A SLFN MODEL WITH R-ELM FOR STOCK PRICE FORECASTING

Yuyang Zhu,* Linxian Zhi,* and Weiduo Zhang**

Abstract

In this paper, a single-layer feedforward neural network (SLFN) model with tapped delay units at input layer is proposed to predict the daily closing price of the S&P 500 index. It is shown that by adding tapped delay units at the input layer, the dynamics in financial data can be captured effectively. In addition, training with the regularized extreme learning machine (R-ELM) method, not only the fast training speed is ensured but also the robustness of the neural model is improved. The performance of the proposed model is evaluated on the S&P 500 index raw dataset. The comparisons with backpropagation neural network (BP-NN), linear regression (LR), ARIMA and GARCH are conducted. The simulation results validate the efficiency and effectiveness of the proposed model.

Key Words

Stock price forecasting, artificial neural network, R-ELM, tapped delay units

1. Introduction

The analysis and prediction of the financial market (such as the stock market) are crucial to financial investors for the formulation of investment strategy and investment risk diversification. Traditionally, fundamental analysis and technical analysis are two effective methods for modelling simple dynamic financial markets [1]–[3]. However, due to the complexity and turbulent data environment of the stock market system, the fundamental analysis method is difficult to make a correct judgement on the short-term market changes and cannot effectively analyse the impact of short-term speculation on the market [4]. On the other hand, the prediction accuracy and reliability of the technical analysis method are poor on the long-term trend of the market due to a large number of technical indicators [5]. Therefore, it is necessary to develop more intelligent models for capturing the complex dynamics of the stock market and enhancing prediction accuracy.

* Lishui University, Wenjin Street, Lishui, China; e-mail: 13625888200@139.com, zhilinxian@163.com

** Department of Technology Service, Zhejiang Rural Credit Union, Hangzhou, China; e-mail: zwd@zjrcu.com
Corresponding author: Linxian Zhi

In this paper, a single hidden layer feedforward neural network (SLFN) with tapped delay units at the input layer is developed to perform the 1-day-ahead closing price of the Standard & Poor's 500 (S&P 500) index. By using the regularized extreme learning machine (R-ELM) algorithm to train the model, the learning speed and the robustness are significantly improved. The simulation results demonstrate the proposed model is capable of capturing the dynamics of the stock market.

The S&P 500 index is a representative index for the stock market of the United States (US), as it is the capitalization-weighted index of the largest 500 listed companies in the US stock market. It is found that the movements of the S&P 500 are affected by many financial factors, such as the gold price, the crude oil price, the other countries' stock market's movements and the exchange rate. Many recent studies have been devoted to improving the prediction accuracy of the S&P 500 index [6]–[8]. In the proposed model, total of 39 financial factors grouped as 6 classes are considered as the inputs. In particular, considering the stock price is not only related to the current financial factors but also related to the historical financial factors and stock price, the tapped delay units are added to the input layer to represent the inputs with temporal pattern vectors. The advantage of tapped delay units is the dynamic features of the stock market can be sufficiently captured.

In order to overcome the limitations of backpropagation (BP) algorithm, Huang et.al proposed the extreme learning machine (ELM) learning scheme for training an SLFN [9]. As a supervised learning algorithm, ELM randomly assigned the input weights and hidden layer biases, and then the output weights can be determined by the Moore–Penrose generalized inverse of hidden layer outputs. Instead of fully tuning all the internal parameters of the BP neural network, the learning speed of the ELM is often extremely fast and it exhibits good generalization performance [10]–[11]. In recent years, ELM-based algorithms have been applied successfully in more and more practical applications [12]–[13]. However, it is noted that ELM-based SLFN models can be very sensitive to noises in the inputs. As a result, structural and empirical risks of the model can be high as output weights also become very sensitive to input noises [14]. By introducing the regularization term into the cost function, the optimized output

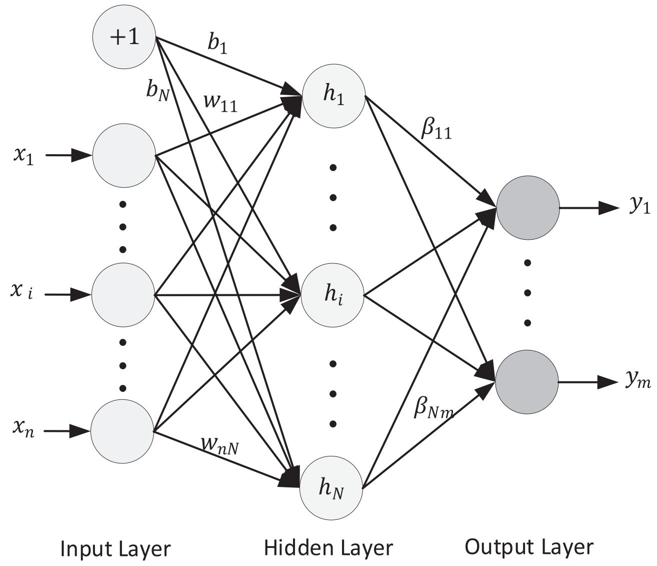

Figure 1. A SLFN model.

weights are penalized in the R-ELM algorithm [15]. In such a way, a smoother hyper-plane is generated by reducing the effect of sudden changes and outliners in the hidden feature space on the formation of the hyper-plane. It is shown that the R-ELM method has the capability of learning the behaviour of many non-linear systems [16]–[20]. Due to the complicated non-linear and dynamic features of the changes of the S&P 500 index, the R-ELM algorithm is employed to train the proposed neural model.

