CrossMark

An innovative neural network approach for stock market prediction

Xiongwen Pang[1] · **Yanqiang Zhou[1]** · **Pan Wang[1]** ·
Weiwei Lin[2] · **Victor Chang[3]**

Abstract This paper aims to develop an innovative neural network approach to achieve better stock market predictions. Data were obtained from the live stock market for real-time and off-line analysis and results of visualizations and analytics to demonstrate Internet of Multimedia of Things for stock analysis. To study the influence of market characteristics on stock prices, traditional neural network algorithms may incorrectly predict the stock market, since the initial weight of the random selection problem can be easily prone to incorrect predictions. Based on the development of word vector in deep learning, we demonstrate the concept of "stock vector." The input is no longer a single index or single stock index, but multi-stock high-dimensional historical data. We propose the deep long short-term memory neural network (LSTM) with embedded layer and the long short-term memory neural network with automatic encoder to predict the stock market. In these two models, we use the embedded layer and the automatic encoder, respectively, to vectorize the data, in a bid to forecast the stock via long short-term memory neural network. The experimental results show that the deep LSTM with embedded layer is better. Specifically, the accuracy of two models is 57.2 and 56.9%, respectively, for the Shanghai A-shares composite index. Furthermore, they are 52.4 and 52.5%, respectively, for individual stocks. We demonstrate research contributions in IMMT for neural network-based financial analysis.

✉ Weiwei Lin
linww@scut.edu.cn

Victor Chang
victorchang.research@gmail.com

[1] School of Computer, South China Normal University, Guangzhou, China

[2] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

[3] International Business School Suzhou, Xi'an Jiaotong-Liverpool University, Suzhou, China

1 Introduction

Since the establishment of the socialist market economy, the financial market plays an irreplaceable role in China. In recent years, with the development of national economy and improvement of financial services, the financial market has attracted the attention of domestic and foreign scholars and investors. They regularly propose various theories which can be applied to the practice, in the attempt to predict market trends. However, as the market is influenced by the national policies, global and regional economics, as well as psychological, human and other factors, the financial market forecasts tend to fail to achieve the desired outcomes on a frequent basis [1].

Based on various researches, neural network has been used in many areas such as pattern recognition, financial securities and signal processing. Additionally, it has been widely regarded for its advantages on regression and classification in the stock market forecast. However, traditional neural network algorithms may also fail to predict the stock market precisely, since the initial weight of the random selection problem is easy to fall into the local optimal resulting in incorrect predictions.

Most of the current researches are based on foreign stock markets, such as the Standard & Poor (S&P) index and the Nasdaq index. The stock markets in developing countries such as the Shanghai stock market make less forecasts, and thus, the effect of the prediction is not significant. Based on studies of the deep learning [2–5], this paper introduces the concept of stock vector by referring to the "word vector of natural language processing" and performs the simulation experiment on the Shanghai A-shares market through the improved long short-term neural network techniques. The neural network can make an effective forecast for the financial market in China. The data can be taken directly from the Internet to provide real-time and off-line data processing and analysis. Results of visualizations and analytics are to demonstrate Internet of Multimedia of Things (IMMT) for stock analysis. The aim is to demonstrate our work can contribute to Internet of Multimedia of Things. It can read and understand the market trends and eventually provide useful analysis for investors. Therefore, our research has combined research contributions from both theoretical and practical perspectives.

2 Related work

2.1 Application of shallow machine learning in stock forecasting

The methods of shallow machine learning commonly used in the current stock forecasting model are neural networks, support vector machines or models that combine them with other algorithms. Hsu et al. [1] demonstrated that the method of machine learning could be used to predict financial market more accurately than economic methods. He also proved that the forecasting effect of the financial market was affected by the maturity of the market, the input variable, the base forecasting time and the forecasting method.

Niaki used the neural network with 27 types of economic variables to predict the S&P index. The result showed that the neural network model, compared with trading strategy, had increased the profit of transactions [6], which relied on the method to initialize the weight. Adhikar proposed random walk (RW) and artificial neural network (ANN), RW and Elman mixed models [7]. Yu [8] came up with principal component analysis (PCA) and support vector machine (SVM) hybrid models. Zbikowski [9] proposed Fisher-based SVM model. All experiments showed that the hybrid model worked better than a single model. In order to predict European and American stock market, Ballings et al. [10], Oztekin et al. [11] and Gerlein et al. [12] introduced a variety of methods, including Bayesian method, neural network and SVM, and compared the effect of each method. Zhong used 60 kinds of economic variables and three kinds of dimensionality reduction, including PCA, kernel PCA (KPCA), fast robust PCA (FRPCA), combined with neural network to predict the S&P index. The result showed that PCA was the best, achieving 57% of accuracy [13].

In these traditional neural networks, there are some issues such as falling into local optimal value and a single hidden layer. These problems eventually lead to the lower accuracy of forecasts, since forecasts are away from actual values in stock market.

2.2 Stock market prediction based on deep learning

Hinton published the "Reducing the Dimensionality of NN" [14] that set off an upsurge of deep learning. Subsequently, a variety of deep learning models have been developed, and these models have been widely used after.

Kuremoto used a three-layer restricted Boltzmann machines (RBMs) to fit a variety of time series data [15] and also proposed to optimize Deep Belief Networks-Multilayer Perceptron (DBN-MLP) via particle swarm optimization (PSO) to predict the chaotic time series [16]; Takeuchi chose the earnings of t-13 to t-2 months and previous 20 days as input, using multiple RBM models to classify US stocks with over 53% of accuracy [17]. Zhu used 14 indicators as input such as the opening, higher, lower, closing price, related with technical analysis indicators ROC and RSI, so as to learn the historical data of the S&P index through the DBN. Since then, he forecasted the stock price. The profit was higher than the ordinary trading strategy [18]. Chang proposed business analytics of stock market performance, by which investors can trace or review the market performance of their chosen stocks. He selected the Heston model and its associated API to compute predicted stock index movement and offer a good extent of accuracy [19]. Sharang took DBN to extract the features of hidden layer and then input these features into three different classifiers to predict the up and down of US Treasury note futures in five days. The accuracy of these three models is 5–10% higher than the random predictor [20]. Batres Estrada selected as the DBN-MLP model to predict the income of the S&P index. The model had a 53% of accuracy, exceeding the regularized logistic regression and some MLP baselines [21]. SH proposed to use the deep learning framework DBN for default forecast. The result showed that the effect of DBN was higher than the traditional machine learning algorithm [22]. Shen applied DBN model to predict the weekly exchange rate and compared with FFNN.

