed of four modules. The deep learning hybrid models, coupled with CEEMD, PCA, LSTM, are introduced in Section 3. Section 4 briefly covers the experiment design, including data chosen and prediction procedure. Then, the experimental results (comparative analysis included) of predictive accuracy and trading simulation are elaborated in Section 5 and 6, respectively. Section 7 conducts the model robustness test. Finally, our conclusions with suggestion for the future research are presented in Section 8.

2. Methodology

With the increasing availability of trading data and the popularity of deep learning, coupled with the unsatisfactory performance of existing models, we objectively construct a theoretical framework of deep learning prediction model for financial time series, which is feasible, effective, and superior. Without relying on econometric assumptions and expert experience, our research can extract high-level abstract features from data that in turn identify hidden nonlinear relationships as well, as an "upgraded version" of existing models and methods. Specifically, we analyze its prediction mechanism in detail, which is in the perspective of Decomposition-Reconstruction-Synthesis with respect to nonlinear, nonstationary and multi-scale complex financial time series.

Our framework is composed of four modules, as illustrated in Fig. 1. First, with financial time series having been preprocessed, sequence smoothing processing and decomposition module (e.g., CEEMD, WT) smooths and denoises the complex signal. A quintessential example should be cited that CEEMD can decom-

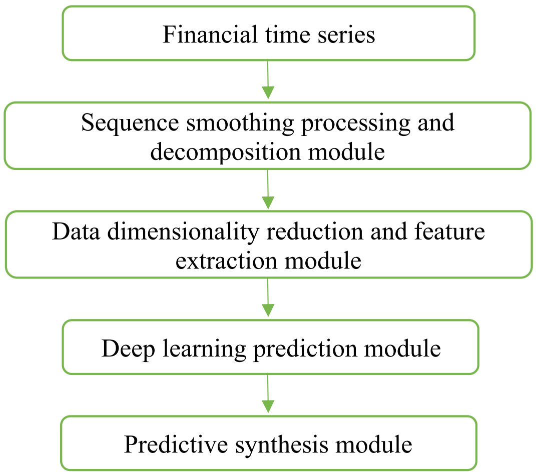

Fig. 1. The theoretical framework of deep learning prediction.

pose the fluctuations or trends of different scales in the time series step by step, yielding a series of intrinsic mode functions (IMF) with different characteristics scales. Second, data dimensionality reduction and feature extraction module: retaining most of the information of the raw data, it can reduce dimensionality of decomposed IMF components, remove redundant information and improve prediction response speed. Third, deep learning prediction module: conventional forecasting method, solely based on econometrics, typically cannot accurately characterize the intricate relationship between stock fluctuations and influencing factors, while deep learning has advantages, such as high nonlinear computing and mapping ability, self-learning and self-organizing ability, high-speed computing ability. Specifically, high-level abstract features are separately fed into LSTM network to predict target stock price, yielding a series of predictive values of individual components. Finally, predictive synthesis module synthesizes all the predicted values of individual components to form a final predicted value.

3. Model

To generate high-level and abstract features for financial time series, this section presents deep learning hybrid models based on the methodology described above, and briefly introduces the hybrid models' components, *i.e.*, EMD family models, PCA and LSTM.

3.1. EMD family models

3.1.1. EMD (Empirical mode decomposition)

The EMD algorithm is based on the assumption that any signal consists of a series of intrinsic mode functions (IMFs), whose amplitude and phase vary with time. Each IMF must satisfy two conditions: a) in the whole dataset, the number of extrema and the number of zero-crossings must either be equal or differ no more than by one; b) the mean value of upper and lower envelopes determined by the maximum and minimum values in the signal is zero, *i.e.*, the upper and lower envelopes are locally symmetrical with respect to the zero axis.

EMD can screen the IMFs from original financial time series$S(t)$ with multiple components step by step, and the specific decomposition steps are as follows:

(1) Determine all maxima and minima in the original time series $S(t)$;
(2) Cubic spline interpolation method or piecewise cubic Hermite interpolating polynomial method is adopted to construct the upper and lower envelopes of $S(t)$ according to the maximum value and the minimum value;
(3) Calculate the local mean $m_{1j}(t)$ of $S(t)$ from the upper and lower envelopes, and subtract $m_{1j}(t)$ from $S(t)$ to obtain a new series $h_{1j}(t)$;
(4) Substitute $h_{1j}(t)$ for the original time series $S(t)$, and repeat steps (1) to (3) above until the variance between $h_{1j}(t)$ and $h_{1(j-1)}(t)$ is less than the set threshold, then the first IMF component $imf_1(t)$ is $h_{1j}(t)$, denoted as $imf_1(t) = h_{1j}(t)$;
(5) $imf_1(t)$ is separated from the original series to get a difference series $r_1(t) = S(t) - imf_1(t)$, dispensing with high frequency components;
(6) Replace $S(t)$ with $r_1(t)$, and repeat steps (1) to (5) above until the termination condition is satisfied (conventionally such that the last residue satisfies monotonicity), which represents that the entire EMD screening process ends. Overall, the decomposition result of $S(t)$ can be denoted as:

$$S(t) = \sum_{j=1}^{K} imf_j(t) + res \tag{1}$$

where *res* is the trend term, $res = R(t)$, representing the trend of original time series $S(t)$, and $imf_j(t)$ is the *jth* IMF (K is the total number of IMFs of EMD).

However, EMD has one major problem that is referred to as modal aliasing, where signals of different scales or frequencies appear in the same IMF component, or where signals of the same scale or frequency are decomposed into multiple different IMF components.

3.1.2. EEMD (Ensemble empirical mode decomposition)

Noise-assisted analysis can effectively suppress the modal aliasing of EMD method. According to the characteristics of frequency-average distribution of Gaussian white noise, the signal has continuity at different scales after Gaussian white noise added, which can reduce the degree of modal aliasing (Wu & Huang, 2009). The specific steps of EEMD are as follows:

(1) Add different Gaussian white noise $n_i(t)$ to the original signal, where the mean is 0 and the standard deviation is constant (usually 0.1 to 0.4 times the standard deviation of the original signal), and $x_i(t)$ can be formulated as:

$$x_i(t) = S(t) + n_i(t) \tag{2}$$

where $x_i(t)$ denotes the signal $S(t)$added with Gaussian white noise $n_i(t)$for the *ith* time.

(2) Perform EMD decomposition on the target signal to obtain IMF components and a trend term *res*;
(3) Repeat steps (1) ~ (2) above for N times;
(4) The corresponding IMF components are averaged to eliminate the influence of multiple added Gaussian white noise on the real IMF. Finally, the *jth* IMF, *i.e.*, imf_j, can be denoted by:

$$imf_j = \frac{1}{N}\sum_{i=1}^{N} imf_{ij} \tag{3}$$

where imf_{ij} represents the *jth* IMF for the *ith* time. The larger the N is, the closer the sum of Gaussian white noise added in IMF is to 0, and the decomposition result, $x(t)$, of EMMD can be expressed as:

$$x(t) = \sum_{j=1}^{K} imf_j(t) + res \tag{4}$$

3.1.3. CEEMD (Complementary ensemble empirical mode decomposition)

CEEMD is an improved algorithm based on EMD and EEMD. Yeh et al. (2010) made further improvements to EEMD by adding random Gaussian white noise to the original signal in positive and negative pairs. This enabled a better elimination effect on the residual auxiliary noise in reconstructed signal as indicated by experience, while making the final decomposition more complete. The steps for CEEMD decomposition are as follows:

(1) By adding N pairs of positive and negative Gaussian white noise to the original signal, $2N$ signal sets are denoted by:

$$\begin{bmatrix} M_1 \\ M_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} S \\ n \end{bmatrix} \tag{5}$$

where S is the original signal, n is the Gaussian white noise, and M_1, M_2 are the sum of original signals with the positive and negative Gaussian white noise, respectively.

(2) EMD decomposition is performed on the target signal, and each signal obtains a set of IMF components, in which the *ith* component of the *jth* IMF is expressed as imf_{ij};
(3) Obtain the results of each IMF after averaging the overall ensemble, which can be formulated by:

$$imf_j = \frac{1}{2N} \sum_{i=1}^{2N} imf_{ij} \tag{6}$$

hence, the final decomposition result, $x(t)$, of CEEMD can be denoted as:

$$x(t) = \sum_{j=1}^{K} imf_j(t) + res \tag{7}$$

In this paper, the parameters for EMD family—max modes (*i.e.*, the number of IMF components) and max iteration are set to 11 and 10000, respectively. Since the financial times series are not smooth, "pchip" is specified as the interpolation method, *i.e.*, piecewise cubic Hermite interpolating polynomial method. For EEMD and CEEMD, particularly, the number (N) and standard deviation of Gaussian white noise are specified as 400 and the standard deviation of the original time series multiplied by 0.2, respectively.

3.2. PCA (Principal component Analysis)

PCA is typically applied to reduce dimensionality of raw dataset, while maintaining the features of the largest variance contribution in the dataset. Thus, it removes noise, eliminates unimportant features, and improves the processing speed of dataset.

In PCA, the extraction sequence $u \in \mathbb{R}^d$ is generated from the raw data $x \in \mathbb{R}^D$ via linear transformation function Φ, which can be formulated as:

$$u = \Phi(x) = Wx + b \tag{8}$$

where $W \in \mathbb{R}^{d \times D}$, $WW^T = I$, $b \in \mathbb{R}^d$, and $u = \Phi(x)$ is called a representation of x, while each element of u is called a feature. The rows of W denote eigenvectors corresponding to the first d largest eigenvalues, and b is usually set to $-WE(x)$, therefore, $E(u) = 0$. On the contrary, we define a reverse map, $\psi : u \to x$. Hence, the raw data is reconstructed by PCA data, and reconstruction data x_{rec} can be denoted as:

$$x_{rec} = \psi(u) = W^T(u - b) \tag{9}$$

3.3. LSTM (Long Short-Term Memory)

LSTM is a representative of deep recurrent neural networks that are particularly expert in processing sequential data. The clever idea of introducing self-loops to produce paths where the gradient can flow for long durations is a core contribution of the initial LSTM model (Hochreiter & Schmidhuber, 1997). Furthermore, A crucial addition has been to make the weight on this self-loop conditioned on the context, rather than fixed (Gers, Schmidhuber, & Cummins, 2000). LSTM network realizes temporal memory function through switch of the gate, and can effectively solve the problem of gradient vanishing and explosion in recurrent neural network.

The key to LSTM is the introduction of a gating unit system that stores historical information through the internal memory unit—cell state unit, using different "gates" to let the network learn dynamically, *i.e.*, when to forget historical information or update cell state with new information.

Fig. 2 is a block diagram of the LSTM neural network "cell". At time t, the internal memory unit records all historical information up to the current time and is controlled by three "gates":

(1) Input Gate: Input gate $g_i^{(t)}$ (for time step t and cell i, similarly in other symbols) controls new information fed into internal memory unit, which can be denoted as:

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right) \tag{10}$$

where the activation function σ is the sigmoid function, $x^{(t)}$ is the current input vector, $h^{(t)}$ is the current hidden layer vector, containing the output of all LSTM cells, and b^g, U^g and W^g indicate bias, input weights, and recurrent weights for input gate, respectively.

(2) Forget Gate: Forget gate unit $f_i^{(t)}$ controls internal memory unit on how much information in the previous moments needs to be saved.

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right) \tag{11}$$

where the activation function σ is the sigmoid function with a value between 0 and 1, b^f, U^f, W^f are bias, input weights, and recurrent weights for forget gate, respectively. When $f_i^{(t)}$ = 1; the forget gate is opened, and cell state at the previous time is fed into cell. Otherwise, when $f_i^{(t)}$ = 0; the forget gate is closed, and cell state at the previous time is discarded.

The internal state unit $S_i^{(t)}$ of the LSTM cell that has a conditional self-loop weight $f_i^{(t)}$ is updated, which is formulated as:

$$S_i^{(t)} = f_i^{(t)} S_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right) \tag{12}$$

where b, U, W denote bias, input weights, and recurrent weights into LSTM cell, respectively. On the right side of equation (12), the former part is the information of cell state controlled by the forget gate at the last moment, and the latter part is the input information controlled by the input gate.