The rest of the paper is organized as follows: Section 2 presents the details of the ELM and R-ELM algorithms. Section 3 is devoted to the proposed ANN model with the R-ELM algorithm for stock price forecasting. In Section 4, the experimental results are discussed. Section 5 gives a conclusion of this work.

2. Extreme Learning Machine (ELM) and Regularized Extreme Learning Machine (R-ELM)

In Fig. 1, a SLFN consists of a output layer with m output nodes, a hidden layer with N hidden nodes and a input layer with n input nodes, where β_{ij}, for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, m$ are the output weights, w_{ij}, for $i = 1, 2, \ldots, n$ and $j = 1,\ 2,\ \ldots, N$ are the input weights, b_i, for $i = 1, 2, \ldots, N$ are the input biases. In order to train the SLFN model with the R-ELM algorithm, M distinct sample data pairs $(\boldsymbol{x}_i, \boldsymbol{d}_i)$, for $i = 1,\ 2, \ldots, M$ are used. Thus, the ith input vector is

$$\boldsymbol{x}_i = [x_{i1}, \ldots, x_{in}]^T\ , \text{ for } i = 1,\ 2, \ldots, M \tag{1}$$

and the corresponding ith target vector is

$$\boldsymbol{d}_i = [d_{i1}, d_{i2}, \ldots, d_{im}]^T\ , \text{ for } i = 1,\ 2, \ldots, M \tag{2}$$

2.1 ELM

In ELM, the input weights w_{ij} and hidden biases b_i are randomly selected from a certain value intervals to ensure the hidden neurons are all working within the linear region of the activation functions. Then, the hidden layer output matrix $\boldsymbol{H}$ can be expressed as:

$$\boldsymbol{H} = \begin{bmatrix} g\left(\boldsymbol{x}_1^T \boldsymbol{w}_1 + b_1\right) & \cdots & g\left(\boldsymbol{x}_1^T \boldsymbol{w}_N + b_N\right) \\ \vdots & \ddots & \vdots \\ g\left(\boldsymbol{x}_M^T \boldsymbol{w}_1 + b_1\right) & \cdots & g\left(\boldsymbol{x}_M^T \boldsymbol{w}_N + b_N\right) \end{bmatrix} \tag{3}$$

with b_i, for $i = 1, 2, \ldots, N$, and $\boldsymbol{w}_i = [w_{1i}, w_{2i}, \ldots, w_{ni}]^T$, for $i = 1, 2, \ldots, N$. $g(\cdot)$ is utilized as the activation function. Thus, the output matrix of the SLFN model can be computed as:

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{\beta} \tag{4}$$

where $\boldsymbol{Y} = \left[\boldsymbol{y}_1^T \ldots \boldsymbol{y}_M^T\right]^T$ and $\boldsymbol{\beta} = \left[\boldsymbol{\beta}_1^T \ldots \boldsymbol{\beta}_N^T\right]^T$.

Hence, the optimal output weight matrix $\boldsymbol{\beta}$ can be obtained through the batch learning linear least squares method with a Moore–Penrose pseudo-inverse as:

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{D} \tag{5}$$

where $\boldsymbol{H}^{\dagger}$ denotes the Moore –Penrose generalized inverse of hidden output matrix H and $\boldsymbol{D}$ is the training target with $\boldsymbol{D} = \left[\boldsymbol{d}_1^T \ldots \boldsymbol{d}_M^T\right]^T$.

It can be seen that ELM determines the output weights matrix $\boldsymbol{\beta}$ by the minimum norm least square solution to the linear system in (5). ELM obtain good generalization performance with dramatically increased learning speed by using Moore –Penrose inverse method [4].

2.2 R-ELM

In order to improve the robustness of the neural model against both internal and external disturbances, the R-ELM algorithm (*i.e.* batch learning type of least square method) is used for computation of the optimal output weight matrix of the neural model. In particular, the Lagrange multipliers are utilized as a constraint condition in the output weights optimization procedure. The optimization problem described above can be expressed as [8]:

$$\text{Minimize } \left\{\frac{\gamma}{2}\|\boldsymbol{\varepsilon}\|^2 + \frac{1}{2}\|\boldsymbol{\beta}\|^2\right\} \tag{6}$$

$$\text{Subject to } \boldsymbol{\varepsilon} = \boldsymbol{D} - \boldsymbol{Y} = \boldsymbol{D} - \boldsymbol{H}\boldsymbol{\beta} \tag{7}$$