It also indicated that the predictive effect of DBN could reach 63% better than 41% of FFNN [23].

Xiong used LSTM to predict the volatility of the S&P index with the Google trend and economic variables as input. The result showed that the map of LAPM was smaller than the linear model and the autoregressive model [24]. Yu applied the deep neural network and LSTM to forecast the trading data of the Amazon stock. It was found that the effect of the deep neural network was better than LSTM, and the prediction accuracy was 54% [25]. Zhang used the improved convolution neural network and SVM to forecast the European and American market and the stock index and exchange rate in Hong kong market. The result showed that the hybrid model had the highest accuracy [26]. Gao used LSTM to forecast six kinds of different industries in the US stock market, using 359 stock features as input, with an average accuracy rate of 54.83% [27]. Sean used the recurrent neural network-LSTM, which was optimized by Bayesian, to predict the price of the coin, reaching 52% of accuracy [28]. Silva used the recurrent neural network (RNN) to forecast the prices of the three stocks. When used the historical data and economic variables as input, he found that the forecast price fitted the actual price better [29].

The input of these models is the basic index based on the composite index or the economic variable of the technical index, so the dimension of input data is not too high. In the literature above, the comparison baseline is based on the traditional machine learning algorithm, in order to prove the effectiveness of those models. Most of the selected objects are more mature stock markets such as the S&P index and the Nasdaq index. Instead, the minority of them are immature stock markets such as the Shanghai stock market, which prediction thereby is not distinct. In the deep learning, the LSTM is used in natural language processing and other serial data because of its unique memory function, but there are not many predictions for stock time series. This paper uses the deep LSTM to obtain useful information from the stock time series and try to predict the immature stock market.

3 Stock forecasting model based on long short-term memory neural network for IMMT

3.1 Stock vector

In natural language processing, there is a large number of unlabeled texts. When computers process text, it is necessary to convert the text into a format that the computer can understand. Therefore, researchers proposed the "word vector," which uses a series of numbers to express the word. The simplest method is the one-hot vector, which represents the position where the word appears as 1, and the rest is expressed as 0. But this method has two major drawbacks:

1. The dimension of the word vector is equal to the size of the dictionary. But the size of the dictionary is so large that the dimension of the vector is large, thereby making calculation inconvenient.
2. This representation cannot reflect the similarity between words, so it is not much of help for text processing and context semantic analysis.

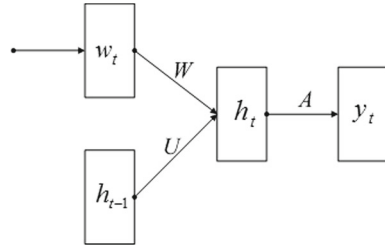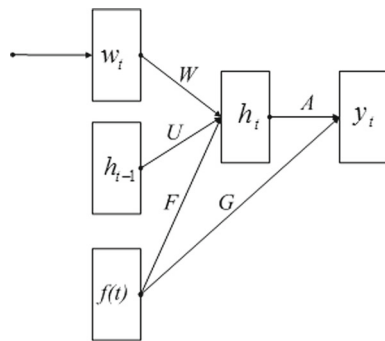**Fig. 1** Recurrent neural network-based language model

Fig. 2 RNN structure of feature layer

In order to fully extract information from the corpus, researchers introduced new machine learning methods, which mainly include RBM, neural network and the correlation between word and context [2].

For a specific language model, given a word sequence $(w_1, w_2, \ldots w_T)$, hidden layer sequence $(h_1, h_2, \ldots h_T)$ and the output layer sequence $(y_1, y_2, \ldots y_T)$ in RNN are described in Eq. (1).

$$h_t = \tanh(W w_t + U h_{t-1} + b_h)$$
$$y_t = A h_t + b_y \tag{1}$$

In Fig. 1, w_t is the one-hot vector of the t word, and $\theta = \{b_h, b_y, W, U, A\}$ is the set of learning parameters. The embedded vector of the word is the matrix W. However, RNN in learning process is prone to exploding and vanishing gradients. For this reason, the extended model of RNN was proposed [3], which added a feature layer to the original structure. The structure is shown in Fig. 2.

In Fig. 2, the hidden layer can be expressed as

$$h_t = \tanh(W w_t + U h_{t-1} + F f(t) + b_h) \tag{2}$$

When the feature layer $f(t)$ is added, the loss information is not attenuated to 0; thus, the vanishing gradient problem is avoided.

In the past stock market forecasting, the input is often a single stock or a single index. Thus, the input dimension will not be too large. However, when the input is historical data of multiple stocks, the input dimension will increase to thousands or even millions. At this time, if we use this original information to predict the stock market directly, it

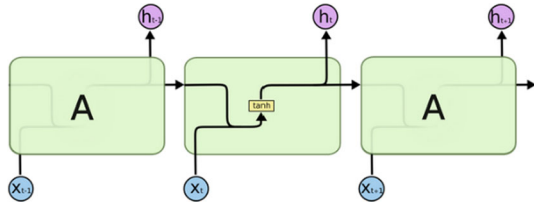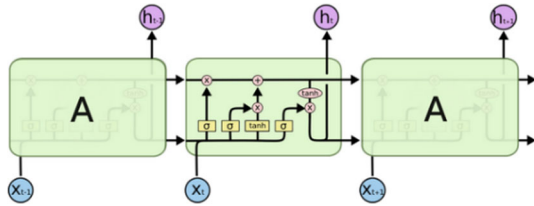**Fig. 3** Standard RNN module diagram

Fig. 4 LSTM module diagram

may result in greater errors due to the impact of information redundancy and irrelevant information.

