Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 12, December 2019, pp. 3271-3278
DOI: 10.23940/ijpe.19.12.p19.32713278

Bayesian Regularization Neural Network Model for Stock Time Series Prediction

Yue Hou*, Bin Xie, and Heng Liu

Lanzhou Jiaotong University, Lanzhou, 730070, China

Abstract

With strong nonlinear characterization ability, a BP neural network can effectively describe the characteristics of nonlinear time series. However, there are still some limitations, such as the ease of falling into a local optimum. Aiming at this problem, the Bayesian regularization optimization algorithm was used to improve the BP neural network. Under the premise of minimizing the objective function, the algorithm adjusts the weight update function through the conditional probability density and the prior probability of the historical data. Thus, the generalization capability of BP neural network will be enhanced. After an empirical study on stock time series prediction, we found that the improved network could prominently increase the prediction ability, while the ability of volatility prediction was better than that of other traditional algorithms.

Keywords: bayesian regulation; neural network; time series prediction

(Submitted on October 13, 2019; Revised on November 15, 2019; Accepted on November 22, 2019)

1. Introduction

Stock investments have considerably high risks. Therefore, it is of practical significance to predict the downside risks of stocks for a reasonable portfolio strategy. Generally, stock risks are evaluated by value at risk (VaR) [1]. Traditional methods uses historical data to construct time series models to calculate VaR values, such as historical simulations [2], quantile regression [3], the Monte Carlo method [4], and autoregressive analysis [5]. However, the volatility of stock prices is inconsistent with efficient market theory, indicating that stock price fluctuations are not completely random, and stock price volatility has certain regularity and predictability [6]. Therefore, it is possible to predict stock prices by establishing a stock time series forecasting model to effectively predict stock downside risks.

In 1990, Wersino and Varfis first adopted the artificial neural network to model the financial time series problem [7]. Subsequently, many scholars conducted in-depth research on it. [8] applied a multilayer BP neural network to forecast stock price and concluded that the BP neural network has stronger prediction ability than traditional statistical and econometric analysis methods. However, the article did not mention the problem that BP neural networks easily fall into local optima, which can affect their generalization ability. Later, many experts and scholars used improved neural network algorithms to build stock time series prediction models [9-10]. The experimental results showed that the neural network model performs well in short-term stock forecasting.

Taking stock time series prediction as an example, the Bayesian regularization algorithm (BR) is used to improve the constraint function of the BP neural network in this paper. The experimental results illustrate that the Bayesian regularization neural network (BRNN) prediction model has higher accuracy and feasibility compared with the traditional BP neural network and its improved model.

* Corresponding author.
E-mail address: 2016023148@qq.com

2. Bayesian Regularization Neural Network Theory

2.1. Regularization Theory

The target of supervised learning is to minimize the loss function.

$$w^* = \arg\min_w \sum_i E(y_i, f(x_i, w)) + \lambda\Omega(w) \tag{1}$$

Where $E(y_i, f(x_i, w))$ is the loss of predicted value $f(x_i, w)$ and expected value y_i; in this work, we take the sum of the squared error (SSE) as the loss function, that is, $E_{SSE} = \sum_{i=1}^{n} w_i (y_i - \hat{y}_i)^2$. In traditional prediction models, λ is usually set to 0, which can ensure a minimized training loss but will result in an overfitting problem. Adding a regularization term $\Omega(w)$ on the original loss function will effectively punish the weight vector w and reduce the possibility of overfitting, which enhances the generalization ability of the model.

2.2. Bayes Theorem

The Bayes theorem [11-12] is an important branch of probability theory. The basic principle of Bayes is to use historical data to obtain the prior probability of the parameter to be estimated in the case of incomplete information, rectify the probability by Equation (2), and finally calculate the optimum value of the parameters based on the expected value and the rectified probability.

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_{i=1}^{n} P(B_i)P(A|B_i)} = \frac{P(A_i)P(B|A_i)}{P(B)} \tag{2}$$

In Equation (2), $P(A_i)$ is the prior probability and $P(A_i|B)$ is the rectified probability under given information, also called the posterior probability. $P(B|A_i)$ is the likelihood function under given information A_i, and $P(B)$ is the proof.

On the perspective of Bayes, the regularization item is equivalent to introducing the prior probability distribution of the weight. Given the sample data S, the process of estimating the weight vector w by maximizing the posterior probability is shown in Equation (3).

$$w^* = \arg\max_w p(w|S) = \arg\max_w \frac{p(S|w) \times p(w)}{p(S)} = \arg\max_w p(S|w)\, p(w) \tag{3}$$

In the above formula, $p(S|w) = \prod_{k=1}^{n} p(S_i|w)$ is the probability of the observation data S given the weight vector w.