(3) Output Gate: Output gate $O_i^{(t)}$ controls internal memory unit, *i.e.*, yields and produces required information, which can be given by:

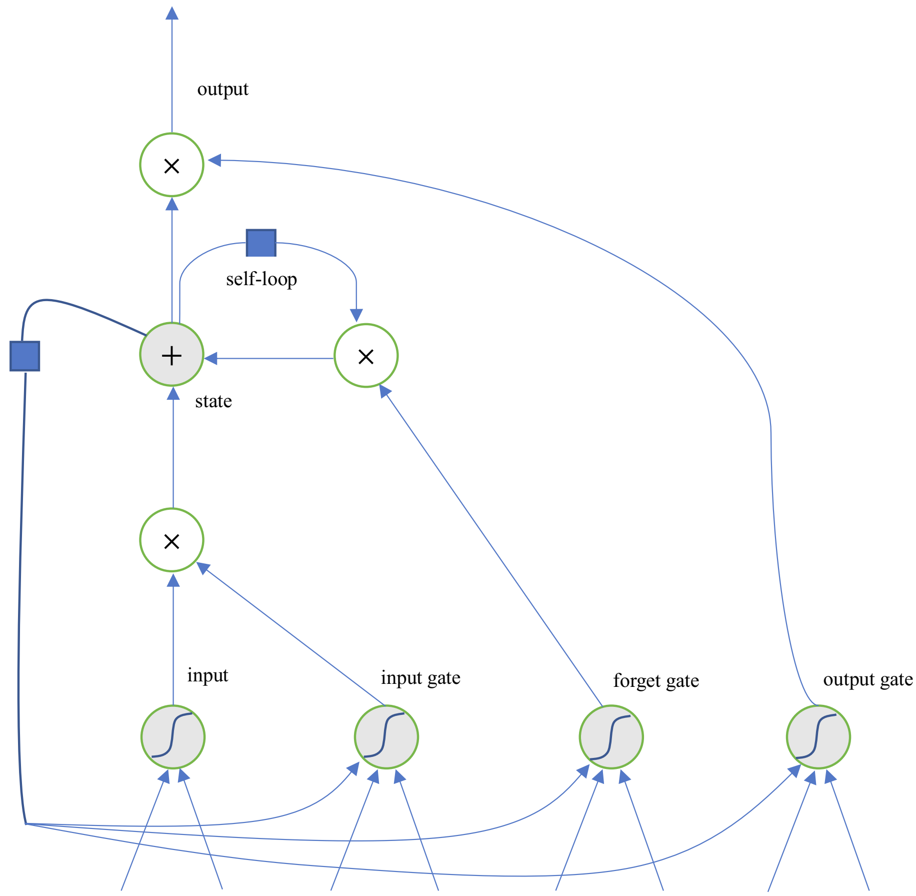

Fig. 2. LSTM neural network "cell" block diagram. Cells are recurrently connected to each other, replacing the common hidden units in general recurrent networks. The input features are calculated here with artificial neuron units. Their value can be accumulated into the state when the sigmoidal input gate allows it. The state unit has a linear self-loop with its weight controlled by the forget gate. The output of the cell can be shut off by the output gate. All the gating units have sigmoid nonlinearity, while the input unit can have any compression nonlinearity. The state unit can also be used as an additional input to the gating unit. The blue square represents the delay of a single time step.

$$h_i^{(t)} = \tanh\left(S_i^{(t)}\right) O_i^{(t)} \tag{13}$$

$$O_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right) \tag{14}$$

where b^o, U^o, W^o denote bias, input weights, and recurrent weights for output gate, respectively. The output of LSTM unit acts as a hidden layer state in the RNN.

For the time being, there is no rule of thumb to specify the number of hidden layers and delays (Palangi, Deng, et al., 2016; Palangi, Ward, & Deng, 2016), which plays a vital role affecting the overall performance of training and testing in the architecture of LSTM network. Specifically, through trial and error, the number of hidden layers and delays are set to 200 and 4, respectively. Moreover, the solver and max epochs are set to "adam" and 1000 epochs, respectively. To prevent gradient from exploding, the gradient threshold is set to 1. Initial learning rate is specified as 0.005, and the learning rate is dropped by multiplying by a factor of 0.2 after 125 iterations. The financial time series data set is divided into three subsets: the training set, the validating set, and the test set, with a set ratio of 7:1:1.

3.4. Deep learning hybrid prediction model

Based on the theoretical framework, we construct a novel deep learning hybrid prediction model, *i.e.*, CEEMD-PCA-LSTM, which extracts deep and abstract features and then is applied to one-step-ahead stock price forecasting. The deep learning hybrid prediction model integrates EMD or its variants with PCA and LSTM. Fig. 3 illustrates the flow chart of the framework in detail that consists of four phases: (1) utilizing CEEMD for data preprocessing, which is applied to decompose financial time series to eliminate noise; (2) adopting PCA to reduce data dimensionality, extract abstract and advanced features and improve computational efficiency; (3) generating a one-step-ahead predictive output for each component using LSTM with delay; and (4) synthesizing all the predictive components to obtain the final predicted value. The detailed LSTM prediction module is further detailed in Fig. 4.

4. Experiment design and data selection

4.1. Financial markets classification and selection

The maturity of financial markets may affect the effectiveness of deep learning predictions. Therefore, we select three types of financial market samples to verify the feasibility, effectiveness and superiority of deep learning model. We select six stock indices, namely, Shanghai Composite Index (000001.SH), SZSE Component Index (399001.SZ), GEM (399006.SZ), Hang Seng Index (HSI.HI), Dow Jones Industrial Average Index (DJI.GI) and S&P 500 Index (SPX.GI). The New York Stock Exchange has its unique developed market, mature market operation system and rich experience, which is generally regarded as the most advanced financial market

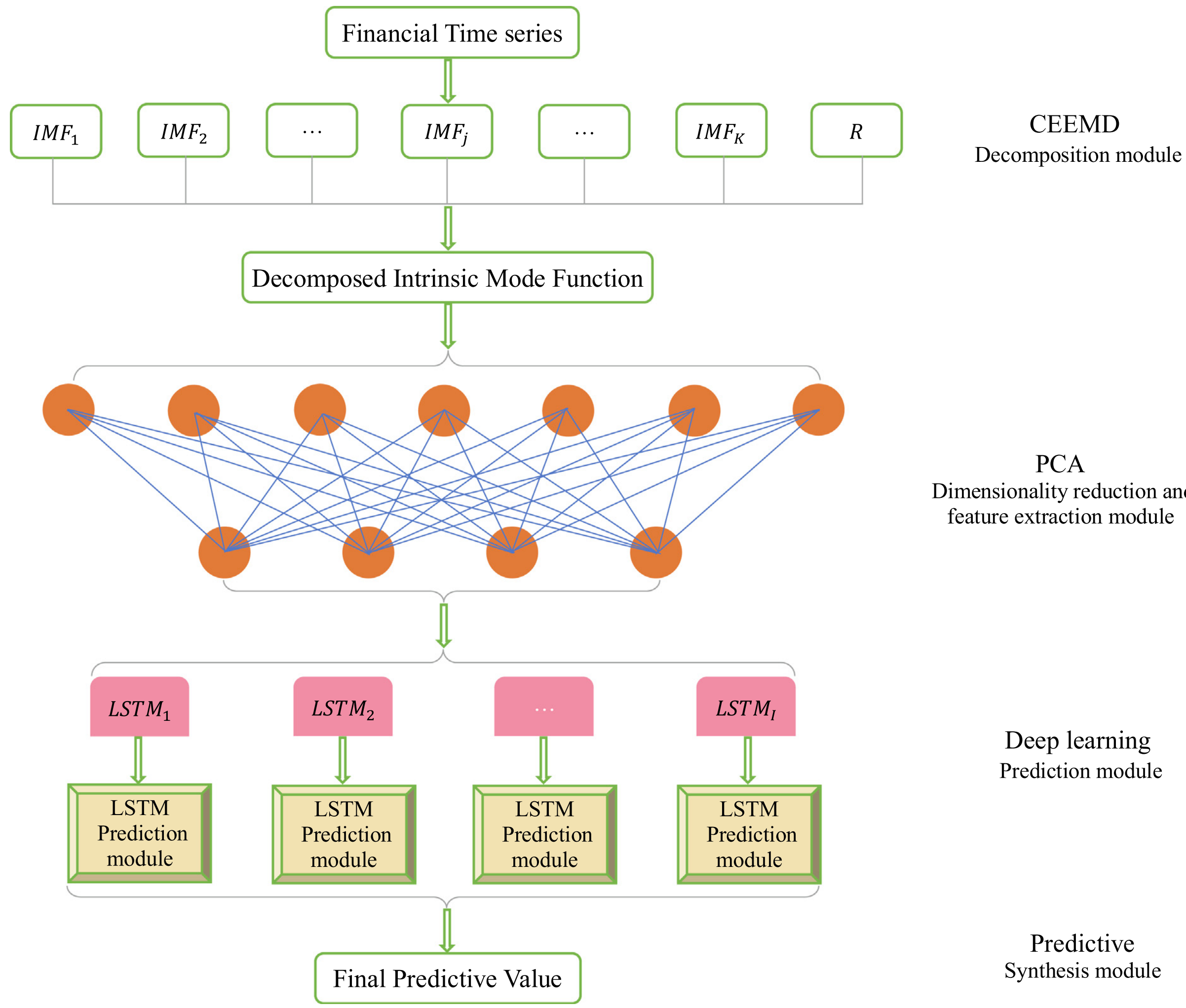

Fig. 3. Flow chart of deep learning hybrid prediction model. IMF_j, R denote the *jth* IMF component and residual term, respectively; $LSTM_i$ is the input of *ith* predictive component, whose total number is the dimension of PCA output.

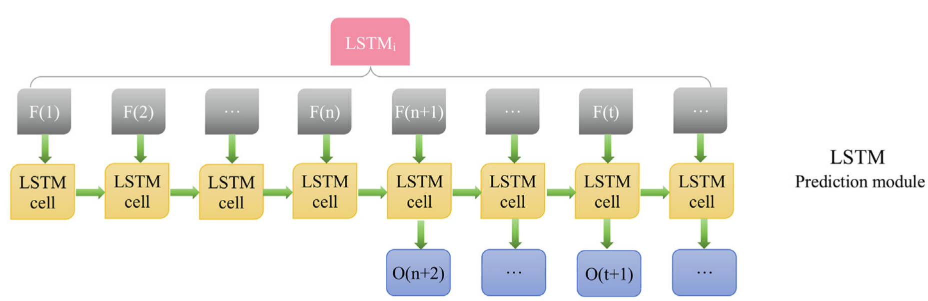

Fig. 4. LSTM prediction module. F_t, O_t represent the denoised components features and the one-step-ahead predictive output at time step t, respectively.

over the world. Specifically, Dow Jones Index and S&P 500 Index are the two most representative stock indices in its market, which are classified in developed markets (or mature markets). On the contrary, operating system of financial markets in mainland China (Shanghai Stock Exchange and Shenzhen Stock Exchange) is not yet perfect enough, so they are classified into developing markets (or emerging markets). We employ Shanghai Composite Index, SZSE Component Index and GEM to represent the developing market. In addition to the above markets, Hang Seng Index represents the Hong Kong market situation between developed and developing markets, which is relatively underdeveloped in comparison to the US stock market and relatively mature to emerging markets. Therefore, Hang Seng Index belongs to relatively developed markets. Accordingly, robustness of the model is tested by financial markets at different degrees of development represented by six stock indices.

4.2. Experiment data

In order to verify the validity, feasibility and superiority of CEEMD-PCA-LSTM model, the experiment data selected in this paper is detailed as follows:

(1) The daily closing prices of Shanghai Composite Index (000001.SH), SZSE Component Index (399001.SZ) and GEM Index (399006.SZ) are selected and processed respectively to obtain logarithmic return r_t, which can be formulated as:

$$r_t = lnP_t - lnP_{t-1} \tag{15}$$

where P_t is the closing price of day t.

(2) Furthermore, the Hang Seng Index (HSI. HI) in Hong Kong Stock Market and the Dow Jones Index (DJI. GI) and S&P 500 Index (SPX. GI) in New York Stock Market are selected and processed to obtain the logarithmic return.

The time span of each stock index runs from January 6th, 2010 to April 27th, 2018. All sample data is collected from Wind database (http://www.wind.com.cn) provided by Shanghai Wind Information Co., Ltd.

4.3. Prediction procedure

In the process of simulating reality investment (*i.e.*, backtesting), we employ prediction approach referred to as forward rolling window (Nair, Mohandas, & Sakthivel, 2010). The prediction procedure refers to the segmentation prediction method described by M'ng & Mehralizadeh (2016), which consists of three parts. First,

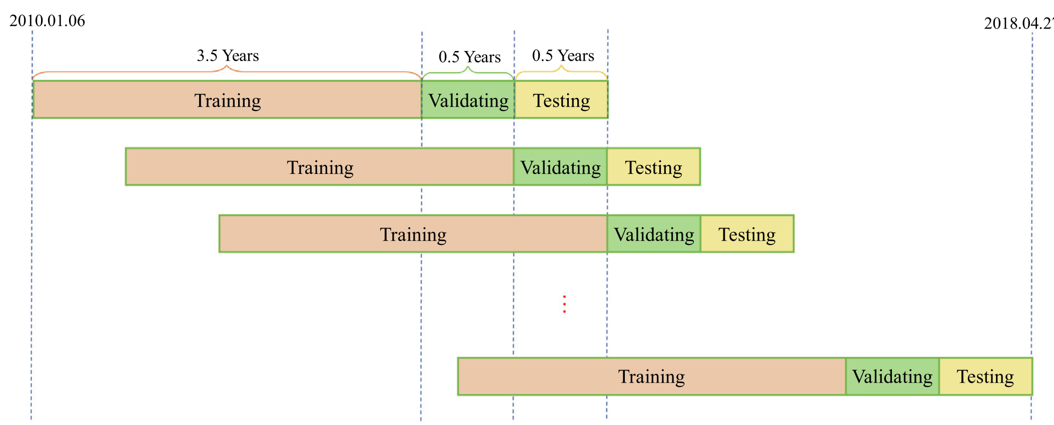

Fig. 5. Training set, validating set and test set distributions of dataset for the whole sample period.