Therefore, we introduce the concept of vectorization for stock market, named stock vector. The stock vector refers to the idea of word vector. First of all, the dimension of the stock vector is reduced, and then it is expressed in a low dimension space. Finally, the stock market is forecasted by stock vector. Based on the method of word vector, two kinds of models of reducing dimension and predicting stock price are proposed: the long short-term memory neural network with embedded layer (ELSTM) and the long short-term memory neural network based on automatic encoder (AELSTM).

3.2 The deep long short-term memory neural network with embedded layer based on stock vector (ELSTM)

Since the RNN is prone to exploding and vanishing gradients during the training process, the gradient cannot be passed down in long sequence. As a result, RNN cannot capture the effect of long distance. Therefore, LSTM, an improved RNN, is used to predict the stock sequence. The repeating module in a standard RNN contains only a single layer such as the tanh layer. But the LSTM has four layers, which are shown in Figs. 3 and 4, respectively.
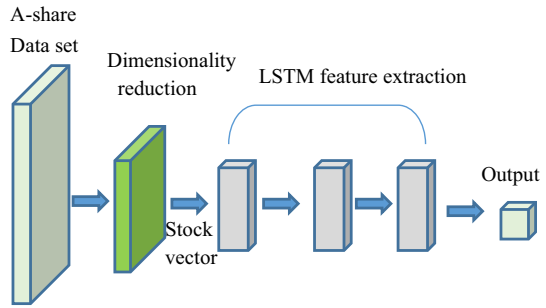
3.2.1 LSTM algorithm

In the timing model above, the core module of network is LSTM.

(1) The first step in our LSTM is to decide what information we are going to throw away from the cell state.

$$f_t = \sigma \left(W_f \cdot \left[h_{t-1}, x_t \right] + b_f \right) \tag{3}$$

A-share Data set

Dimensionality reduction

LSTM feature extraction

Stock vector

Output

(2) The next step is to decide what new information we are going to store in the cell state.

$$i_t = \sigma \left(W_i \cdot [h_{t-1}, x_t] + b_i \right) \tag{4}$$

$$C_t = \tanh \left(W_{\tilde{C}} \cdot [h_{t-1}, x_t] + b_C \right) \tag{5}$$

(3) It is now time to update the old cell state into the new cell state. The previous steps already decided what to do, we just need to actually do it.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

(4) Finally, we need to decide what we are going to output. This output will be based on our cell state, but will be a filtered version.

$$o_t = \sigma \left(W_o \left[h_{t-1}, x_t \right] + b_o \right) \tag{7}$$

$$h_t = o_t * \tanh (C_t) \tag{8}$$

3.2.2 Construction of ELSTM

The opening time of each stock in the A-shares market is not the same, which makes the stock gap too big. For example, the opening time of Dongfeng Motor and China International Trade is in 1999, Crystal Technology and the Ming shares is in 2014. In order to align the data, the missing data are padded with 0, so there will be some sparse submatrices. At the same time, in order to reduce the dimension of data and filter the irrelevant information, we add the embedded layer in front of the LSTM layer to achieve precisely stock prediction. The structure of the entire network is shown in Fig. 5.

It can be seen from the figure above that the main part of the framework is embedded layer and LSTM layer. The working principle of each layer is as follows:

(1) Embedded layer. The embedded layer is initialized to random matrix. We convert high-dimensional data into low-dimensional data by matrix transformation. In Fig. 5, the A-shared dataset is converted to a stock vector that is the input of neural network. In the whole model training process, the matrix in the embedded layer is trained as the parameters of the network. The training matrix is the prerequisite
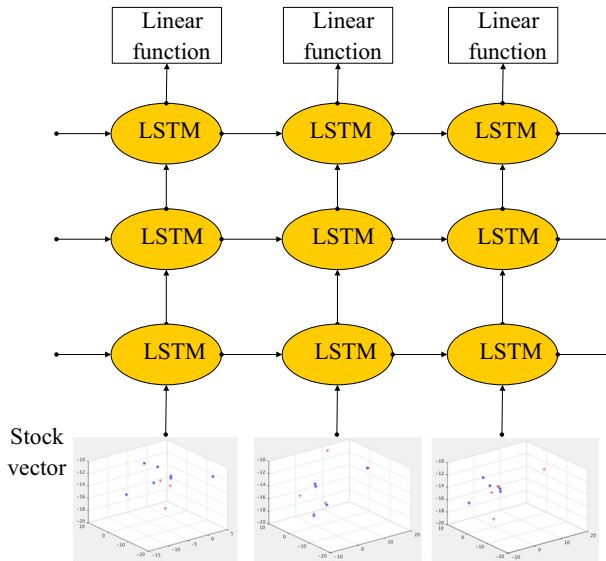
Fig. 6 Timing diagram of LSTM prediction

and basis of further feature extraction. Throughout the error back propagation (EBP) process, the training goal is minimum error.

(2) LSTM layer. After the reduced dimension processing in the embedded layer, we got the stock vector. The LSTM circularly reads the stock vector and further extracts the feature information. Then, it predicts the stock value and compares it with the standard value. Finally, it updates the parameters via EBP in the LSTM model.

(3) Output layer. For the regression prediction, the output layer has a single neuron. While the direction of ups and downs forecasts, the output layer has two neurons.

Figure 6 shows the timing structure of the LSTM module. The stock vector is transferred to the LSTM of the three layers, and the output sequence is calculated iteratively. Finally, the output is processed by the activation function to get the result. The input of each LSTM is not only affected by a lower layer, but also receives the effect of the same layer output, taking full account of the characteristic of time series. When regression prediction has been performed, the last layer uses a linear function to predict the output. While the ups and downs are predicted, the classification function is *logistic*.