where γ is a constant balancing parameter for adjusting the balance of the empirical risk and the structural risk. This problem can be solved by using the method of Lagrange multipliers:

$$\begin{aligned} L = {} & \frac{\gamma}{2}\sum_{i=1}^{N}\sum_{j=1}^{m}\varepsilon_{ij}^2 + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{N}\beta_{ij}^2 \\ & - \sum_{k=1}^{N}\sum_{p=1}^{m}\lambda_{kp}\left(\boldsymbol{h}_k^T\overline{\boldsymbol{\beta}}_p - d_{kp} - \varepsilon_{kp}\right) \end{aligned} \tag{8}$$

where ε_{ij} is the ijth element of the error matrix $\boldsymbol{\varepsilon}$, β_{ij} is the ijth element of the output weight matrix $\boldsymbol{\beta}$, d_{ij} is the ijth element of the target output matrix $\boldsymbol{D}$, $\boldsymbol{h}_k$ is the kth column of the hidden layer output matrix $\boldsymbol{H}$, $\overline{\boldsymbol{\beta}}_p$ is the pth column of the output weight matrix $\boldsymbol{\beta}$, λ_{kp} is the kpth Lagrange multiplier, γ is real positive regularization parameter and $\|\boldsymbol{\beta}\|^2$ is the regularization term of the output layer.

Differentiating L in (8) with respect to β_{ij}, the formula is expressed as:

$$\begin{aligned}\frac{\partial L}{\partial \beta_{ij}} = \beta_{ij} - \sum_{k=1}^{N} \lambda_{kj} h_{ki}^T = \beta_{ij} \\ - (\lambda_{1j} h_{1j} + \lambda_{2j} h_{2j} + \cdots + \lambda_{Nj} h_{Nj})\end{aligned} \tag{9}$$

Therefore, the output weight matrix $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} = \boldsymbol{H\lambda} \tag{10}$$

Differentiating L in (8) with respect to ε_{ij}, the formula is shown as:

$$\frac{\partial L}{\partial \varepsilon_{ij}} = \gamma \varepsilon_{ij} + \lambda_{ij} \tag{11}$$

Furthermore, the vector form of the (11) can be expressed as:

$$\boldsymbol{\lambda} = -\gamma \boldsymbol{\varepsilon} \tag{12}$$

Considering the constraint in (6), (12) can be expressed as:

$$\boldsymbol{\lambda} = -\gamma \left(\boldsymbol{H\beta} - \boldsymbol{D} \right) \tag{13}$$

Then, the optimal output weight matrix $\boldsymbol{\beta}$ can be computed as:

$$\boldsymbol{\beta} = \left(\frac{1}{\gamma} \boldsymbol{I} + \boldsymbol{H}^T \boldsymbol{H} \right)^{-1} \boldsymbol{H}^T \boldsymbol{D} \tag{14}$$

where $\boldsymbol{I}$ is a $N \times N$ unity matrix and γ is the regularization parameter.

The regularization term $\frac{1}{2}\|\boldsymbol{\beta}\|^2$ in (6) is set to reduce both empirical risks and structural risks and increase the robustness of the SLFN model. γ in (14) is the regularization parameter which controls the sensitivity of the output weights to the hidden layer outputs. In particular, when a small value of γ is used, the values of the optimized output weights become reluctant to large changes in the hidden outputs which might be caused by both internal and external disturbances. On the other hand, a large γ means the constraint is weak, and hence the output weight is sensitive to any outliners in the hidden feature space. When $\gamma \to \infty$, the R-ELM is equivalent to ELM.

3. The Proposed Model

In this work, a SLFN model with tapped delay units at the input layer is developed to predict the daily closing price of the S&P 500 index. In order to incorporate as many impact factors of the S&P 500 index as possible, total 39 financial factors grouped as 5 classes are treated as inputs in this study. The structure of the modified SLFN model is shown in Fig. 2.

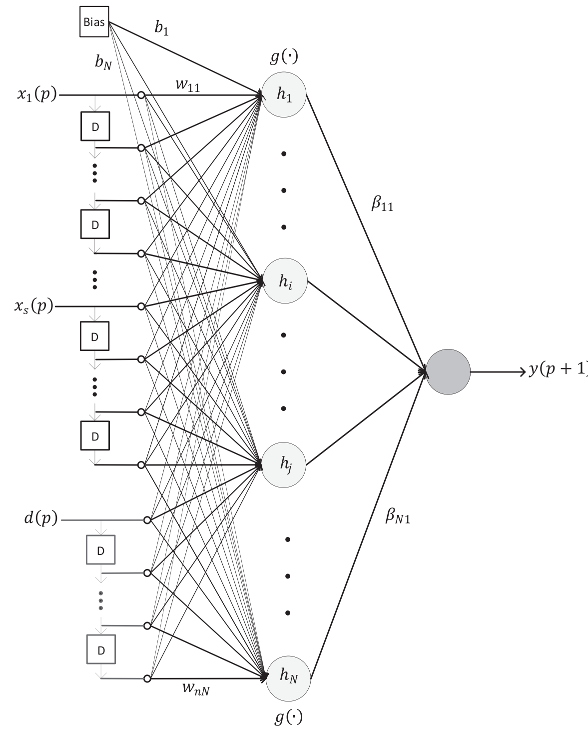

Figure 2. The proposed ANN model.