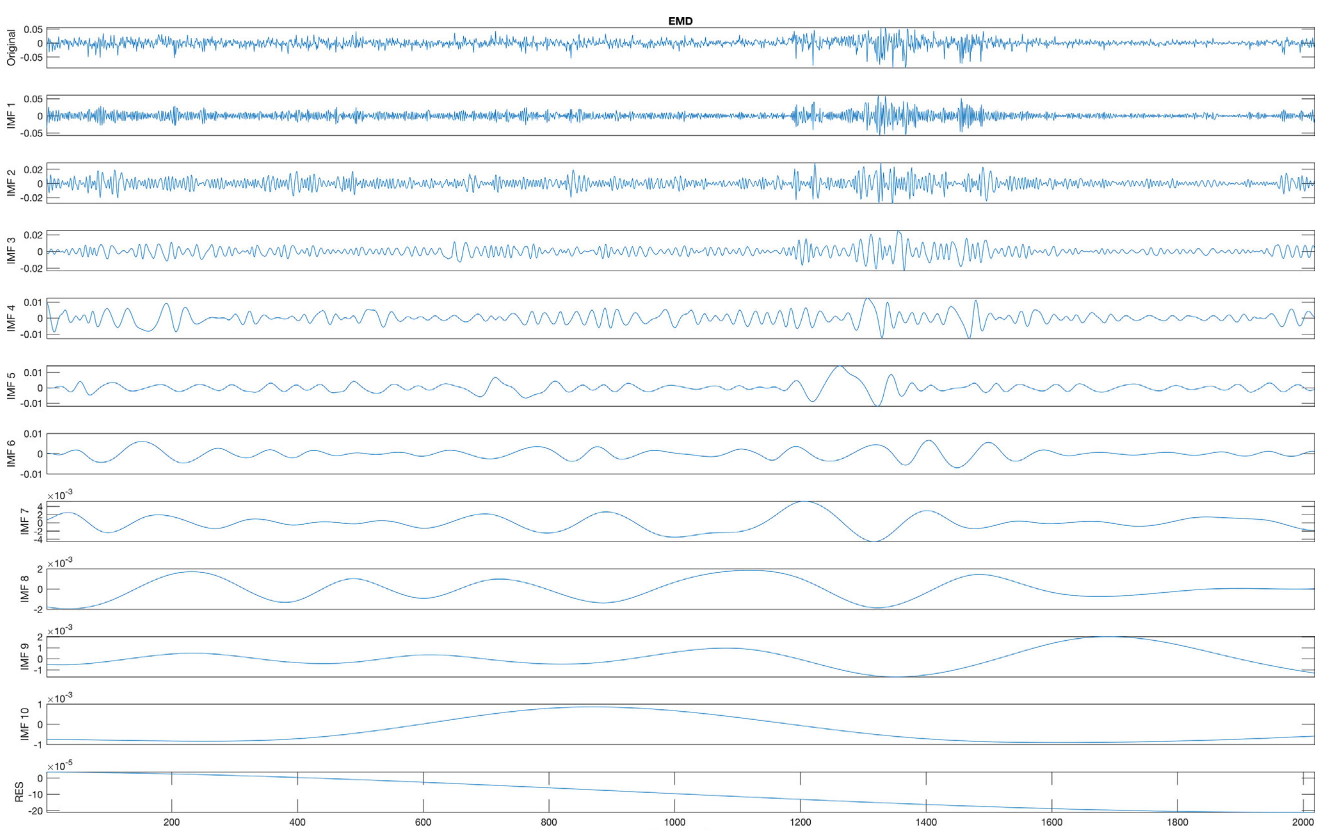

Fig. 6. EMD, EEMD, and CEEMD decomposition results of return of Shanghai Composite Index.

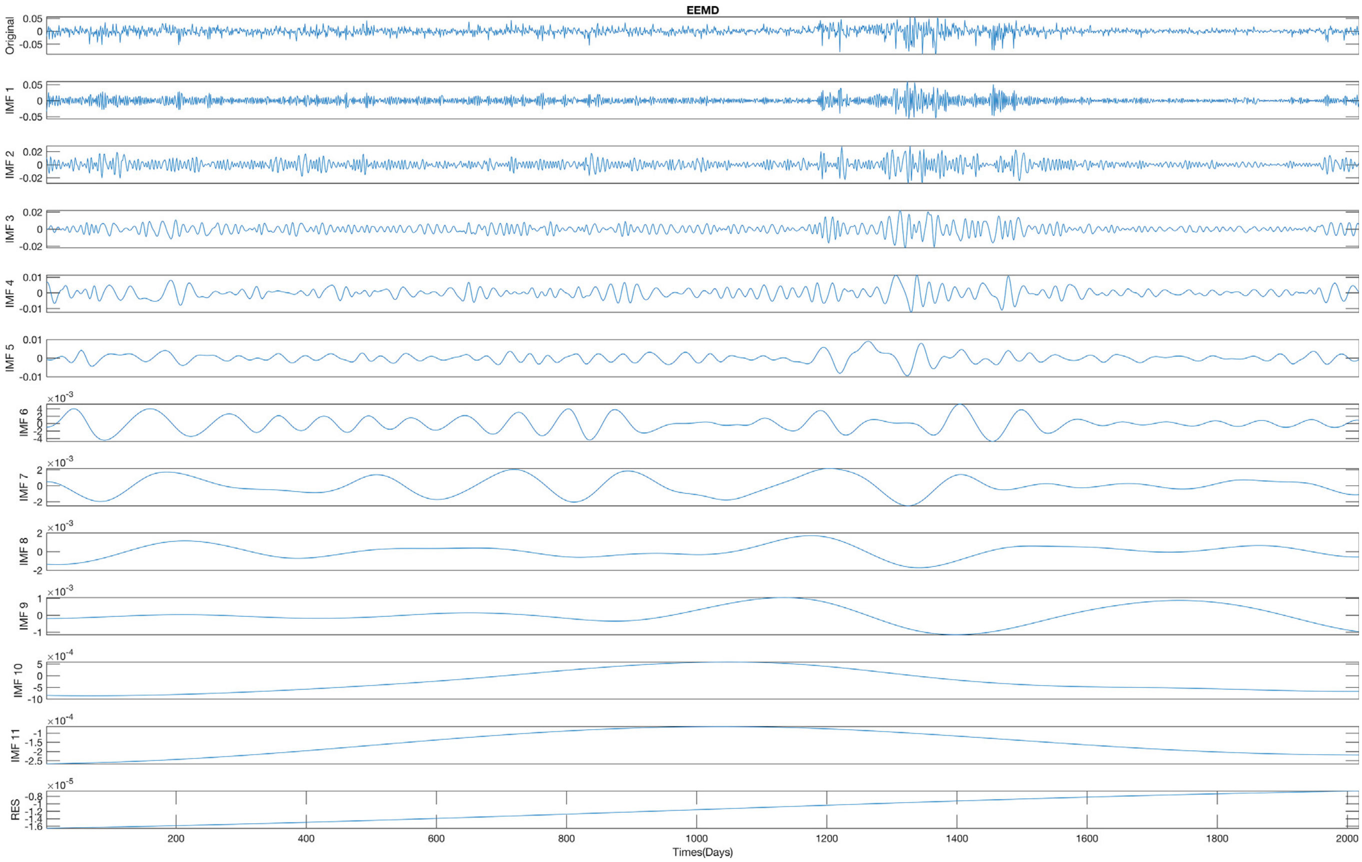

Fig. 6 (*continued*)

for the training part, we select the dataset length of three and a half years, which is used to train model and update model parameters. Second, for validating part, dataset length is selected as half a year, which is utilized to fine-tuning the hyperparameters to obtain a best configuration for the model. Finally, for the test part, the dataset length is chosen as half a year, thus enabling optimal model to predict time series. Consequently, the ratio of training set, validating set, and test set of one cycle is taken as 7:1:1, and the specific details are illustrated in Fig. 5.

5. Empirical results and analysis

In this section, the evaluation indicators for predictive performance are employed to analyze empirical intermediate results (namely results of CEEMD, PCA, and LSTM) and final prediction results of deep learning hybrid prediction model. For the convenience and simplicity of our research, the results of the intermediate results regarding CEEMD and PCA in this paper only take Shanghai Composite Index as an example.

5.1. Evaluation indicators for predictive performance

In this paper, RMSE, MAE and NMSE are selected to evaluate the error between predicted value and observed value. Moreover, DS, namely hit rate, is chosen to measure the accuracy of directional movement of stock. Henceforth, for the convenience, the logarithm return of stock indices are denoted by return throughout the paper.

(1) RMSE (Root Mean Squared Error)

Given a set of return observations of financial time series and the corresponding predicted values, RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(r_{t+1} - \widehat{r}_{t+1}\right)^2} \tag{16}$$

where r_{t+1} and $\widehat{r}_{t+1}$ are the same as r_{t+1} and $\widehat{r}_{t+1}$ in the equations (17)-(19), respectively denoting the observed return and the predicted return at the time of day $t+1$.

(2) MAE (Mean Absolute Error)

MAE is the mean of absolute values of deviation of the arithmetic mean from all individual observations, which can be defined as:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}\left|r_{t+1} - \widehat{r}_{t+1}\right| \tag{17}$$

(3) NMSE (Normalized Mean Squared Error)

NMSE is an estimate of total deviation between predicted value and measured value, which can be defined as:

$$NMSE = \frac{1}{N}\frac{\sum_{t=1}^{N}\left(r_{t+1} - \widehat{r}_{t+1}\right)^2}{var(r_{t+1})} \tag{18}$$

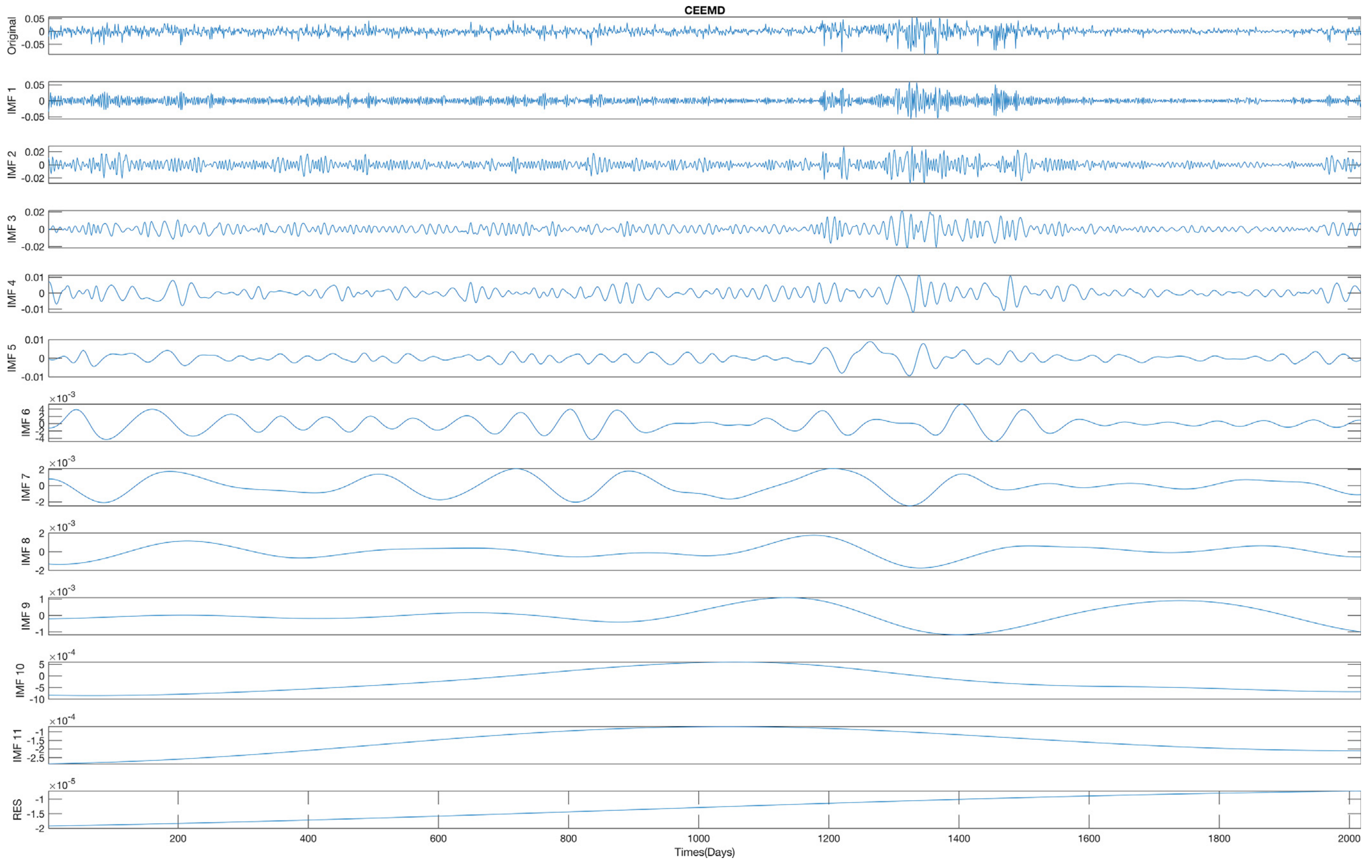

Fig. 6 (*continued*)

(4) DS (Directional Symmetry)

DS is a statistical measure of a model's performance in predicting the direction of change, positive or negative, of a time series from one time period to the next, which assesses the consistency between predicted and observed value in the direction of return movement and can be defined as:

$$DS = \frac{1}{N}\sum_{i}^{n}\mu_i$$

$$\text{where}\quad \mu_i = \begin{cases} 1, & \left(r_{t+1}^n - r_t^n\right)\left(\hat{r}_{t+1}^n - \hat{r}_t^n\right) > 0 \\ 0, & \text{others} \end{cases} \tag{19}$$

5.2. Decomposition results and analysis of CEEMD part

A series of IMF components and a trend term are obtained, which are illustrated in Fig. 6, after sequence smoothing processing and decomposition module (*i.e.*, EMD, EEMD, CEEMD) for return of Shanghai Composite Index. In the perspective of CEEMD results, each IMF component reflects the fluctuation characteristics of different frequencies of investors. Specifically, the high frequency IMF component demonstrates imbalance of short-term market and represents the mood fluctuation characteristics of short-term investors. On the contrary, the low-frequency IMF component reveals volatility characteristics of medium-term market and is mainly affected by policies and regulations. In addition, the trend term *res* indicates the long-term fluctuation characteristics of stock market, representing the overall trend of investor sentiments. As is shown in the last graph of Fig. 6 in terms of *res*, Shanghai Composite Index continues to maintain the momentum of steady growth, coinciding with the fact that continued stable growth is scored in Chinese economy since global financial crisis.