3.3 Long short-term memory neural network with automatic encoder based on stock vector (AELSTM)

3.3.1 Optimized automatic encoder with continuous restricted Boltzmann machine

The value of each node of the automatic encoder is {0,1}, and the whole network is adjusted by the random gradient descent. The training result depends on the selection of

Fig. 7 Pre-training process of automatic encoder
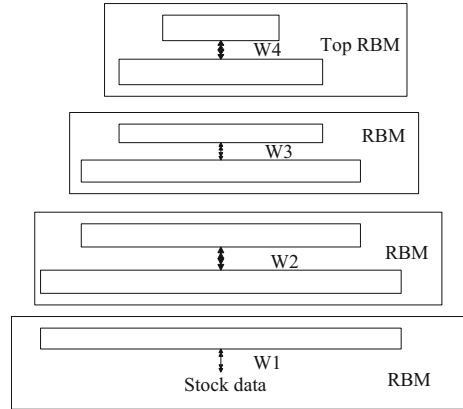
the initial weight. If the initial value is too large, it is easy to fall into local optimization. If the selection is too small and the gradient descent is very slow, the updated weight will slow down, resulting in increasing training time. Therefore, it is not suitable for dimensionality reduction of continuous stock data. In this paper, we use the automatic encoder with continuous restricted Boltzmann machine (CRBM) for the stock data. Every two layers of the network learn the initial weight with CRBM, and the top linear-RBM is aimed to output the stock data after the dimensionality reduction. This is a pre-training process, and the structure is shown in Fig. 7.

3.3.2 The working principle of CRBM network

For lower CRBM, the input of each node in the hidden layer is related to states of all nodes in the visual layer. We assume that $\{s_i\}$ is the state set of the visual layer, $\{s_j\}$ is the state set of the hidden layer, and the φ_j is sigmoid activation function.

$$s_j = \varphi_j \left(\sum_i w_{ij} s_i + b_i \right) \tag{9}$$

Hidden layer node state is still {0,1}, and the specific value is as follows:

$$s_j = \begin{cases} 0, s_j < U(0, 1) \\ 1, s_j > U(0, 1) \end{cases} \tag{10}$$

The $U(0, 1)$ represents the uniform distribution in (0,1). For the top linear-RBM, a random continuous unit of Gaussian noise is added to the visible layer:

$$s_j = \varphi_j \left(\sum_i w_{ij} s_i + b_i + N_j(0, 1) \right) \tag{11}$$
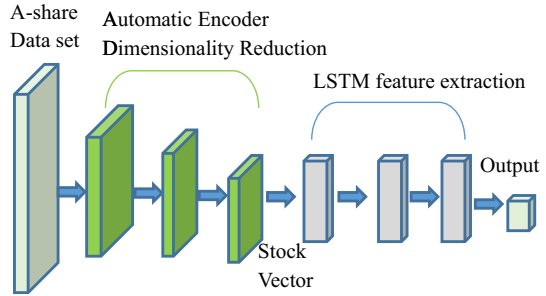
Fig. 8 AELSTM network structure diagram

Table 1 Detailed experimental environment parameters

Parameter name	Parameter value
RAM	62G
CPU	Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30 GHz
Hard disk	5T
OS	Centos 64 bit
Python	Python 2.7.9
MATLAB	MATLAB R2014a
Eclipse	Release 4.6.0
MySQL	5.1.73

The $N_j(0, 1)$ is a Gaussian random variable with a mean of 0 and a variance of 1, and the activation function is a linear function. In the experiment, the value of each RBM visual layer is [0,1]. The input of the top RBM is activation probability in the hidden layer of the previous RBM. Apart from top RBM, the other RBM hidden layer nodes are random binary values.

The AELSTM model is formed by the input of the stock vector which has reduced dimension by automatic encoder to the LSTM. The LSTM module has the same algorithm as the LSTM in the ELSTM model. We use AELSTM to learn the characteristics of the data and then predict the output. The overall structure is shown in Fig. 8.

4 Experimental results and analysis of the model for IMMT

4.1 Experimental preparations

This experiment uses Java, MATLAB, Python and other languages to achieve the installation of the corresponding environment on the server. Among them, MATLAB is used for the dimensionality reduction algorithm of automatic encoder; Java is used to process the experimental dataset; based on the Theano framework, Python is used to train RNN. Datasets are stored in a MySQL database. Detailed experimental environment is shown in Table 1. The specific process of languages usage is shown in Fig. 9.

Fig. 9 Specific process of languages usage
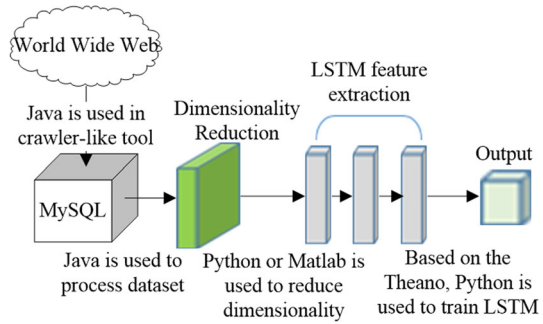
4.1.1 Web crawler

This paper designs a crawler, named StockCrawler, to crawl stock data on the Web sites (such as SINA finance and economics, RoyalFlush finance and economics, Stockstar), involving stock data, stock news, capital stock and shareholders, and financial analysis, etc. The architecture of Web crawler is shown in Fig. 10. It mainly contains three modules: URL management module, parsing module and storage module:

(1) The basic workflow of the URL management module is as follows:
    1) Select the seeds of URL to be crawled;
    2) Put these URLs into the queue of URL to be crawled;
    3) Take out the URLs from the queue to be crawled, then resolve the DNS and download the webpage corresponding to the URL and save the downloaded webpage to webpage library. In addition, put these URLs into the crawled queue of URL.
    4) Analyze the URL in the crawled queue, then put the URLs which meet the requirements into the queue to be crawl and go to the next cycle. Besides, iterate it until the crawl process has been completed. The depth-first traversal strategy is applied in the whole process.
(2) Parsing module: This module analyzes the crawled information of form and webpage, in order to extract the stock information to meet the requirements. If the analysis is completed, the data will be stored in the database.
(3) Storage module: The historical stock data, stock news, industry information and other related information are saved to the corresponding table in the MySQL database.