In Fig. 2, 1-day ahead daily closing price of the S&P 500 is regarded as the output of the neural network model, N is the number of the hidden nodes, the input layer has $s+1$ tapped delay lines with total n input nodes, w_{ij}, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, N$, are the input weights, β_j, for $j = 1, 2, \ldots, N$, are the output weights, b_i, for $i = 1,\ 2, \ldots,\ N$, are the bias of hidden layer. The input vector is denoted as following:

$$\begin{aligned}\boldsymbol{x}(p) = [x_1(p), \ldots, x_1(p - n_1 + 1), \ldots, x_s(p), \ldots, \\ x_s(p - n_s + 1), d(p), \ldots, d(p - n_0 + 1)]^T\end{aligned} \tag{15}$$

where $x_i(p)$ is the ith factors at pth day, $d(p)$ is the historical closing price at pth day, $n = n_0 + n_1 + \cdots + n_s$ is the number of inputs, $n_0, n_1, \ldots, n_s$ are the length of unit delays of each factor and s is the number of total factors. $y(p+1)$ is the corresponding output, the corresponding target output vector $\boldsymbol{d} = [d(p+1), d(p), \ldots, d(p-M)]^T$. By feeding the M input vectors into the model, the corresponding output vector $\boldsymbol{y} = [y(p+1), y(p), \ldots, y(p-M)]^T$ and can be computed as:

$$\boldsymbol{y} = \boldsymbol{H\beta} \tag{16}$$

Remark 1. *According to the R-ELM algorithm, the input weights and biases are randomly generated. In this work, the sigmoid function is used as the hidden layer activation function due to its boundedness and rich dynamics. In order to improve the model's hidden neurons activity level, the input weights and biases need to be selected from a small value range, for example,* 0 *to* 1*. As a result, the output of hidden nodes is highly responsive to subtle changes in the input data. Such design of the hidden neurons' activeness is mainly due to the movements of stock prices being sensitive to the changes of other financial factors or events.*

Remark 2. *It is noted that the ELM algorithm performs extremely fast learning speed and global optimization. In using the batch learning type of R-ELM, the following advantages can be achieved: (i) the local minima in recursive learning algorithms is avoided; (ii) the global optimization in the parameter space can arrive and (iii) the learning speed is extremely fast. In financial modelling and prediction applications, timely and robust predictive models are sought after due to the fast changing and noisy nature of the market environment. Under such circumstances, R-ELM-based models are considered a suitable approach for this type of problem.*

4. Experimental Validation and Comparison

4.1 Data Description

The S&P 500 index is not only a financial indicator for individual investment but also an important economic indicator for the global economy. As the development of global economy, the volatility of S&P 500 is affected by other countries' stock market's movements such as China and Japan. Therefore, including as many impact factors and technical indicators as possible is more sufficient to the optimization of the proposed ANN model.

Specifically, the raw datasets used in the experiments are collected from the Yahoo! Finance website [6] and Investing website [7]. Different financial factors and stock prices data are collected from the time period between 01/01/2004 to 06/31/2018, which includes 3,415 trading days. In addition, due to the S&P 500 index is more complicated and active than other stock market, more extra financial and economic factors are used as the input signals for the SLFN model. Totally 39 types of financial and economic data related to the S&P 500 index are divided into 6 classes and listed in Table 1. The dataset collected above is normalized and processed according to the normalization theory before fed into the training of the proposed ANN model. The dataset is segmented into two parts: the first 80% is used for training and the remaining 20% is the testing data.

4.2 Experiment Results

In this model, due to 39 types of financial and economic data are adopted as the input data, 39 tapped-delay-line memories are added into the input layer, each tapped-delay-unit memory contains a kind of financial or economic factor. The length of each tapped-delay-units memory is set as 5 to capture the sufficient representative features. Then, the total number of nodes at the input layer is equal 195. All experiments are implemented in MATLAB R2018a in a computer with Intel i7 processor.

Firstly, the randomly selected range of input weights and hidden biases should be small to effectively map features into feature space. Secondly, the sigmoid function is applied as the activation function of the hidden layer nodes and the linear function is the activation function of the output layer nodes. To validate the performance, two most common used performance evaluation criteria, the rooted mean squared error (RMSE) and mean absolute error (MAE), are used to evaluate the performance of the proposed and compared models based on error value or forecasting accuracy.