In comparison with EEMD, CEEMD exhibits the similar decomposition result. By contrast, EMD demonstrates a significantly different decomposition result, where EMD has satisfied the terminating conditions (last residue satisfies monotonicity) for the decomposition in advance after 10 IMFs obtained and the trend term shows the opposite trend compared to EEMD and CEEMD. This is largely due to modal aliasing and incomplete decomposition, which can be addressed by CEEMD perfectly.

5.3. Results and analysis of PCA

PCA transforms high-dimensional correlational input vectors into low-dimensional uncorrelated principal components. What is more, extracting the first few principal components can almost

Table 1
Cumulative contribution rate of Shanghai Composite Index after PCA.

Number of principal components	Cumulative contribution rate (%)
1	59.29081332
2	78.45224105
3	90.06017875
4	94.8133505
5	97.51199089
6	98.98049677
7	99.63406147
8	99.8492953
9	99.95074365
10	99.99984157
11	99.99998588
12	100

ensure the validity and completeness of original data. It can also achieve dimensionality reduction and reduce training time of deep learning that in turn improves operating efficiency of model and saves operating costs. Table 1 demonstrates the cumulative contribution rate of Shanghai Composite Index in terms of PCA, where cumulative contribution rate of the first three principal components exceeds 90% after PCA for IMF sequences decomposed by CEEMD. In order to gain better predictive performance of model, we take the first four principal components (cumulative contribution rate as high as 94.81%) as the training sample data fed into LSTM.

5.4. Return prediction results and analysis

First, we present the final prediction results of Shanghai Composite Index, as illustrated in Fig. 7. Where the observed return fluctuates less, the predicted result is highly consistent with the observed one; while where the observed return fluctuates largely, the predicted result is somewhat less consistent with the observed one, which is in line with the fact that the greater the fluctuation, the lower the predictive accuracy under real-life financial environments. In general, our proposed model, *i.e.*, CEEMD-PCA-LSTM, yields excellent predictive performance in all the three markets. Additionally, as visualized from the stem diagram in Fig. 8, RMSE in return prediction of Shanghai Composite Index oscillates up and down around the zero axis, and is substantially locally symmetrical concerning the zero axis.

Moreover, we exhibit the other dimension of predicted stock indices: prices based on the predicted logarithmic return, as is denoted by equation (15). According to Fig. 9, we can find that the predicted stock indices prices of RNN and LSTM have larger volatility and prediction error than those of deep learning hybrid model. In contrast, CEEMD-PCA-LSTM reveals better predictive performance and is closer to the observed value of stock indices in comparison to EEMD-PCA-LSTM and EMD-PCA-LSTM.

To evaluate the advantage of predictive performance of the proposed model, we test the predictive error (*i.e.*, RMSE, MAE, NMSE) and DS of CEEMD-PCA-LSTM compared with RNN, LSTM, EMD-PCA-LSTM, and EEMD-PCA-LSTM models. It can be observed from Table 2 that CEEMD-PCA-LSTM yields the smallest RMSE with an average of 0.0127, which is decreased by 76.91% when compared to RNN. RMSE of the large-cap stock index is smaller than that of a specific industry & sector index. This is because the large-cap stock index comprehensively measures and reflects the overall price level of stock market and its directional trend, with an outcome of relatively smaller fluctuation and forecasting error. On

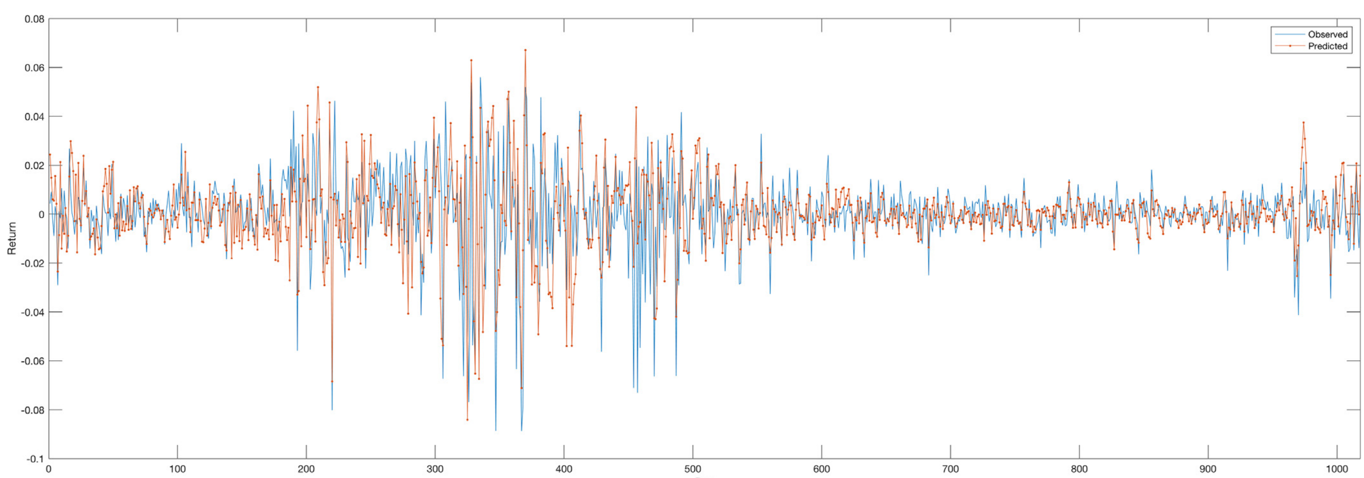

Fig. 7. Prediction results of return of Shanghai Composite Index.

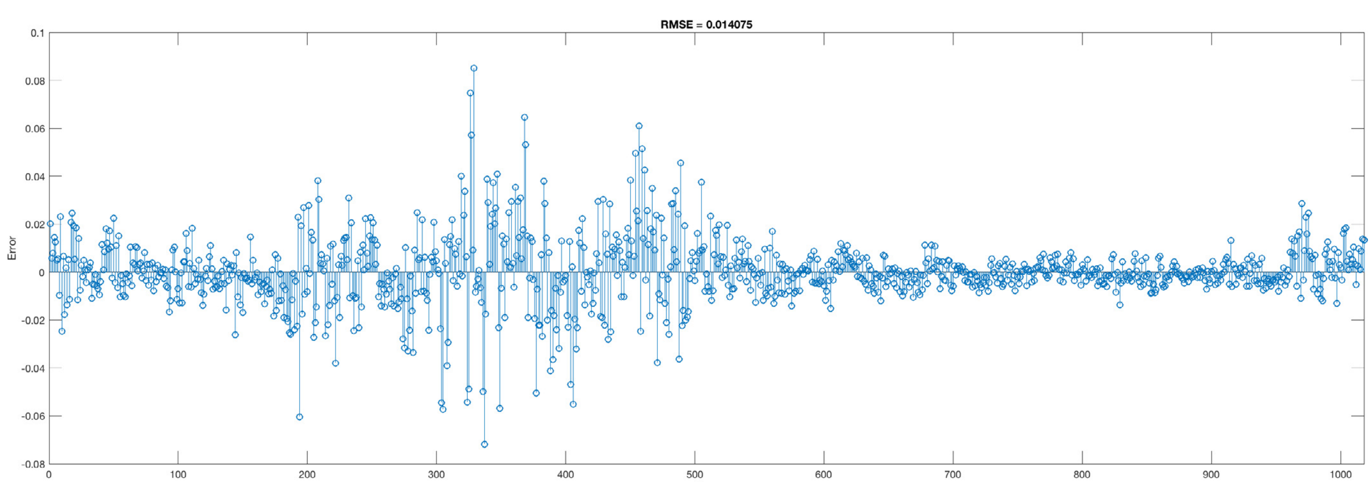

Fig. 8. RMSE in return prediction of Shanghai Composite Index.

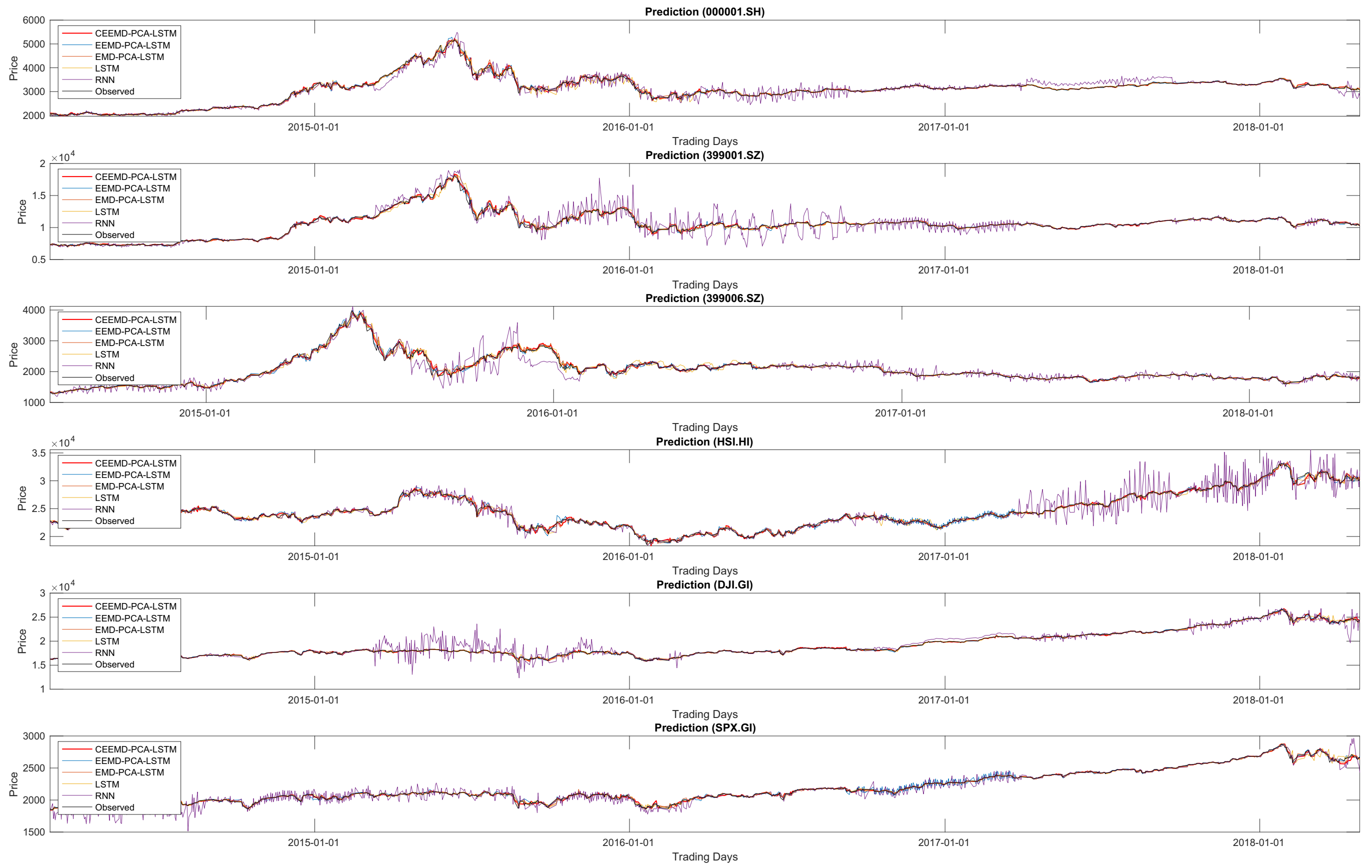

Fig. 9. Observed and predicted prices for each stock index on the back-testing from five models.

Table 2
Root Mean Square Error (RMSE) comparison of return forecasting.

Model	000001.SH	399001.SZ	399006.SZ	HSI.HI	DJI.GI	SPX.GI	Average
RNN	0.0487	0.0801	0.0733	0.0422	0.0476	0.0380	0.0550
LSTM	0.0214	0.0221	0.0286	0.0123	0.0093	0.0094	0.0172
EMD-PCA-LSTM	0.0144	0.0169	0.0230	0.0108	0.0073	0.0072	0.0133
EEMD-PCA-LSTM	0.0145	0.0157	0.0205	0.0121	0.0072	0.0106	0.0134
CEEMD-PCA-LSTM	0.0141	0.0172	0.0200	0.0105	0.0069	0.0074	0.0127

Table 3
Mean Absolute Error (MAE) comparison of return forecasting.

Model	000001.SH	399001.SZ	399006.SZ	HSI.HI	DJI.GI	SPX.GI	Average
RNN	0.0354	0.0498	0.0482	0.0258	0.0259	0.0242	0.0349
LSTM	0.0129	0.0141	0.0191	0.0090	0.0061	0.0062	0.0112
EMD-PCA-LSTM	0.0092	0.0106	0.0145	0.0080	0.0052	0.0051	0.0088
EEMD-PCA-LSTM	0.0088	0.0106	0.0138	0.0090	0.0052	0.0073	0.0091
CEEMD-PCA-LSTM	0.0092	0.0110	0.0130	0.0077	0.0050	0.0050	0.0085

Table 4
Normalized Mean Square Error (NMSE) comparison of return forecasting.