4.1.2 Model inputs

This article crawls the relevant information of stock via crawler technology, and the summarized information is shown in Table 2. In this paper, the stock-related data tables are the stock list and the historical stock data table. Specifically, the stock list includes stock code, opening date, company address, turnover and other related information of the Shanghai A-share market and Shenzhen stock market. Rather, the historical stock data table contains the information of all the stocks in the stock list such as the opening
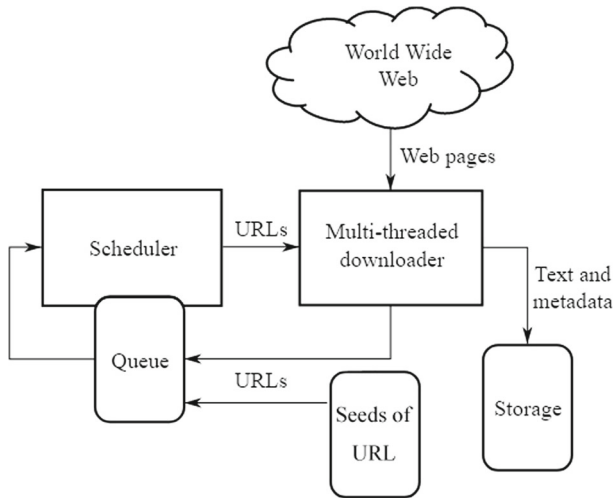
Fig. 10 Architecture of Web crawler

Table 2 Stock market statistics

Table name	Data size (row)
News_newslist	1,222,135
News_reportlist	127,521
News_sourcelist	13
News_tradeinfolist	374,542
Stockhangyelist	2797
Stockhistorydata	7,902,686
Stocklist	2691

price, the highest price, lowest price, closing price, transaction date, volume and so on.

We use crawler technology to grab information about the stock. This article includes two experiments:

(1) We filter the historical data of single stock on the Shanghai A-share market, including the opening price, the highest price, the lowest price, the closing price and the volume. With them, it is possible to predict the price and trend of the Shanghai A-share composite index.

(2) Based on the original five indicators, we add some additional indicators, such as daily amplitude, 5-day amplitude, 10-day amplitude and amplitude of fluctuation of the volumes, amounting to a total of 9, so as to predict the price and trend of a single stock.

The data of January 1, 2006–October 19, 2016, for 10 years were selected as the object of study, 70% of which were training set, 10% were validation set, and 20% were test set. First, the dimensions of the input dataset are aligned, and then the data are normalized on [0, 1]. The specific description is as follows:

$$x^* = \frac{x - \min}{\max - \min} \tag{12}$$

The max is the maximum value of the sample, and the min is the minimum of the sample, so that the x is mapped to [0, 1].

4.1.3 Model output

In order to analyze the effect of the model more comprehensively, this paper evaluates the performance of the model for short-term stock price forecast from two aspects.

(1) The mean square error (MSE): the expected value of the difference between the predicted and the true values, calculated as Eq. 13.

$$\text{MSE} = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N} \tag{13}$$

The y_i denotes the real stock value, and the $\hat{y}_i$ is the predictive value.

(2) Data accuracy (DA): When predicting the ups and downs, the stock dataset is divided into two categories. Specifically, 1 represents the rise and 0 represents decline or constant. The class labels and accuracy are depicted in Eq. 14 and Eq. 15, respectively:

$$y = \begin{cases} 1, & (y_{i+1} - y_i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

$$\text{DA} = \frac{1}{N} \sum_{i=1}^{N} a_i \tag{15}$$

The y_i denotes the actual value, and a_i represents the same number of predicted and actual values.

4.2 Visualization of stock vector

On the basis of these two experiments, we make a visual analysis of the stock vector in the ELSTM model and observe the morphological characteristics of the stock data. The structures of dimensionality reduction are 5325-2000-800-200-3 and 9585-5000-1000-200-3. The results are shown in Figs. 11 and 12.

In these two diagrams above, + (red) corresponding to the class label denotes down, and * (blue) denotes up. It can be seen from the figures mentioned above that instead of the image dataset which can be separated from each other, the daily ups and downs of the stock are intertwined, indicating the uncertainty about the daily trend of stock price. Compared with Figs. 10 and 11, it can be concluded that when the input dimension increases, the points after dimension reduction are clustered more closely, and the span becomes smaller.
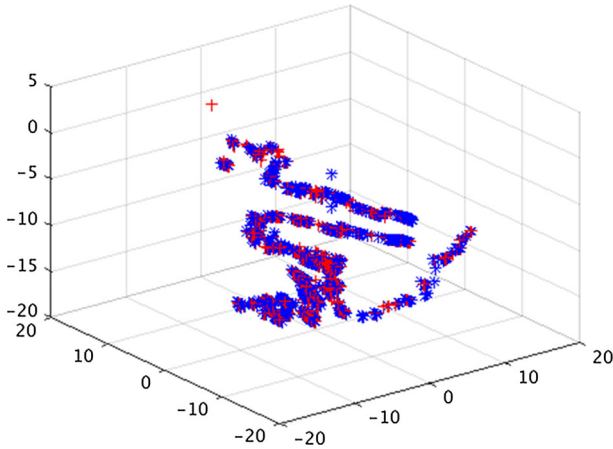
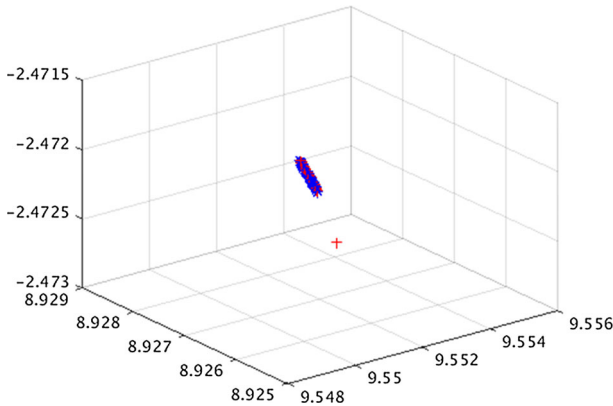**Fig. 11** Input data visualization of Shanghai A-share composite index

Fig. 12 Input data visualization of Sinopec

On the one hand, it portrays that the more dimension that are high, the stable the input state will be. On the other hand, it also manifests that the information displayed by the high-dimensional input is not as rich as the information of the low dimension, which indirectly indicates that the majority of high-dimensional data are loss after dimension reduction. We have witnessed the same results in our experiment. The mean variance loss after dimension reduction is 29 in Fig. 11, while it is 238 in Fig. 12.