4.2.1 Model Parameters Analysis

According to previous research in the R-ELM algorithm, we use different combinations of parameter $\boldsymbol{\gamma}$ and the number of hidden nodes N. The number of hidden nodes N are gradually increased by an interval of 50 in [50 2,500] and $\boldsymbol{\gamma} = \left[\mathbf{10^{-1}}, \mathbf{10^{0}}, \mathbf{10^{1}}, \mathbf{10^{2}}, \mathbf{10^{3}}, \mathbf{10^{4}}, \mathbf{10^{5}}\right]$ are recursively grouped to train and test. For the training dataset, the RMSE results of different combinations of $(\boldsymbol{N}, \boldsymbol{\gamma})$ are shown in Fig. 3.

It can be seen from Fig. 3, when $\boldsymbol{\gamma}$ is between 0.1 and 1,000, the adjustment of $\boldsymbol{N}$ cannot improve the testing performance. The difference of $\boldsymbol{\gamma} = \mathbf{10^4}, \mathbf{10^5}, \mathbf{10^6}$ is not obvious, and some parts of these three values of $\boldsymbol{\gamma}$ entangle to each other in Fig. 3. In order to further investigate the effect of $(\boldsymbol{N}, \boldsymbol{\gamma})$ on forecasting performance, the enlarged RMSE curves is shown in Fig. 4. The performances of $\boldsymbol{\gamma} = \mathbf{10^4}, \mathbf{10^5}, \mathbf{10^6}$ are clearer in Fig. 4 than that in Fig. 3. Obviously, with large number of hidden nodes, the model can memorize many noise and disturbance which have no relevant to the target values. Therefore, the performance of N from 250 to 1,250 is become worse. When the value of $\boldsymbol{\gamma}$ is larger, the hidden nodes of the SLFN model are easier to capture redundant information and affect by irrelevant noise in input data during the training processing.

4.2.2 Performance in Training and Testing

Figure 5 shows the performances of the R-ELM-based SLFN model with $\boldsymbol{\gamma} = \mathbf{10^5}$ and $N = 550$ in the stages of training and testing, respectively. In Fig. 5, the red curve is the output of the model and the blue curve is the actual closing price of the S&P 500 index. For the training data, the outputs of ANN model fit the training target very well. This results shows that the proposed model with R-ELM algorithm can learn the dynamics of the changes of S&P index sufficient with proper parameters selection. In addition, two huge increasing trends and one decreasing trend are also captured by the ANN model. In Fig. 5(b), it is noted that the output curve approximates the target curve well. In the first 200 days, the volatilities of stock market are very instable and have many ups and downs. In most positions, the forecasting values are close to targets. From the 201th to the 600th days, the stock market is stable and steadily increasing. The two performance results clearly demonstrate that the developed ANN model with R-ELM algorithm has strong capability of capturing the dynamic non-linear relation between the financial impact factors with the S&P index.

4.2.3 Comparison with the ELM Algorithm

A comparison between the R-ELM and the ELM algorithms is also performed in optimizing the proposed ANN model. For comparison, we select the same value of some

Table 1
The 39 Financial Factors Related to the S&P 500 Index

Classes	Name	Description
Original S&P index	OPEN	Opening prices of the S&P 500 index
	HIGH	High prices of the S&P 500 index
	LOW	Low prices of the S&P 500 index
	CLOSE	Closing prices of the S&P 500 index
	VOLUME	Volume of the S&P 500 index
Technical indicators	EMA	Exponential moving average
	SMA	Simple moving average
	TEMA	Triple exponential moving average
	PPO	The percentage price oscillator
	MACD	Moving average convergence divergence
	RSI	Relative strength index
	OBV	On-balance volume
	STOS	Stochastic oscillator
Exchange rate of USD with some major currencies	USD/CAD	Exchange rate between US dollar and Canadian dollar
	USD/CNY	Exchange rate between US dollar and Chinese Yuan
	USD/EUR	Exchange rate between US dollar and European Dollar
	USD/GBP	Exchange rate between US dollar and British Pound
	USD/JPY	Exchange rate between US dollar and Japanese Yen
Financial and economic indices	WFC	Wells Fargo stock price
	DAAA	The Moody's yield on seasoned corporate bonds
	DBAA	The Moody's yield on month corporate bonds
	DGS	Gold price
	DTB6	Market yield on US Treasury securities at 6 months
	DTB3	Market yield on US Treasury securities at 3 months
	DTB1	Market yield on US Treasury securities at 1 month
	RWTCD	Relative change in the price of the crude oil
The closing price of another important stock markets	GDAXI	DAX index return
	GE	General Electric stock price
	IXIC	NASDAQ composite price
	DJI	Dow Jones Industrial Average price
	HIS	Hang Seng index price
	SEE	Shang Hai stock exchange price
The closing price of seven companies	GOOG	Google Inc. price
	AAPL	Apple Inc. stock price
	JNJ	Johnson and Johnson stock price
	JPM	JPMorgan Chase & Co stock price
	MSFT	Microsoft stock price
	AMZN	Amazon.com Inc. stock price
	XOM	Exxon Mobil stock price