Model	000001.SH	399001.SZ	399006.SZ	HSI.HI	DJI.GI	SPX.GI	Average
RNN	9.8622	20.2164	11.1687	15.1299	34.5512	21.8278	18.7927
LSTM	1.9015	1.5367	1.6931	1.2934	1.3396	1.3318	1.5160
EMD-PCA-LSTM	0.8610	0.8975	1.1011	0.9921	0.8040	0.7951	0.9085
EEMD-PCA-LSTM	0.8799	0.7774	0.8734	1.2443	0.7998	1.7002	1.0458
CEEMD-PCA-LSTM	0.8251	0.9371	0.8334	0.9376	0.7230	0.8266	0.8471

the contrary, volatility of a specific industry & sector is affected by a confluence of factors, from weaker short-range correlation to investors sentiments caused by psychological forces, with an outcome of relatively larger fluctuation and forecasting error. For example, in predicting GEM Index and Shanghai Composite Index using CEEMD-PCA-LSTM model, RMSE of the former is significantly increased by 41.84% compared with the latter despite the two indices are both from the developing financial market in mainland China. To ensure the robustness of empirical results, we evaluate statistical significance of the differences between deep learning hybrid models and the rest two models, utilizing statistical approach, T-test. Finally, the statistical evidence proves that the differences between deep learning hybrid models and the rest two single models are all statistically significant at 99% confidence level for stock indices.

Similarly, MAE and NMSE predicted by the five models are compared, as reported in Tables 3 and 4, respectively. Specifically, MAE and NMSE of CEEMD-PCA-LSTM reach 0.0085 and 0.8471, respectively, which are smaller than the other four models and indicate the better predictive performance. As illustrated in Table 5, the DS (*i.e.*, predictive directional accuracy) of CEEMD-PCA-LSTM is higher than the other four models, with the average 0.7333. Meanwhile, it can be found that the prediction errors based on the deep learning hybrid models—CEEMD-PCA-LSTM, EEMD-PCA-LSTM, EMD-PCA-LSTM—are significantly smaller than those based on single models (RNN, LSTM). Specifically, hybrid models' evaluation indicators RMSE, MAE, and NMSE decrease by 63.61%, 61.88%, and 90.80%, respectively, of the corresponding forecasting errors of single models, while hybrid models' predictive directional accuracy, DS, improves over single models by 41.21%. Still, these differences between deep learning hybrid models and the other single models pass the T-test at 99% significant confidence level. Consequently, it can be concluded that hybrid models based on deep learning have great advantages and spectacular potential on financial time series forecasting.

In the perspective of different developed levels of financial markets, the following three conclusions can be drawn.

Firstly, the forecasting error indicators RMSE, MAE, and NMSE are generally declining as the market maturity increases, which can be denoted as:

$$RMSE_{Developing} > RMSE_{Relatively\ Developed} > RMSE_{Developed} \tag{20}$$

$$MAE_{Developing} > MAE_{Relatively\ Developed} > MAE_{Developed} \tag{21}$$

$$NMSE_{Developing} > NMSE_{Relatively\ Developed} > NMSE_{Developed} \tag{22}$$

In other words, the more mature and developed the market, the better the stability and the smaller the volatility. Moreover, the smaller the test error between predicted and observed value, the better the predictive performance. This can be attributed to that the emerging markets are susceptible to investor sentiments, along with inaccurate positioning of stock market function and imperfect innovation system.

Secondly, if two stock indices belong to the same developed level markets (or same degree of development), the difference between forecasting errors of a specific model in forecasting the two indices is extremely small. On the contrary, when the two stock indices belong to different developed level markets, the difference between forecasting errors of a specific model in forecasting the two indices increases significantly. For example, MAEs of Shanghai Composite Index, SZSE Component Index and GEM Index predicted by CEEMD-PCA-LSTM are 0.0092, 0.0110 and 0.0130, respectively. Whereas, MAEs of Dow Jones Index and S&P 500 Index predicted by the same method are 0.0050 and 0.0050, respectively. Similarly, this conclusion can be analogized to other four models.

Thirdly, the fluctuation degree of forecasting error is also different when the same model is applied in different markets. For example, when LSTM model is utilized to predict the six indices, MAE value varies from 0.0061 to 0.0191 with a large volatility, which means that the MAE in worst-case is three times the MAE in the best forecasting case. In contrast, CEEMD-PCA-LSTM performs with great consistency in various markets, which may be attributed to CEEMD's better performance in data denoising than other models. Consequently, in an extent, CEEMD-PCA-LSTM can be presumably superior, more practical and more feasible than other models when predicting less mature and more volatile financial markets.

6. Trading simulation – Profitability performance evaluation

To verify the validity and superiority of deep learning hybrid prediction model—CEEMD-PCA-LSTM proposed in this paper, we conduct trading simulations to evaluate the profitability performance, where absolute return indicators and risk-adjusted return indicators are adopted to measure the profitability performance. In addition, we compare the trading simulation results using CEEMD-PCA-LSTM model with those of benchmark strategies. Furthermore, we conduct a sensitivity analysis of transaction cost.

6.1. Strategy

The benchmark strategy is Buy and Hold (B&H), also called position trading, which is an investment strategy where an investor buys stocks and holds them for a long time, with the goal that stocks will gradually increase in value over a long period of time. In this paper, we improve the simulation trading strategy reported by Baek and Kim (2018) via adding a real-world constraint factor—transaction cost that plays an essential role in trading, and we abbreviate it as STS. Table 6 lists the specifics of this trading strategy. First, expected rate of return, r_t, of the underlying asset on the trading day t is calculated as:

$$r_t = \frac{Close_t - Open_t}{Open_t} \tag{23}$$

where $Open_t$ is the opening price of trading day t, $Close_t$ and $Close_t$ denote the predicted closing price and observed closing price, respectively, on the trading day t. Then, threshold level $\kappa(\kappa > 0)$ is reasonably set, which is based on transaction decisions as pre-

Table 5
Directional Symmetry (DS) comparison of return forecasting. ** and *** indicate that it is significantly greater than 0.7000 at 95% and 99% confidence levels, respectively, based on the T-test.

Model	000001.SH	399001.SZ	399006.SZ	HSI.HI	DJI.GI	SPX.GI	Average
RNN	0.4769	0.4867	0.5348	0.4907	0.5074	0.5064	0.5005
LSTM	0.4631	0.5015	0.5098	0.5438	0.5497	0.5733	0.5235
EMD-PCA-LSTM	0.7384	0.7178	0.7272	0.6952	0.7375	0.7217	0.7230**
EEMD-PCA-LSTM	0.7227	0.7217	0.7076	0.6991	0.7119	0.7139	0.7128**
CEEMD-PCA-LSTM	0.7345	0.7443	0.7467	0.7099	0.7365	0.7276	0.7333***

Table 6
A trading strategy using the predicted output by deep learning model.

Conditions	Transactions	Profit
$r_t > \kappa$	Buy→Sell	$Close_t(1-cost)-Open_t(1+cost)$
$r_t < -\kappa$	Sell→Buy	$Open_t(1-cost)-Close_t(1+cost)$
$-\kappa < r_t < \kappa$	Hold	0

sented in the second column of Table 6. For the convenient, the transaction cost is specified as a fixed proportion of transaction amount. Specifically, if expected rate of return r_t is greater than threshold level κ, transactions of purchasing a unit of underlying asset at the opening price and selling the unit at the closing price are done. If expected rate of return r_t is less than threshold level $-\kappa$, then the reversal transactions are made, that is, selling a unit of underlying asset at the opening price and purchasing the unit at the closing price. Otherwise, no transactions occur, with holding the underlying asset. The daily trading profit is calculated as indicated in the last column of Table 6. For the sake of convenience, we abbreviate the STS using the predicted output by CEEMD-PCA-LSTM as CEEMD-PCA-LSTM strategy, whose abbreviation can be analogized to other models. In addition, CEEMD-PCA-LSTM, EEMD-PCA-LSTM, EMD-PCA-LSTM strategies are collectively referred to as deep learning hybrid strategies.

6.2. Trading simulation results and analysis

We select Shanghai Composite Index (000001.SH), Hang Seng Index (HSI.HI), Dow Jones Index (DJI.GI) and S&P 500 Index (SPX.GI) as the underlying assets, whose sample data is acquired from Wind Database. The back-testing trading simulation is performed over the period from February 28, 2014 to April 27, 2018. Meanwhile, the threshold level κ of four target assets and transaction cost are specified as 0.5% and 0.015%, respectively. Furthermore, absolute return indicators and risk-adjusted return indicators are utilized to evaluate the profitability performance. Specifically, risk-adjusted return defines an investment's return by measuring how much risk is involved in producing that return, among which we choose the common risk measures, including Information Ratio, Sharpe Ratio, Treynor Ratio and Jensen'α. Overall, for the underlying assets, the trading simulation results based on deep learning hybrid strategies outperform those with the B&H, LSTM and RNN strategies.

Table 7 summarizes the absolute return indicators and risk-adjusted return indicators of Shanghai Composite Index using six investment strategies. The results indicate that the deep learning hybrid strategies significantly outperform single investment strategies, *i.e.*, LSTM and RNN and B&H strategies. Specifically, the average annualized returns of CEEMD-PCA-LSTM EEMD-PCA-LSTM and EMD-PCA-LSTM strategy are 51.22%, 50.83% and 50.14%, respectively, which are all approximately 5 times higher than B&H strategy. The total investment returns of LSTM and RNN are both negative, which are consistent with the large prediction errors and low accuracies of the two models as reported in Tables 2 to 5. Additionally, this is partly due to the related trading strategy settings (threshold level, transaction cost and allowing short), and partly due to partial test interval being in the stock market crash (Shanghai Composite Index fell from 5,507 points on June 8 to 3,507 points on July 8 in 2015, which lasted only 18 trading days, plummeting 32.11%). In the perspective of Max drawdown indicator, CEEMD-PCA-LSTM strategy achieves the smallest (5.35%), which is slightly less than 6.74% of EEMD-PCA-LSTM strategy, whereas far less than 48.60% of B&H strategy. Therefore, when the investment asset goes down (especially the plunge), deep learning hybrid strategies can effectively avoid the risk of drawdown and acquire substantial earnings. Furthermore, deep learning hybrid strategies can achieve exceptional performance on risk-adjusted return indicators as well. In terms of Sharpe ratio indicator, for example, CEEMD-PCA-LSTM strategy reaches the highest at 0.1895, followed by EEMD-PCA-LSTM and EMD-PCA-LSTM strategy, which is seven times as high as the benchmark strategy B&H. Intriguingly, in terms of the rest of indicators, *i.e.*, Treynor Ratio, Information Ratio, and Jensen'α, it can be found that deep learning hybrid strategies obviously outperform benchmark strategies, consistent with the findings above.

Table 8 and Tables 9-10 report profitability performance indicators (*i.e.*, absolute return indicators and risk-adjusted return indicators) of the six trading strategies in Hong Kong stock market and US stock market, respectively. Similarly, it can be found that deep learning hybrid strategies significantly outperform both single strategy and B&H strategy, where CEEMD-PCA-LSTM strategy is superior to the other two deep learning hybrid strategies.

Table 7
Investment performance of deep learning trading strategies in China's Shanghai Stock Market (000001.SH).

Trading Strategy	CEEMD-PCA-LSTM	EEMD-PCA-LSTM	EMD-PCA-LSTM	LSTM	RNN	B&H
Total investment return	438.68%	433.12%	423.20%	−164.06%	−453.43%	50.55%
Average annualized return	51.22%	50.83%	50.14%	−40.29%	−111.35%	10.57%
Max drawdown	5.35%	6.74%	12.11%	135.92	138.40	48.60%
Information ratio	0.0588	0.0555	0.0569	−0.0123	0.0038	−0.0478
Sharpe ratio	0.1895	0.1700	0.1767	−0.0118	−0.0394	0.0251
Treynor ratio	0.0630	−0.1078	0.0499	−0.0117	0.0138	0.0004
Jensen'α	0.1500	0.0016	0.0015	−0.0187	0.0038	−0.0001

Table 8
Investment performance of deep learning trading strategies in Hong Kong Stock Market (HSI.HI).

Trading Strategy	CEEMD-PCA-LSTM	EEMD-PCA-LSTM	EMD-PCA-LSTM	LSTM	RNN	B&H
Total investment return	47.53%	20.44%	10.70%	−132.97%	−270.01%	32.65%
Average annualized return	10.02%	4.67%	2.53%	−32.65%	−66.31%	7.18%
Max drawdown	7.82%	29.90%	21.05%	94.31	231.80	34.81%
Information ratio	0.0047	−0.0115	−0.1670	0.0150	0.0386	0.0092
Sharpe ratio	0.0784	0.0360	0.0187	0.0171	0.0408	0.0307
Treynor ratio	−0.0112	0.0925	−0.0017	−0.0115	−0.0088	0.0004
Jensen'α	0.0004	0.0001	0.0001	0.0028	0.0071	0.0001

Figs. 10–13 plot the cumulative returns of deep learning hybrid strategy over the trading simulation of Shanghai Composite Index, Hang Seng Index, Dow Jones Index, and S&P 500 Index, respectively. We observe that CEEMD-PCA-LSTM strategy yields the highest cumulative return and outperforms other trading strategies regardless of market developed degree. In the trading simulation of Shanghai Composite Index in developing markets, the cumulative return of deep learning hybrid trading strategy is considerably higher than that of B&H strategy, where CEEMD-PCA-LSTM reaches as high as 438.68%. Specifically, As illustrated in Fig. 10, there was

Table 9
Investment performance of deep learning trading strategies in United States Stock Market (SPX.GI).