4.3 Comparison and analysis of models

The models of AELSTM, ELSTM, DBN, DBN-MLP and multilayer perceptrons (MLP) are, respectively, experimented on two datasets. The results are shown in Figs. 13, 14, 15 and 16.

Fig. 13 Comparative MSE of Shanghai A-share composite index
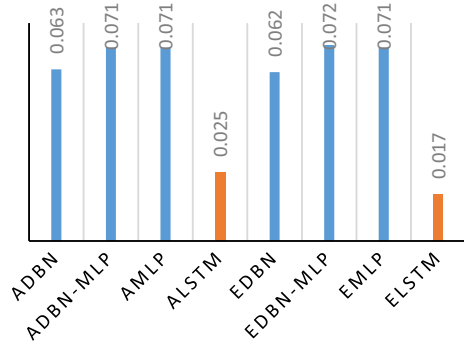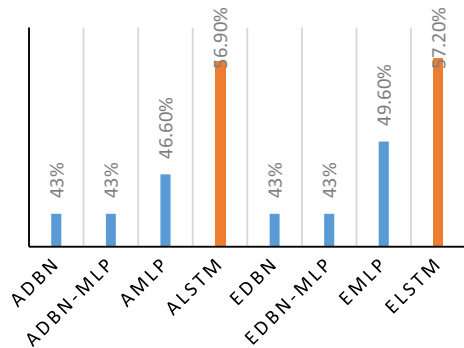

Fig. 14 Comparative accuracy of Shanghai A-share composite index

The model with A as the initials in Figs. 13 and 14 represents the predictive model after dimensionality reduction by automatic encoder, while the model with E signifies the predictive model of the original input data.

According to Figs. 13 and 14, it can be seen that the MSE and accuracy of the two kinds of LSTM are the best. Specifically, the accuracy is about 10% higher than other models and approximately 7% higher than the stochastic prediction. In additions, the MSE reaches the minimum. In the case of two kinds of input, when it comes to the A-share composite index, the predictive accuracy of the shallow learning MLP is better than the DBN of deep network and DBN-MLP. While they are lower than the stochastic prediction. The MSE of the six contrast models is as the same overall. In general, the model with initials E is slightly better than the models with A.

The accuracy of the Shanghai A-share composite index by LSTM is 57%, the same with literature [13] on the S&P 500. From the literature [1], we can see that the market forecast accuracy is affected by market maturity. The more mature the market is, the higher the accuracy rate is. Apart from it, the factors can be listed as follows: (1) Choice of models is not always relevant; (2) theoretical assumptions are not consistent with real-life stock brokerage and data; (3) behavioral patterns are not fully considered, because some results (in stock) are not logically but sentimentally driven, similar to what the 2017 Nobel Laureate in Economics has demonstrated; (4) models, computation, data processing and analysis are not quick enough to rectify. According to Fig. 13, we also can see that the accuracy varies between 43 and 58%.

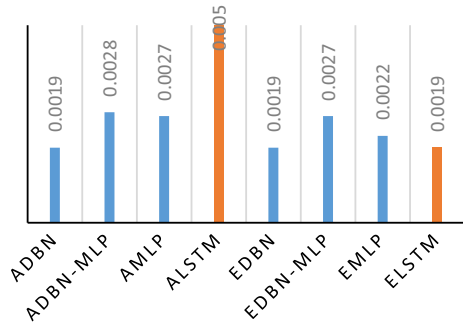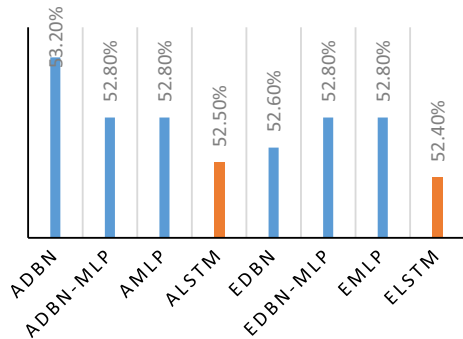Fig. 15 Comparative MSE of Sinopec

Fig. 16 Comparative accuracy of Sinopec

The AELSTM and ELSTM are better than traditional neural network. In fact, the Chinese stock market is less mature than the USA. Fortunately, we took the Chinese market and index into account, and the effectiveness of the method used in this paper is still demonstrated better than other competing methods to some extent.

It is easy to draw a conclusion from Figs. 15 and 16 that such models above have minor differences in accuracy. To be more specific, the best model is ADBN, reaching 53.20%, while the worse is ELSTM, about 52.40%. Although there is an information loss in the process of dimension reduction, the accuracy of prediction after dimension reduction is not always worse than that of the original input. As for all models, the accuracy is always higher than 50%, which indicate that the input information of two kinds of stock in this paper does improve the accuracy of Sinopec.

The AELSTM model for Sinopec is not dominant. Probably in that Sinopec stock is vulnerable to external factors. For example, stock suspension will result in instability in stock prices. Therefore, it has failed to demonstrate the characteristics of time series. In fact, it is inevitable to avoid information loss for stock vector in the process of dimensionality reduction. Hence, it is unsuccessful to utilize the advantages of AELSTM model.