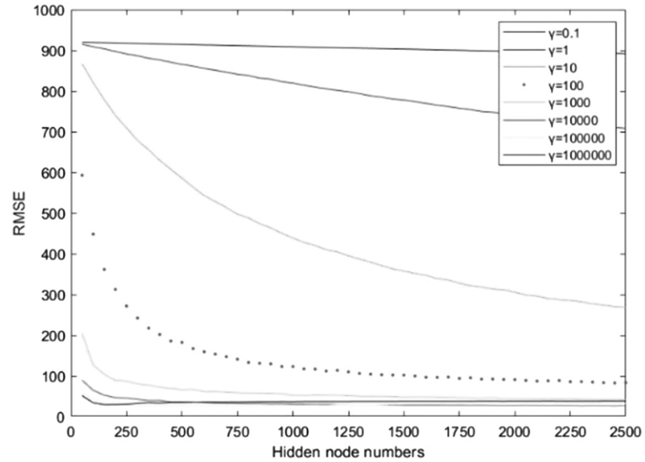

Figure 3. RMSE of different combinations of $(\boldsymbol{N}, \gamma)$.

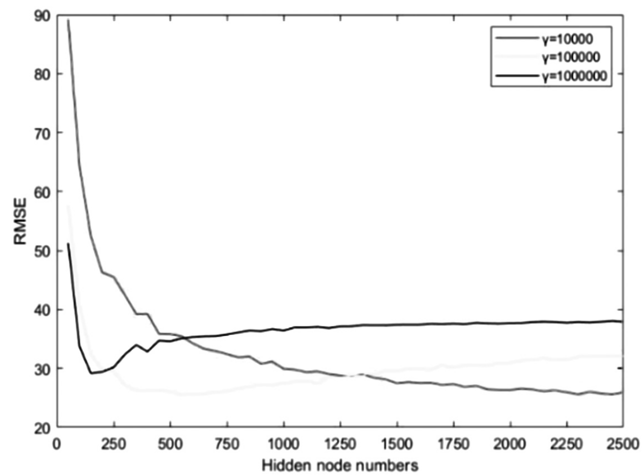

Figure 4. RMSEs of different hidden neurons with $\boldsymbol{\gamma = 10^4, 10^5, 10^6}$.

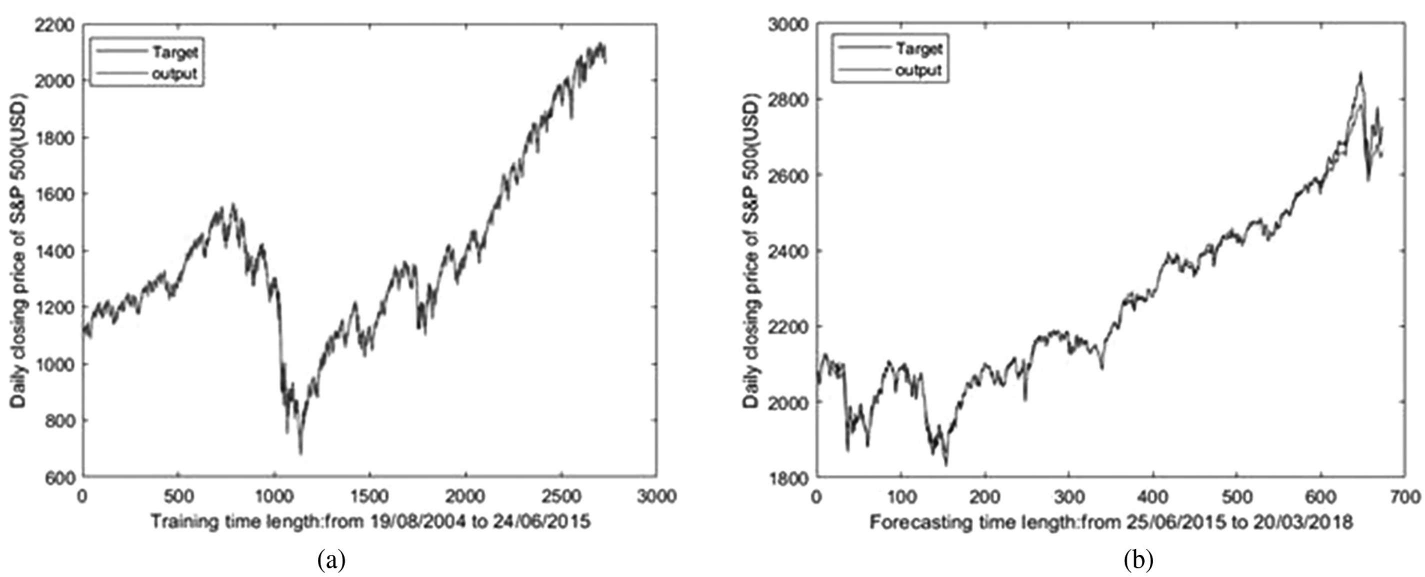

Figure 5. (a) Training performances of the SLFN model trained with the R-ELM algorithm and (b) Testing performances of the SLFN model trained with the R-ELM algorithm.