Trading Strategy	CEEMD-PCA-LSTM	EEMD-PCA-LSTM	EMD-PCA-LSTM	LSTM	RNN	B&H
Total investment return	83.88%	62.25%	88.93%	−59.69%	−293.19%	43.99%
Average annualized return	16.14%	12.62%	16.91%	−14.66%	−72.00%	9.37%
Max drawdown	4.74%	22.71%	4.33%	61.11%	848.14	12.74%
Information ratio	0.0190	0.0064	0.0225	−0.1213	−0.0148	−0.0441
Sharpe ratio	0.1429	0.0929	0.1436	−0.1066	−0.0121	0.0458
Treynor ratio	−0.0106	0.0535	−0.0783	−0.0055	0.0344	0.0004
Jensen'α	0.0006	0.0004	0.0006	−0.0009	−0.0018	−0.00004

Table 10
Investment performance of deep learning trading strategies in United States Stock Market (DJI.GI).

Trading Strategy	CEEMD-PCA-LSTM	EEMD-PCA-LSTM	EMD-PCA-LSTM	LSTM	RNN	B&H
Total investment return	116.83%	91.40%	92.25%	−47.24%	−281.07%	49.40%
Average annualized return	20.93%	17.28%	17.41%	−11.60%	−69.03%	10.36%
Max drawdown	7.93%	3.89%	7.92%	50.99%	402.99	14.45%
Information ratio	0.0360	0.0233	0.0232	−0.0974	−0.0272	−0.0239
Sharpe ratio	0.1929	0.1475	0.1495	−0.0788	−0.0263	0.0504
Treynor ratio	−0.0105	−0.0149	−0.0082	−0.4724	−0.0370	0.0005
Jensen'α	0.0007	0.0006	0.0006	−0.0006	−0.0120	0.00002

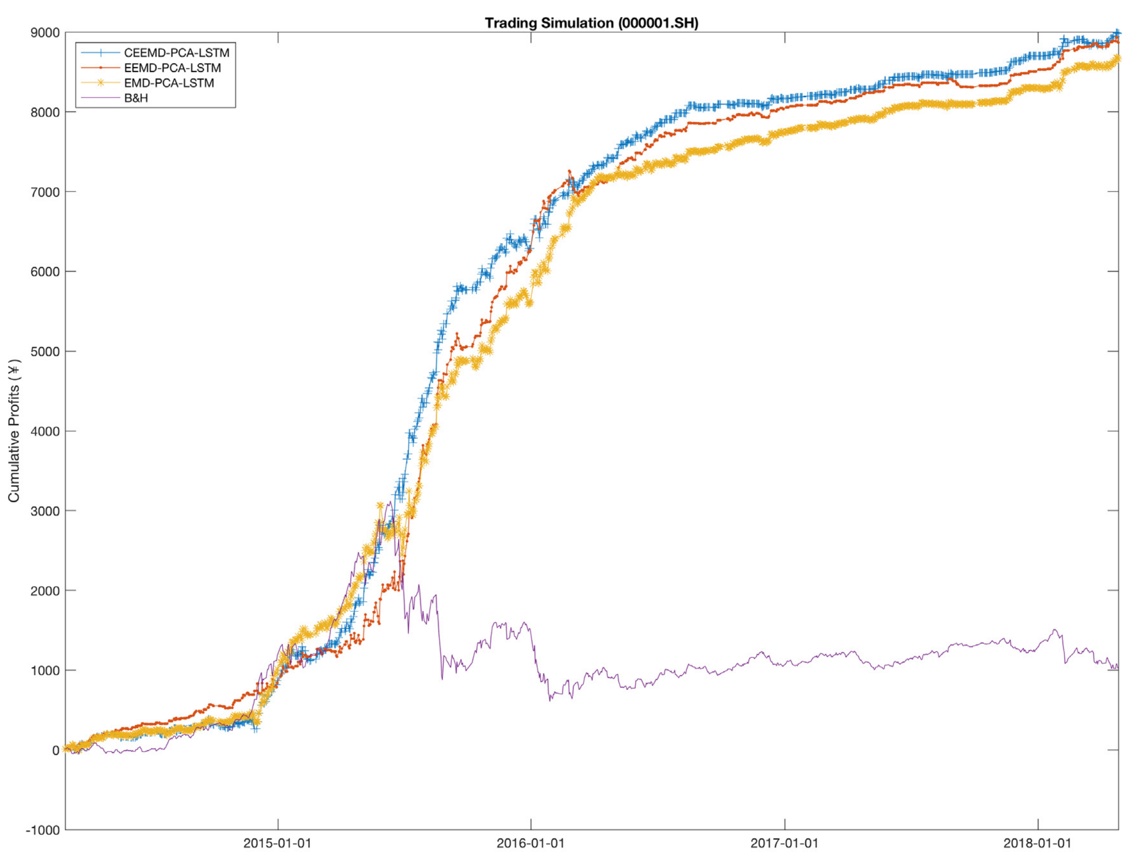

Fig. 10. Cumulative return of Shanghai Composite Index trading simulation.

almost no difference in the cumulative return of the four strategies, when the stock market rose before the collapse of 2015. However, after the stock market crashed, the returns of the three deep learning hybrid strategies soared sharply, on the premise that short-selling was allowed in the trading simulation, while limited in the actual trading. When the stock market gradually rebounded after bottoming out, return growth of deep learning hybrid strategies slowed down drastically, but still marginally higher than that of B&H strategy.

In the perspective of Hang Seng Index in a relatively developed market, Dow Jones Index and S&P 500 Index in a developed market, the gap of returns between the deep learning hybrid strategies and the benchmark strategy is apparently narrowed. More specifically, the total return of CEEMD-PCA-LSTM strategy reaches 47.53% in the trading simulation of Hang Seng Index, which is higher than 32.63% of benchmark strategy—B&H strategy, while the total returns of the other two deep learning hybrid strategies all unexpectedly underperform the benchmark strategy. However, all the deep learning hybrid strategies outperform benchmark strategy in terms of total return of Dow Jones Index during the period of trading simulation. Particularly, in the upward trend of the three stock indices, deep learning hybrid trading strategies achieved little advantages over the benchmark strategy, where the cumulative returns partially remain steady and partially drop gradually for the Hang Seng Index, almost stay stable for the Dow Jones Index, and mostly level off for S&P 500 Index except for using EEMD, respectively. On the other hand, in the downward trend of the stock market, the deep learning hybrid trading strategies can control the risk of drawdown appropriately, achieving the exceptional investment performance superior to the benchmark strategy. This can be concluded that deep learning hybrid strategies outperform the benchmark, especially when the stock market falls or plunges, indicating that the strategies can effectively avoid risk and can specialize in shorting to gain profits with the support of outstanding predictive performance.

6.3. Sensitivity analysis of transaction cost

In the quantitative investment strategy, transaction cost obviously plays a pivotal role among indicators affecting earnings. Since it is vital to consider such conditions when making trading decisions, trading systems should be elaborated under real-life financial environments. If excess returns cannot be obtained with a certain strategy in trading simulation, the strategy tends to be even more inappropriate in actual transactions with uncertainty and idiosyncratic risk. Transaction costs include stamp duty (only sellers charge), securities management fees, securities transaction handling fees, transfer fees, trading commissions, etc. For the convenience of research, we adopt bilateral proportional transaction costs, which are fixed at fifteen in ten thousand of transaction amount. This paper only takes into account explicit transaction costs, but not implicit ones, such as information costs, impact costs, etc.

Fig. 14 demonstrates the average annualized returns of CEEMD-PCA-LSTM, EEMD-PCA-LSTM, and EMD-PCA-LSTM strategy at different level of transaction costs. It can be observed that the average annualized return of deep learning hybrid strategies falls gradually with the mounting of transaction cost, among which CEEMD-PCA-LSTM has the slowest decline rate, while EMD-PCA-LSTM declines fastest. This is in large part to the capacity that CEEMD yields the completeness and orthogonality of decomposition result and addresses the issue of modal aliasing, to which EMD is incompe-

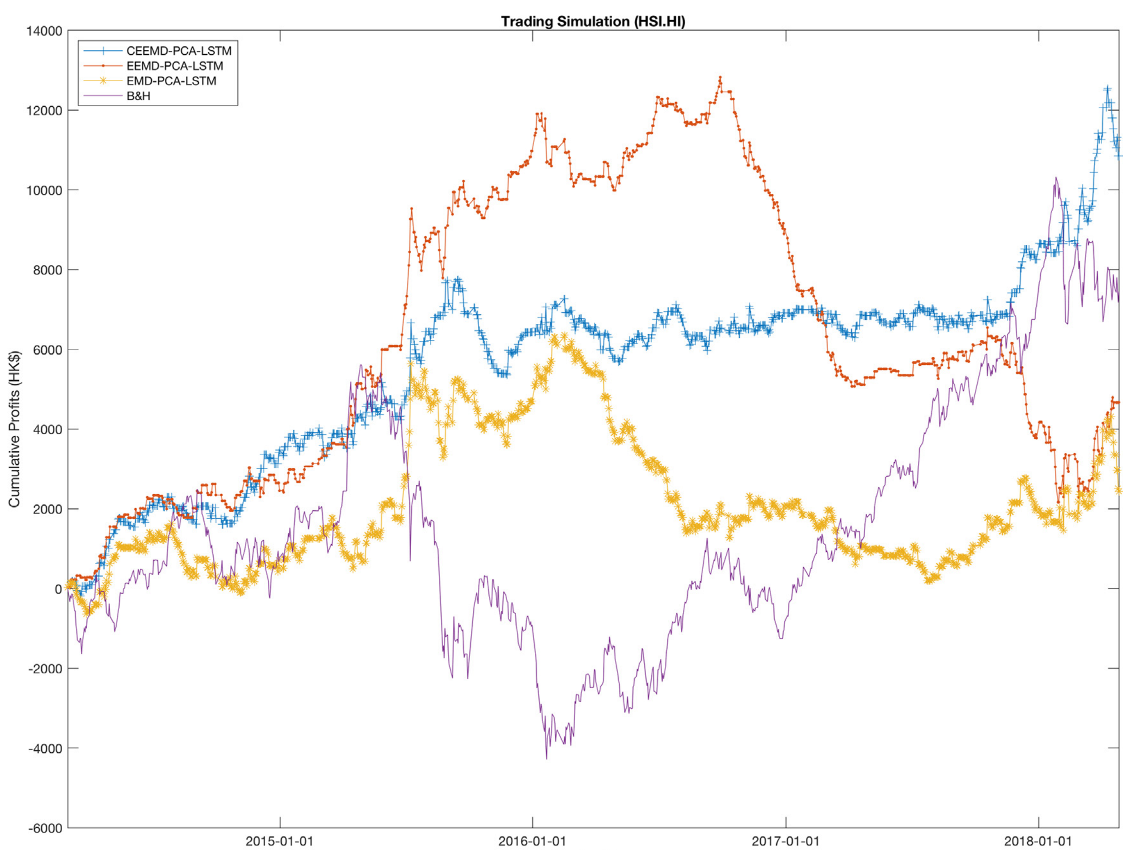

Fig. 11. Cumulative return of Hang Seng Index trading simulation.

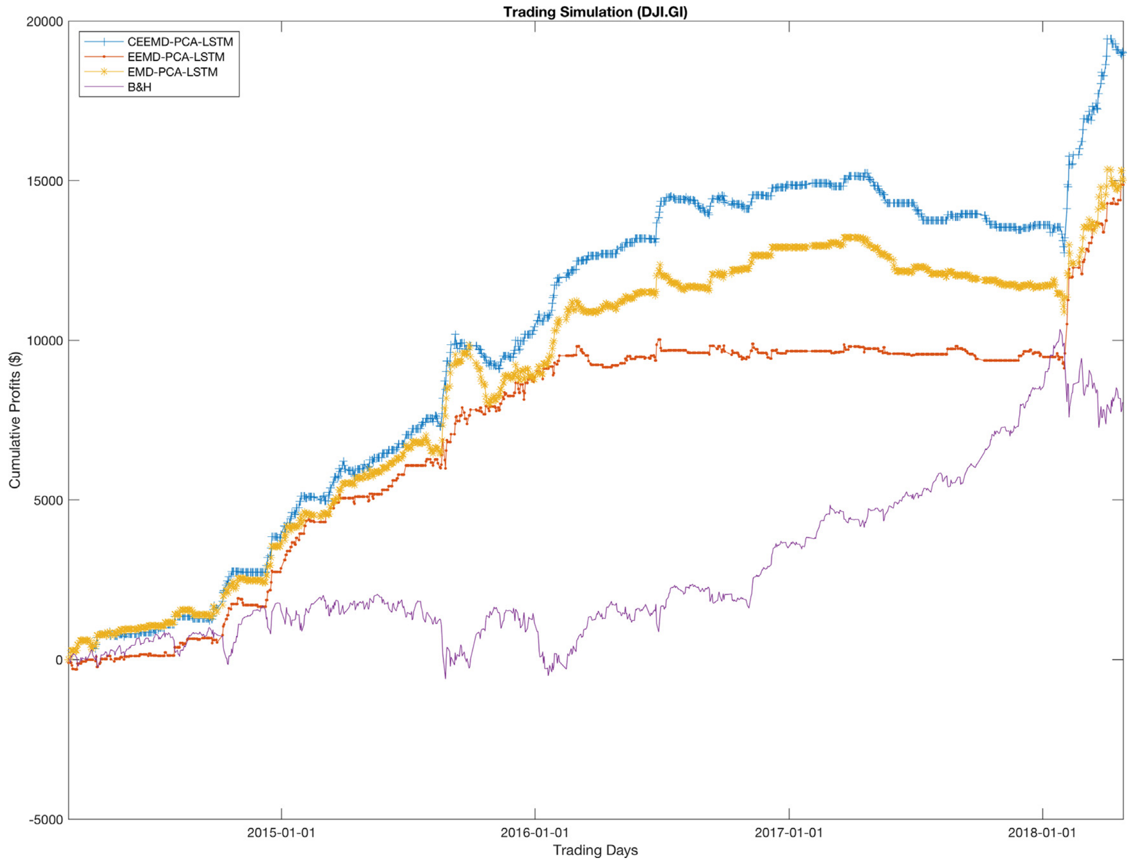

Fig. 12. Cumulative return of Dow Jones Index trading simulation.

tent. Furthermore, in the actual transaction, with the day-to-day competition of securities firms and the rapid development of information technology, the transaction cost will not be higher than fifteen in ten thousand deployed in this paper.