Taking into account the results of predicted Shanghai A-share composite index and Sinopec, the results are more stable with the ELSTM method, since it will not fluctuate greatly due to the randomization of the initial weights. Therefore, we choose ELSTM model to conduct empirical analysis of the stock. The ELSTM on the Shanghai A-share and the Sinopec forecast trend is shown in Figs. 17 and 18, respectively.
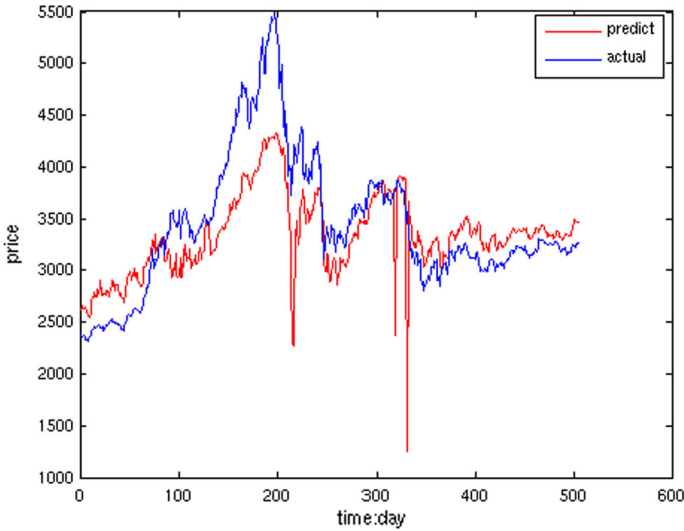
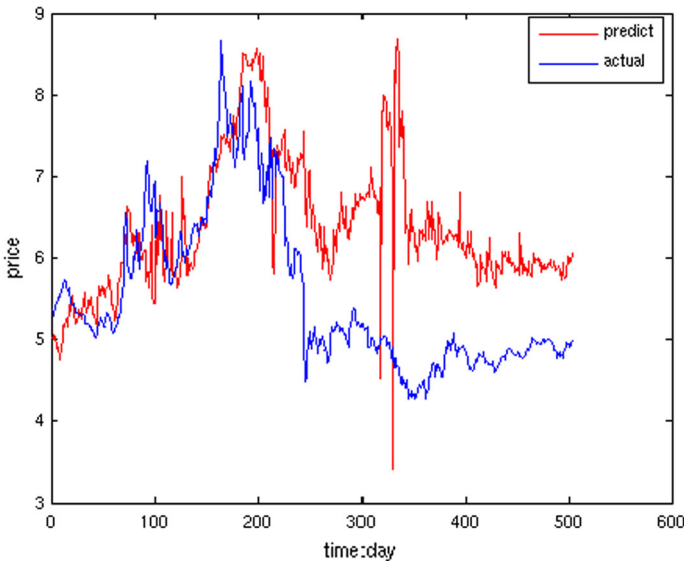**Fig. 17** Comparison of predictive value and true value of the Shanghai A-share composite index

Fig. 18 Comparison of predictive value and true value of the Sinopec

It can be seen from Fig. 17 that the waveform of the predictive curve and the actual curve for Shanghai A-share composite index are mostly consistent. Between 100 and 300 days, the peaks of the two curves are the same, while the small fluctuation in other states is kept in a similar state. From Fig. 18, it is depicted that the actual waveform of Sinopec and the predicted are also similar to a degree. Nevertheless, the price gap

between 200 and 500 is comparatively larger. Thus, the effect is not as good as the Shanghai A-share composite index.

In summary, the individual stock is sensitive to external factors such as the suspension of the stock market and some error of the training data, which affected the test results. The Shanghai A-share composite index is determined by A-share data rather than a certain stock. Hence, predictive effect is relatively satisfying.

4.4 Empirical analysis of different stocks

In order to further verify the effectiveness of the model, we randomly selected two different stocks on the Shanghai A-share market for empirical analysis, which are the Tianjin Marine Shipping (TMSE) (600751) and TBEA (600089). We select daily closing price of the two stocks from January 2006 to October 2016 as a predicted target. The accuracy of the TBEA at different learning rates is shown in Fig. 19, and the accuracy of three stocks is shown in Fig. 20.

It can be seen from Fig. 19 that the change in learning rate has little effect on the accuracy of TMSE prediction. As a matter of fact, the best effect among those models is just 48.2%. It is mainly resulted from the instability of the stock itself. The whole dataset is suspended for a long time, which has a great impact for stock learning, so the predictive effect is not ideal. And it is also the reason why the composite index is selected when we forecast the stock market. According to Fig. 20, we can learn that the model has the best effect of classification in TMSE reaching 58.9%, the worst effect is TBEA reaching 48.2%, and the average accuracy of three stocks is 53.2%.

The accuracy is affected by the maturity of the market, and the stock is selected randomly in this paper. When the selected stocks are affected by some factors, the range of fluctuation may be large, which will have some influence on the result. On the basis of this, the average accuracy of the three stocks in the Shanghai stock market is 53.2%. In addition, the accuracy of the stocks in the US market in the literature [3] is 54%. By contrast, the effectiveness of the proposed method in the stock forecast can also be verified to some extent.

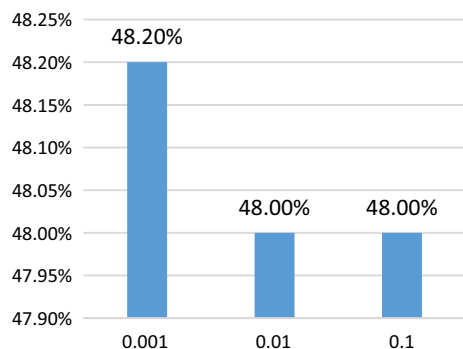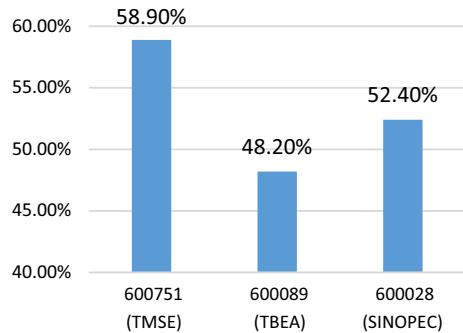**Fig. 19** Accuracy of TBEA at different learning rates

Fig. 20 Predictive accuracy of the three stocks

4.5 Stock analysis and research contributions for IMMT

IMMT requires methods, algorithms and steps to extract and process data, so that large-scale simulations can be performed [30,31]. Outputs of analysis can be presented in visualization and analytics to allow anyone to understand the implications and meanings of research outputs. This is highly essential for stock analysis and forecasting, since results can be followed, understood and even adopted by investors, clients and users of financial services.