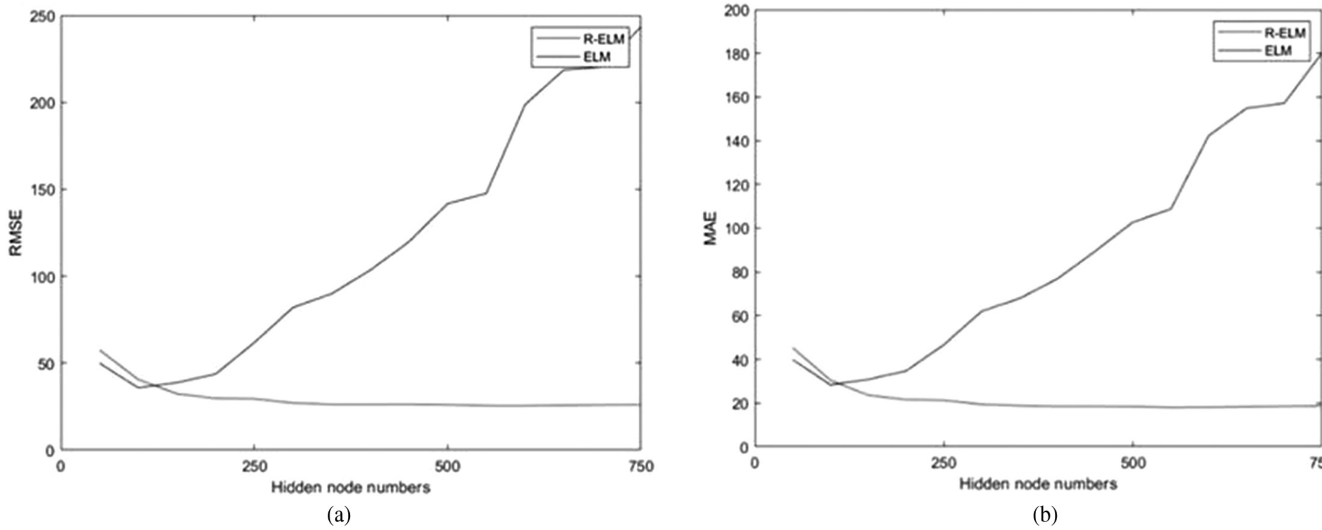

Figure 6. (a) RMSEs of the SFLN model trained with the ELM and the R-ELM and (b) MAEs of the SLFN model trained with the ELM and the R-ELM.

parameters in the ELM training. The number of hidden nodes is gradually increased by an interval of 50 in [50, 2,500].

Figure 6 shows the RMSE and MAEs of the ELM and the R-ELM algorithms with different number of hidden nodes. It is obviously that with the increasing of hidden neuron numbers, the better performance can be obtained with R-ELM algorithm. In addition, in the processing of the R-ELM, with the regularization term, the RMSE performance is remarkable and stably. With the regularization term, the disturbances and noises can be removed, so the model trained with R-ELM not only has better performance but also has high level of robustness, as shown in the figures. The results of above experiments verified that the R-ELM algorithm is more suitable for training the proposed ANN model for stock price forecasting.

4.2.4 Model Comparisons with Classical Models

The proposed R-ELM-based SLFN model is also compared with four classical models, backpropagation neural network (BP-NN), linear regression (LR), ARIMA and GARCH. The forecasting target of all these models is the 1-day ahead S&P 500 stock closing price. For BP-NN, R-ELM and LR models, the 39 financial factors shown in Table 1 are used as the inputs and the delay units of these inputs are selected as 15. The inputs for ARIMA and GARCH models are the delays of closing price. The length of delay is selected from statistical tests in literature [6]. The RMSE and MAE are shown in Table 2.

As shown in Table 2, the RMSE and MAE of SLFN model trained with R-ELM exhibits remarkable performances in both RMSE and MAE comparing to the other four models. The testing RMSE and MAE of R-ELM-based SLFN model are 25.53 and 18.01, respectively, which are much smaller than other models. With good hyper-parameter tuning, the SLFN model trained with R-ELM is able to achieve accurate results.

Table 2
Comparison with a Few Existing Models

Models	RMSE	MAE
R-ELM	25.53	18.01
BP-NN	38.16	30.21
LR	41.16	32.21
ARIMA	42.12	33.12
GARCH	41.02	32.12

According to all the experiments and comparisons shown above, the modelling and prediction performance of the proposed ANN model are very good. This is because the tapped delay units are added to the input layer and the R-ELM algorithm is appropriately chosen. Therefore, the developed model has the capability of forecasting stock price in practical applications with satisfying accuracy.