7. Model robustness test

The experimental results confirm that deep learning model prediction has a strong practical investment value. Therefore, this section analyzes the robustness of CEEMD-PCA-LSTM itself and the sensitivity of parameters. We evaluate the robustness of deep learning model by varying the number of principal components and the number of LSTM hidden layers. For the sake of simplicity, we only take the Shanghai Composite Index as the target dataset in this section.

Number of principal components. Theoretically, PCA can ensure the validity and completeness of data, and simultaneously can achieve dimensionality reduction, reduce training time of deep learning, and greatly improve operational efficiency of overall model. As mounting the number of principal components, the cumulative contribution rate increases, but obviously more time is required for training deep learning. Fig. 15 depicts a graph illustrating the predictive error, RMSE, of deep learning hybrid models CEEMD-PCA-LSTM, EEMD-PCA-LSTM and EMD-PCA-LSTM when varying the number of principal components of PCA. As observed from Fig. 15, when the number of principal components ascends, the prediction error RMSE exhibits a process of first decreasing and then bottoming out at around 0.0055, where the prediction error RMSE of EMD-PCA-LSTM fluctuates significantly. The three models in ascending order of overall prediction error RMSE are CEEMD-PCA-LSTM, EEMD-PCA-LSTM, EMD-PCA-LSTM. Therefore, CEEMD-PCA-LSTM outperform the latter two in terms of model robustness when varying the number of principal components.

Number of LSTM hidden layers. Theoretically, with the increase of the number of LSTM hidden layers, the more abstract features deep learning extracts, the more approximate it can be to financial time series, which is more favorable for prediction. Fig. 16 depicts a graph illustrating the predictive error, RMSE, of deep learning models CEEMD-PCA-LSTM, EEMD-PCA-LSTM, and EMD-PCA-LSTM when changing the number of LSTM hidden layers. It can be observed from Fig. 16 that RMSE gradually decreases except for outliers as the number of LSTM hidden layers climbs from 1 to 5. In addition, at the stage when the number of hidden layers of LSTM varies from 6 to 40, the RMSE fluctuates relatively greatly, with some individual points oscillating upwards. However, when the number of LSTM hidden layers continues to mount (*i.e.*, greater than 40), RMSE tends to level off at 0.006 ~ 0.007 and predictive error stays relatively stable, indicating that all three models demonstrate steady robustness. Furthermore, at the stage when prediction error tends to be stationary, the three models with prediction errors from low to high are: CEEMD-PCA-LSTM, EEMD-PCA-LSTM, EMD-PCA-LSTM, indicating that CEEMD-PCA-LSTM achieves better predictive performance and more stable robustness than the latter two.

8. Conclusion

In this paper, a novel methodology for financial time series forecasting based on deep learning is proposed. On that basis, a deep learning hybrid model — CEEMD-PCA-LSTM —to predict the one-step-ahead closing price of stock indices is constructed. The proposed model consists of four modules: Firstly, CEEMD is applied

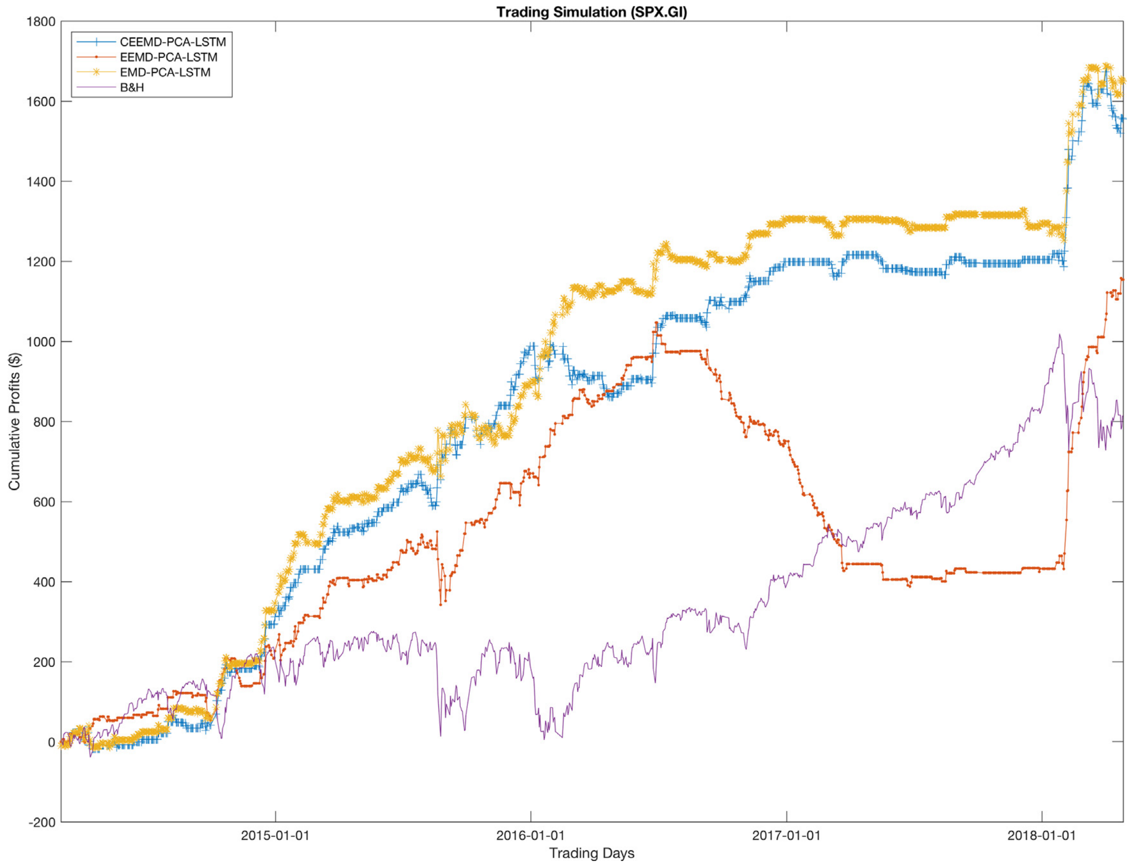

Fig. 13. Cumulative return of S&P 500 Index trading simulation.

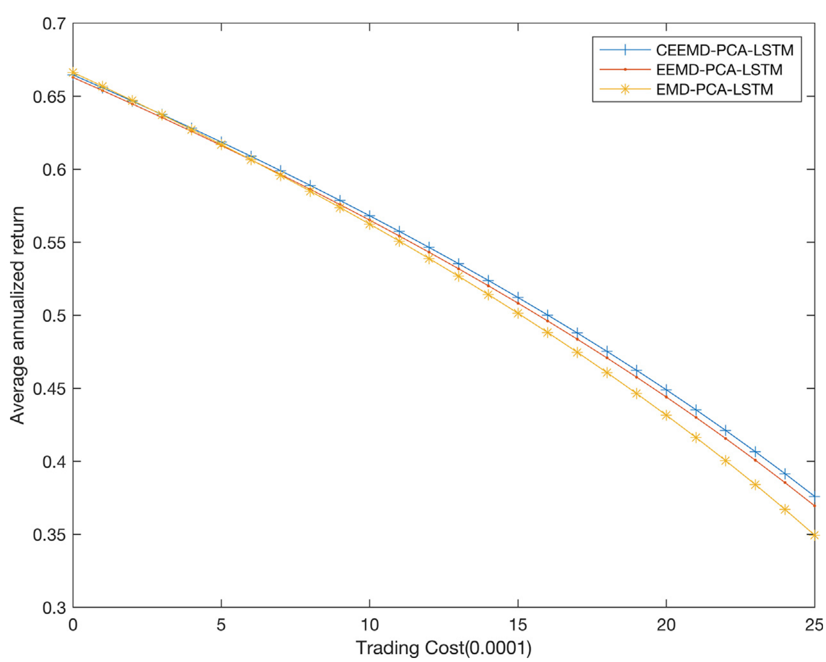

Fig. 14. Sensitivity analysis of deep learning trading strategies to transaction costs.

to decompose the fluctuations or trends of different scales, namely IMF, from the original financial time series. Then, PCA is used to perform data dimensionality reduction and extract the abstract and high-level features, laying a solid foundation for subsequent modules. After that, the features are separately fed into the LSTM networks to predict the closing price of the next trading day for each component. Finally, the predicted values of individual components are synthesized to obtain a final predicted value.

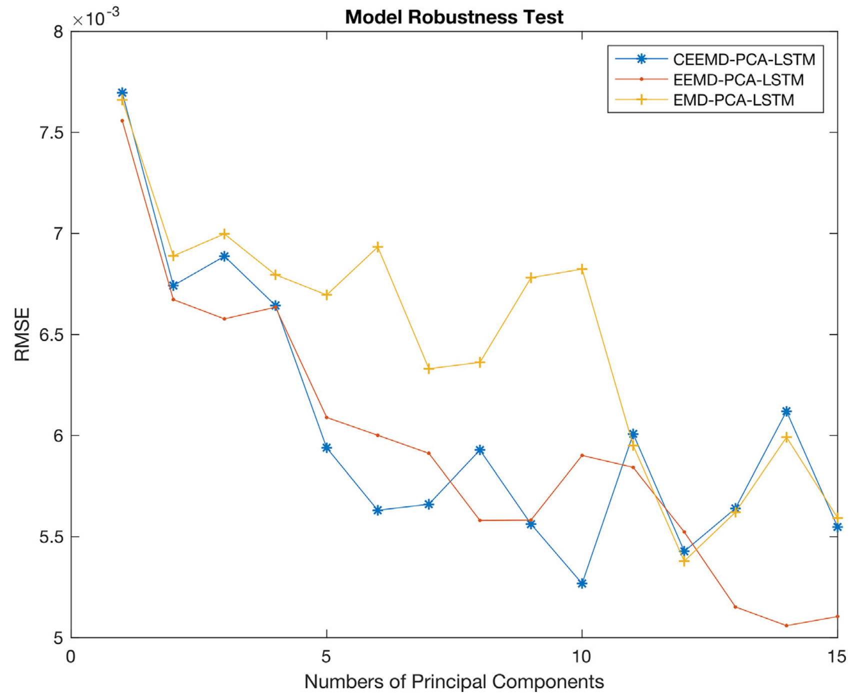

Fig. 15. Effect of the number of principal components on RMSE predicted by deep learning hybrid models.

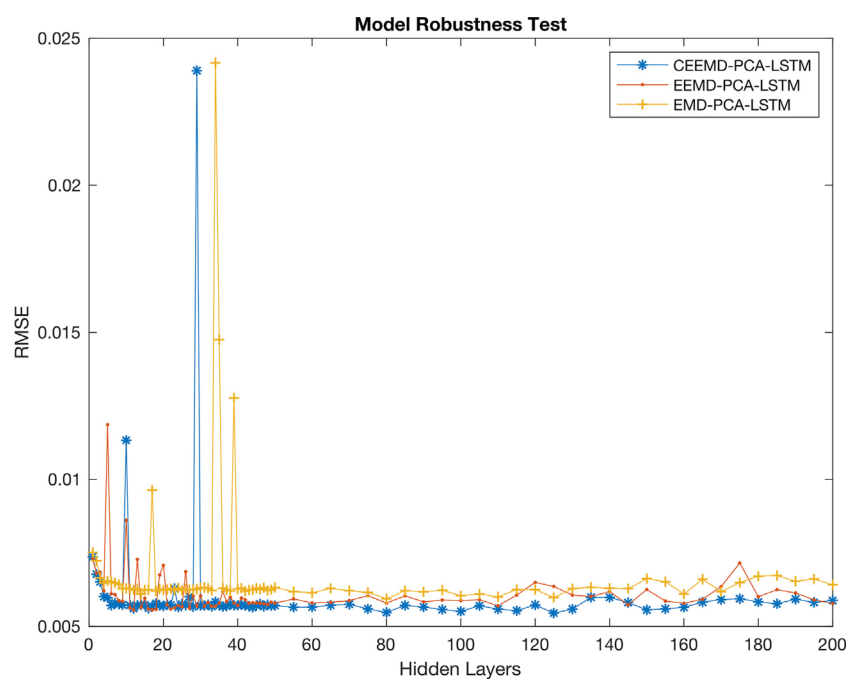

Fig. 16. Effect of the number of hidden layers on RMSE of returns predicted by deep learning hybrid models.