We fully demonstrate research contributions in IMMT as follows. First, the data can be extracted and processed. Second, our LSTM neural network can perform financial analysis and forecasts. Third, our work supports analytics and visualization, a core component in IMMT.

5 Conclusion and future work

In this paper, the LSTM neural network with embedded layer and the long short-term memory neural network with automatic encoder are proposed on the basis of LSTM neural network. Firstly, we verify the performance of the models in Shanghai A-share composite index and Sinopec. Then we use a better ELSTM model for other selected stocks. The average accuracy of the three stocks is 53.2%, and the A-share composite index is 57%, which are higher than the stochastic forecast. In summary, the methods mentioned in this paper have better predictive performance for the Shanghai A-share composite index. We also explain how our work is relevant and related to IMMT for financial analysis. This is a pioneering research since we have blended different techniques and algorithms together with the cross-disciplinary approach.

Although the models can improve the predicted effect of Shanghai A-share composite index to a certain extent, there are still some deficiencies in the input of historical data. The text information in the stock market such as news is not fully utilized. So in the next step, we will consider adding text information factor to the model to further improve the performance. In addition, our model is aimed at Chinese stock market forecasting without involving the European and American stock market. In practice, it is also our next research direction to expand our research datasets and predict whether

our models are equally applicable in European and American stock market with a high accuracy.

References

1. Hsu MW, Lessmann S, Sung MC et al (2016) Bridging the divide in financial market forecasting: machine learners vs. financial economists. Expert Syst Appl 61:215–234
2. Chen E-H, Qiu S-Y, Xu C et al (2014) Word embedding—continuous space representation of natural language. Data Acquis Process 29(1):19–29
3. Mikolov T, Zweig G (2013) Context dependent recurrent neural network language model. In: Spoken Language Technology Workshop. IEEE, pp 234–239
4. Chen C et al (2017) A rear-end collision prediction scheme based on deep learning in the internet of vehicles. J Parallel Distrib Comput. https://doi.org/10.1016/j.jpdc.2017.08.014
5. Chauhan R, Kaur H, Chang VJ (2017) Advancement and applicability of classifiers for variant exponential model to optimize the accuracy for deep learning. J Ambient Intell Hum Comput, https://doi.org/10.1007/s12652-017-0561-x
6. Niaki STA, Hoseinzade S (2013) Forecasting S&P 500 index using artificial neural networks and design of experiments. J Ind Eng Int 9(1):1–9
7. Adhikari R, Agrawal RK (2014) A combination of artificial neural network and random walk models for financial time series forecasting. Neural Comput Appl 24(5):1441–1449
8. Yu H, Chen R, Zhang G (2014) A SVM stock selection model within PCA. Proc Comput Sci 31(31):406–412
9. Zbikowski K (2015) Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. Expert Syst Appl 42(4):1797–1805
10. Ballings M, Van den Poel D, Hespeels N et al (2015) Evaluating multiple classifiers for stock price direction prediction. Expert Syst Appl 42(20):7046–7056
11. Oztekin A, Kizilaslan R, Freund S et al (2016) A data analytic approach to forecasting daily stock returns in an emerging market. Eur J Oper Res 253(3):697–710
12. Gerlein EA, McGinnity M, Belatreche A et al (2016) Evaluating machine learning classification for financial trading: an empirical approach. Expert Syst Appl 54:193–207
13. Zhong X, Enke D (2017) Forecasting daily stock market return using dimensionality reduction. Expert Syst Appl 67:126–139
14. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504
15. Kuremoto T, Kimura S, Kobayashi K et al (2014) Time series forecasting using a deep belief network with restricted Boltzmann machines. Neurocomputing 137:47–56
16. Kuremoto T, Obayashi M, Kobayashi K et al (2014) Forecast chaotic time series data by DBNs. In: 2014 7th International Congress on Image and Signal Processing (CISP). IEEE, pp 1130–1135
17. Takeuchi L, Lee YYA (2013) Applying deep learning to enhance momentum trading strategies in stocks. http://cs229.stanford.edu/proj2013/TakeuchiLeeApplyingDeepLearningToEnhanceMomentumTradingStrategiesInStocks.pdf
18. Zhu C, Yin J, Li Q (2014) A stock decision support system based on DBNs. J Comput Inf Syst 10(2):883–893
19. Chang V (2014) The business intelligence as a service in the cloud. Future Gener Comput Syst 37(2014):512–534
20. Sharang A, Rao C (2015) Using machine learning for medium frequency derivative portfolio trading. Preprint arXiv:1512.06228

21. Batres-Estrada B (2015) Deep learning for multivariate financial time series. KTH Royal Institute of Technology, Stockholm
22. Yeh SH, Wang CJ, Tsai MF (2015) Deep belief networks for predicting corporate defaults. In: 2015 24th Wireless and Optical Communication Conference (WOCC). IEEE, pp 159–163
23. Shen F, Chao J, Zhao J (2015) Forecasting exchange rate using deep belief networks and conjugate gradient method. Neurocomputing 167(C):243–253
24. Xiong R, Nichols EP, Shen Y (2015) Deep learning stock volatility with google domestic trends. Preprint arXiv:1512.04916
25. Yu F-F (2016) Based on the neural network of stock forecast [D]. Harbin Institute of Technology, Harbin
26. Zhang G-Y (2016) Application of improved convolution neural network in financial forecasting. Zhengzhou University, Henan
27. Gao Q (2016) Stock market forecasting using recurrent neural network. University of Missouri, Columbia
28. McNally S (2016) Predicting the price of Bitcoin using machine learning. National College of Ireland, Dublin
29. Silva IND, Spatti DH, Flauzino RA et al (2017) Forecast of stock market trends using recurrent networks. In: da Silva IN, Spatti DH, Flauzino RA, Liboni LHB, Alves SFR (eds) Artificial Neural Networks. Springer International Publishing, pp 221–227
30. Chang V (2017) Towards data analysis for weather cloud computing. Knowl Based Syst 127:29–45
31. Chang V (2017) Data analytics and visualization for inspecting cancers and genes. Multimed Tools Appl, https://doi.org/10.1007/s11042-017-5186-8