5. Conclusion

In this paper, a modified SLFN model with R-ELM algorithm has been developed for modelling the daily closing price of the S&P 500 index. The main contributions of this work are as follows: (i) the tapped delay units are applied at the input layer to help that the model can effectively capture the dynamics in financial data; (ii) the R-ELM algorithm overcomes the over-fitting problem which found in ELM models, especially in case of fluctuations in stock sector. The experimental results have proven that the proposed model achieved the lowest error value for all compared evaluation criteria (RMSE and MAE) followed by the other four compared models. In order to overcome the influence of uncertain structural state information on the stock price, the work on developing prediction models based on unsupervised deep learning technique using the real-time state date is to be investigated in the future.

References

[1] T.Z. Tan, C. Quek, and G.S. Ng, Brain-inspired genetic complementary learning for stock market prediction, *The 2005 IEEE Congress on Evolutionary Computation, 3*, 2005, 2653–2660.

[2] G.E. Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung, *Time series analysis: Forecasting and control* (John Wiley and Sons, Hoboken, New Jersey, 2015).

[3] W. Hong *et al.*, Stock Price movements Prediction with Textual Information, *Mechatronics Systems and Control, 46*, 2018, 3039–3047.

[4] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics, 31*, 1986, 307–327.

[5] D.K. Mohanty, A.K. Parida, and S.S. Khuntia, Financial market prediction under deep learning framework using auto encoder and kernel extreme learning machine, *Applied Soft Computing*, *99*, 2021, 106898.

[6] X. Zhong and D. Enke, A comprehensive cluster and classification mining procedure for daily stock market return forecasting, *Neurocomputing, 267*, 2017, 152–168.

[7] X. Li *et al.*, Empirical analysis: Stock market prediction via extreme learning machine, *Neural Computing and Applications, 27*, 2016, 67–78.

[8] J. Sweafford Jr. and F. Fahimi, A neural network model-based control method for a class of discrete-time nonlinear systems, *Mechatronic Systems and Control, 49*, 2021, 0134–0142.

[9] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, Extreme learning machine: Theory and applications, *Neurocomputing, 70*, 2006, 489–501.

[10] Y. Hu, H. Wang, A. Yazdani, *et al.*, Adaptive full order sliding mode control for electronic throttle valve system with fixed time convergence using extreme learning machine, *Neural Computing and Applications*, *34*, 2021, 5241–5253.

[11] J. Zhang, H. Wang, M. Ma, M. Yu, A. Yazdani, and L. Chen, Active front steering-based electronic stability control for steer-by-wire vehicles via terminal sliding mode and extreme learning machine, *IEEE Transactions on Vehicular Technology, 69*, 2020, 14713–14726.

[12] M. Ye and H. Wang, Robust adaptive integral terminal sliding mode control for steer-by-wire systems based on extreme learning machine, *Computers and Electrical Engineering, 86*, 2020, 106756.

[13] Y. Hu, H. Wang, Z. Cao, *et al.*, Extreme-learning-machine-based FNTSM control strategy for electronic throttle, *Neural Computing and Applications, 32*, 2020, 14507–14518.

[14] V. Vapnik, *The nature of statistical learning theory* (Springer Science and Business Media, New York, 2013).

[15] W. Deng, Q. Zheng, and L. Chen, Regularized extreme learning machine, *Proc. IEEE Symposium on CIDM 2009*, Nashville, TN, USA, 2009, 389–395.

[16] S. Niazi *et al.*, Neuro-predictive controller for stabilization of Gimbal Mechanism with Cross-coupling, *Mechatronics Systems and Control*, *49*, 2021, 0198–0205.

[17] J. Tavoosi, A novel recurrent type-2 fuzzy neural network for stepper motor control, *Mechatronics Systems and Control*, *49*, 2021, 0097–0105.

[18] M.C. Khani and S. Shabanlou, Assessment of discharge coefficient in trapezoidal and rectangular canals through regularized extreme learning machine, *Measurement*, *180*, 2021, 109493.

[19] Z.Z. Man, Z. Cao *et al.*, A new intelligent pattern classifier based on structured sparse representation, *Computers and Electrical Engineering*, *84*, 2020, 106641.

[20] B. Liu, G. Chen, H. Lin, *et al.*, Prediction of IGBT junction temperature using improved cuckoo search-based extreme learning machine, *Microelectronics Reliability*, *124*, 2021, 114267.

Biographies

Yuyang Zhu received his B.S degree from Zhejiang Normal University, China, and the M.E degree from East China Normal University, China. Since 2001, he works in Lishui University, China. His current research focuses on data mining and computational intelligence systems for Bioinformatics, Information Retrieval and Engineering Applications.

Linxian Zhi received her B.S degree from Jiangxi Normal University, China, in 2002, the M.E degree from ZheJiang University, China, in 2007, and the Ph.D. degree from Swinburne University of Technology, Australia in 2019. Since 2002, she has been the associate professor in Lishui University, China. Her research interests are in neural networks, data mining and computational intelligence systems and machine learning.

Weiduo Zhang received his B.S. degree from Zhejiang Gongshang University, China, in 2011, and the M.E. degree from Swinburne University of Technology, Australia in 2019. Since 2019, he has been the Data scientist in the Zhejiang Rural Credit Union. His current research focuses on neural networks and data mining.