We apply the proposed model, along with other benchmark models, into six stock indices from three different developed stock markets to test the predictive accuracy, including test error (RMSE, MAE, NMSE) and directional symmetry. Furthermore, we evaluate the absolute profitability performance and risk-adjusted profitability performance of deep learning hybrid strategies based on proposed model compared to benchmark strategies in the trading simulation. We observe that our proposed model, CEEMD-PCA-LSTM, outperforms benchmark models in both predictive accuracy and profitability performance regardless of which stock index is chosen for experiment. Intriguingly, some practical conclusions are derived from our experiment. First, the test error indicators RMSE, MAE, and NMSE generally drops as the stock markets maturity degree increases. Second, as indicated from the absolute return indicators and risk-adjusted return indicators, deep learning hybrid strategies can yield stable and high return and can effectively avoid the risk of drawdown, especially when the stock market declines or plunges.

With our work, we make three key contributions to the existing literature:

Firstly, a novel methodology of financial time series prediction based on deep learning is proposed, an idea of "Decomposition-Reconstruction-Synthesis" that makes it feasible and efficient to model and forecast the nonlinear, non-stationary and multi-scale complex financial time series. Within this methodology and framework, the forecasting model can be improved by replacing each module with a state-of-the-art method in the areas of denoising, deep features extracting or time series fitting. Secondly, CEEMD, derived from and superior to EMD and EEMD, is applied to the prediction of financial time series, decomposing the IMFs of different scale from which high-level abstract features are subsequently extracted for deep learning. Thirdly, conceptual and empirical aspects on CEEMD-PCA-LSTM outlined in this paper go beyond a pure financial market application, but are intended as guideline for other researchers, wishing to deploy this effective methodology to other time series forecasting tasks.

This paper sheds light on deep learning framework for forecasting financial time series, an attempt to continue to push the boundaries of deep learning and financial application domain, and does suggest promising extensions and directions for future investigation. However, it still has some limitations. For example, additional factors that are known to carry information regarding the future price movement, such as technical indicators trading volume and the price of a derivative linked to the asset, can be augmented (Akita, Yoshihara, Matsubara, & Uehara, 2016) to the input. In addition, numerous studies unveiled that stock market movement can be modeled with various information from different categories, such as macroeconomic factors, investors' sentiments and news events (Akita et al., 2016; Li, Bu, & Wu, 2017). Furthermore, a more advanced hyper-parameters selection scheme might be embedded in the system to further optimize the proposed deep learning methodology. All of these could be enhanced by future studies.

Financial support

This work was supported by National Natural Science Foundation of China (NSFC) funded by the Ministry of Education, Science and Technology (NSFC-70972115). The funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

CRediT authorship contribution statement

Yong'an Zhang: Conceptualization, Investigation, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Binbin Yan:** Writing - original draft, Methodology, Software, Data curation, Visualization. **Memon Aasma:** Writing - review & editing, Investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). Deep learning for stock prediction using numerical and textual information. In Proceedings of the Ieee/Acis 15th International Conference on Computer and Information Science (ICIS) (pp. 945-950). Okayama, JAPAN.

Arel, I., Rose, D. C., & Karnowski, T. P. (2010). Deep machine learning-a new frontier in artificial intelligence research. *IEEE Computational Intelligence Magazine, 5*(4), 13–18. https://doi.org/10.1109/mci.2010.938364.

Awajan, A. M., & Ismail, M. T. (2017). A hybrid approach EMD-HW for short-term forecasting of daily stock market time series data. In Proceedings of the 24th National Symposium on Mathematical Sciences (SKSM)

Awajan, A. M., Ismail, M. T., & Al Wadi, S. (2018). Improving forecasting accuracy for stock market data using EMD-HW bagging. *PLoS One, 13*(7). https://doi.org/10.1371/journal.pone.0199582.

Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications, 113*, 457–480. https://doi.org/10.1016/j.eswa.2018.07.019.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS One, 12*(7). https://doi.org/10.1371/journal.pone.0180944 e0180944.

Bisoi, R., Dash, P. K., & Parida, A. K. (2019). Hybrid variational mode decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis. *Applied Soft Computing, 74*, 652–678. https://doi.org/10.1016/j.asoc.2018.11.008.

Cavalcante, R. C., Brasileiro, R. C., Souza, V. L. P., Nobrega, J. P., & Oliveira, A. L. I. (2016). Computational Intelligence and Financial Markets: A Survey and Future Directions. *Expert Systems with Applications, 55*, 194–211. https://doi.org/10.1016/j.eswa.2016.02.006.

Chatzis, S. P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., & Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications, 112*, 353–371. https://doi.org/10.1016/j.eswa.2018.06.032.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications, 83*, 187–205. https://doi.org/10.1016/j.eswa.2017.04.030.

Chung, H., & Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability, 10*(10), 18. https://doi.org/10.3390/su10103765.

Cortez, B., Carrera, B., Kim, Y. J., & Jung, J. Y. (2018). An architecture for emergency event prediction using LSTM recurrent neural networks. *Expert Systems with Applications, 97*, 315–324. https://doi.org/10.1016/j.eswa.2017.12.037.

Day, M. Y., & Lee, C. C. (2016). Deep learning for financial sentiment analysis on finance news providers. In Proceedings of the 8th IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 1127-1134). New York, USA.

Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends®. Signal Processing, 7*(3–4), 197–387.

Dixon, M., Klabjan, D., & Bang, J. H. (2017). Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance, 6*(3–4), 67–77.

Fehrera, R., & Feuerriegela, S. (2015). Improving decision analytics with deep learning: The case of financial disclosures. stat, 1050, 9.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation, 12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015.

Gocken, M., Ozcalici, M., Boru, A., & Dosdogru, A. T. (2016). Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction. *Expert Systems with Applications, 44*, 320–331. https://doi.org/10.1016/j.eswa.2015.09.029.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507. https://doi.org/10.1126/science.1127647.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hsieh, T. J., Hsiao, H. F., & Yeh, W. C. (2011). Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied Soft Computing, 11*(2), 2510–2525. https://doi.org/10.1016/j.asoc.2010.09.007.

Huang, C. L., & Tsai, C. Y. (2009). A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Systems with Applications, 36*(2), 1529–1539. https://doi.org/10.1016/j.eswa.2007.11.062.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., . . . Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 454(1971), 903-995.

Huck, N. (2019). Large data sets and machine learning: Applications to statistical arbitrage. *European Journal of Operational Research, 278*(1), 330–342. https://doi.org/10.1016/j.ejor.2019.04.013.

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications, 100*, 234–245. https://doi.org/10.1016/j.eswa.2018.01.037.

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications, 103*, 25–37. https://doi.org/10.1016/j.eswa.2018.03.002.

Kim, T. Y., & Cho, S. B. (2018). Web traffic anomaly detection using C-LSTM neural networks. *Expert Systems with Applications, 106*, 66–76. https://doi.org/10.1016/j.eswa.2018.04.004.

Kristjanpoller, W., & Hernandez, P. E. (2017). Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. *Expert Systems with Applications, 84*, 290–300. https://doi.org/10.1016/j.eswa.2017.05.024.
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the Acm, 60*(6), 84–90. https://doi.org/10.1145/3065386.
Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In Proceedings of the Advances in neural information processing systems (pp. 231-238).
Langkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters, 42*, 11–24. https://doi.org/10.1016/j.patrec.2014.01.008.
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539.
Leong, L. Y., Hew, T. S., Ooi, K. B., Lee, V. H., & Hew, J. J. (2019). A hybrid SEM-neural network analysis of social media addiction. *Expert Systems with Applications, 133*, 296–316. https://doi.org/10.1016/j.eswa.2019.05.024.
Li, H. F. (2017). Price forecasting of stock index futures based on a new hybrid EMD-RBF neural network model. *Agro Food Industry Hi-Tech, 28*(1), 1744–1747.
Li, J., Bu, H., & Wu, J. (2017). Sentiment-aware stock market prediction: A deep learning method. In Proceedings of the 14th International Conference on Services Systems and Services Management (ICSSSM).
Li, Z. X., & Tam, V. (2017). Combining the real-time wavelet denoising and long-short-term-memory neural network for predicting stock indexes. In Proceedings of the IEEE Symposium Series on Computational Intelligence (IEEE SSCI) (pp. 82-89). New York, USA.
Liu, Y. (2019). Novel volatility forecasting using deep learning-Long Short Term Memory Recurrent Neural Networks. *Expert Systems with Applications, 132*, 99–109. https://doi.org/10.1016/j.eswa.2019.04.038.
M'ng, J. C. P., & Mehralizadeh, M. (2016). Forecasting east Asian indices futures via a novel hybrid of Wavelet-PCA denoising and artificial neural network models. PLoS One, 11(6), 29. doi:10.1371/journal.pone.0156338.
Makinen, Y., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2019). Forecasting jump arrivals in stock prices: New attention-based network architecture using limit order book data. *Quantitative Finance, 18*. https://doi.org/10.1080/14697688.2019.1634277.
Minh, D. L., Sadeghi-Niaraki, A., Huy, N. D., Min, K., & Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *Ieee Access, 6*, 55392–55404. https://doi.org/10.1109/access.2018.2868970.
Nair, B. B., Mohandas, V., & Sakthivel, N. (2010). A decision tree—rough set hybrid system for stock market trend prediction. *International Journal of Computer Applications, 6*(9), 1–6.
Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., ... Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *Ieee-Acm Transactions on Audio Speech and Language Processing, 24*(4), 694–707. https://doi.org/10.1109/taslp.2016.2520371.
Palangi, H., Ward, R., & Deng, L. (2016). Distributed compressive sensing: A deep learning approach. *Ieee Transactions on Signal Processing, 64*(17), 4504–4518. https://doi.org/10.1109/tsp.2016.2557301.
Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications, 42*(4), 2162–2172. https://doi.org/10.1016/j.eswa.2014.10.031.
Petersen, N. C., Rodrigues, F., & Pereira, F. C. (2019). Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Systems with Applications, 120*, 426–435. https://doi.org/10.1016/j.eswa.2018.11.028.
Qiu, M. Y., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS One, 11*(5), 11. https://doi.org/10.1371/journal.pone.0155133.
Ramirez, E., Melin, P., & Prado-Arechiga, G. (2019). Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 2-lead cardiac arrhythmia classification. *Expert Systems with Applications, 126*, 295–307. https://doi.org/10.1016/j.eswa.2019.02.035.
Salakhutdinov, R., & Hinton, G. (2009). Deep boltzmann machines. In Proceedings of the Artificial intelligence and statistics (pp. 448-455).
Sarikaya, R., Hinton, G. E., & Deoras, A. (2014). Application of deep belief networks for natural language understanding. *Ieee-Acm Transactions on Audio Speech and Language Processing, 22*(4), 778–784. https://doi.org/10.1109/taslp.2014.2303296.
Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing, 70*, 525–538. https://doi.org/10.1016/j.asoc.2018.04.024.
Shen, F., Chao, J., & Zhao, J. (2015). Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing, 167*, 243–253. https://doi.org/10.1016/j.neucom.2015.04.071.
Sim, H. S., Kim, H. I., & Ahn, J. J. (2019). Is deep learning for image recognition applicable to stock market prediction?. *Complexity, 10*. https://doi.org/10.1155/2019/4324878.
Singh, R., & Srivastava, S. (2017). Stock prediction using deep learning. *Multimedia Tools and Applications, 76*(18), 18569–18584. https://doi.org/10.1007/s11042-016-4159-7.
Sović, A., & Seršić, D. (2012). Signal decomposition methods for reducing drawbacks of the DWT. *Engineering Review, 32*(2), 70–77.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*, 1929–1958.
Wang, J., & Wang, J. (2017). Forecasting stochastic neural network based on financial empirical mode decomposition. *Neural Networks, 90*, 8–20. https://doi.org/10.1016/j.neunet.2017.03.004.
Wei, L. Y. (2016). A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting. *Applied Soft Computing, 42*, 368–376. https://doi.org/10.1016/j.asoc.2016.01.027.
Wen, M., Li, P., Zhang, L. F., & Chen, Y. (2019). Stock market trend prediction using high-order information of time series. *Ieee Access, 7*, 28299–28308. https://doi.org/10.1109/access.2019.2901842.
Wu, Z., & Huang, N. E. (2009). Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Advances in Adaptive Data Analysis, 1*(01), 1–41.
Xiong, R., Nicholas, E., & Shen, Y. (2016). Deep learning stock volatilities with google domestic trends. arXiv preprint arXiv:1512.04916.
Xiu, Y., & Chen, X. Y. (2017). Study on prediction of the shanghai composite index based on EMD and NARX neural network. In Proceedings of the 2nd International Conference on Information Technology and Intelligent Transportation Systems (ITITS) (pp. 590-596).
Yang, R. X., He, J. Y., Xu, M. Y., Ni, H. Q., Jones, P., & Samatova, N. (2018). An intelligent and hybrid weighted fuzzy time series model based on empirical mode decomposition for financial markets forecasting. In P. Perner (Ed.), Advances in Data Mining: Applications and Theoretical Aspects (Vol. 10933, pp. 104-118).
Yeh, J. R., Shieh, J. S., & Huang, N. E. (2010). Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Advances in Adaptive Data Analysis, 2*(02), 135–156.
Zhang, N. N., Lin, A. J., & Shang, P. J. (2017). Multidimensional k-nearest neighbor model based on EEMD for financial time series forecasting. *Physica a-Statistical Mechanics and Its Applications, 477*, 161–173. https://doi.org/10.1016/j.physa.2017.02.072.
Zhang, Z. H., Zohren, S., & Roberts, S. (2019). DeepLOB: Deep convolutional neural networks for limit order books. *Ieee Transactions on Signal Processing, 67*(11), 3001–3012. https://doi.org/10.1109/tsp.2019.2907260.