Taking the logarithm of the posterior likelihood to simplify our computation, we get

$$w_* = \arg\max_w \sum_{k=0}^{n} \ln p(S|w) + \ln p(w) = \arg\min_w - \sum_{k=0}^{n} \ln p(S|w) - \ln p(w) \tag{4}$$

If the the prior probability satisfies a Laplace distribution, that is,

$$p(w_i) = N(w_i \mid \mu, b) = \frac{1}{2b} e^{-\frac{|w_i - \mu|}{b}} \tag{5}$$

Equation (4) can be rewritten as follows:

$$\begin{aligned} w_* &= \arg\min_w -\ln\sum_{k=0}^{n} p(S_i|w) - \sum_{i=0}^{m}\ln p(w_i) \\ &= \arg\min_w -\ln\sum_{k=0}^{n} p(S_i|w) - \sum_{i=0}^{m}\frac{1}{b}|w_i\text{-}\mu| \quad , \quad \mu=0,\ b=\frac{1}{\lambda} \\ &= \arg\min_w -\ln\sum_{k=0}^{n} p(S_i|w) + \lambda\sum_{i=0}^{m}|w_i| \end{aligned} \tag{6}$$

The likelihood function of the above formula corresponds to the loss function of the objective function, and the prior probability part is the regularization term. Through this method, we optimize the prediction model and improve the strategy to update the weights, thereby alleviating the risk of overfitting to some extent as well as boosting the generalization ability of the prediction model.

3. Time Series Prediction Modeling

3.1. Parameters Setting

The following is the process of building our prediction model.

We experimented on the stock data of nearly 10 years of Bao Gang (stock code: 600019, data from Tushare financial data interface) with the closing price (Close), opening price (Open), previous trading day's closing price (PreClose), highs (High), lows (Low), average price of the past 10 days (Avg10), average price of the past 10 days (Avg20), and historical volatility (HV). The dataset is depicted as Table 1.

Table 1. Description of Bao Gang

Date	Close	Open	PreClose	High	Low	Avg10	Avg20	HV	Label
20060515	4.55	4.34	4.33	4.55	4.34	4.290	4.3225	0.0496	4.49
20060516	4.49	4.55	4.55	4.69	4.43	4.313	4.3375	-0.0133	4.63
20060518	4.63	4.49	4.49	4.67	4.45	4.341	4.3480	0.0307	4.85
20060519	4.85	4.63	4.63	4.98	4.61	4.377	4.3665	0.0464	4.91
20060522	4.91	4.85	4.85	5.05	4.85	4.439	4.3965	0.0123	4.8
...	...	...	...	...	...	...	...	...	...
20160523	5.22	5.21	5.21	5.24	5.17	5.208	5.4180	0.0019	5.16
20160524	5.16	5.2	5.22	5.2	5.14	5.213	5.3940	-0.0116	5.17
20160525	5.17	5.19	5.16	5.21	5.15	5.208	5.3630	0.0019	5.17
20160526	5.17	5.15	5.17	5.19	5.11	5.203	5.3330	0.0000	5.21
20160527	5.21	5.15	5.17	5.29	5.14	5.195	5.3070	0.0077	5.18

Then, we adopted the mapminmax function (Equation (7)) to normalize the date.

$$y_r = (y_{\max} - y_{\min}) \times (x - x_{\min}) / (x_{\max} - x_{\min}) + y_{\min} \tag{7}$$

Where x is the data to be normalized and $x_{\max}$ and $x_{\min}$ are the maximum value and minimum value, respectively. To prevent sample data from entering the saturation region of the activation function and simultaneously reduce the data noise, the normalized value range was set to [02, 0.8], which means $y_{\max} = 0.8$ and $y_{\min} = 0.2$. After normalization, we obtained the input date.

We grouped the input data by adopting the sliding window method [13], with *n* being the input size of the sliding window. Then, we split the input data into a training set and a test set by the portion 8:2, which means 80% of the original dataset is the training set, and the rest is the test set.

We chose the gradient descent algorithm (GD) [14-15], Levenberg Marquardt algorithm (LM) [16], and Bayesian regulation algorithm (BR) [17-18] to train our neural network model. Figure 1 shows the prediction error of two algorithms with respect to the different datasets and network structures, where the horizontal axis represents the datasets and the

vertical axis is the RMSE. The four different shaped curves represent the different net structures (e.g., 10-4-1 means 10 input nodes, 4 hidden nodes, and 1 output node).

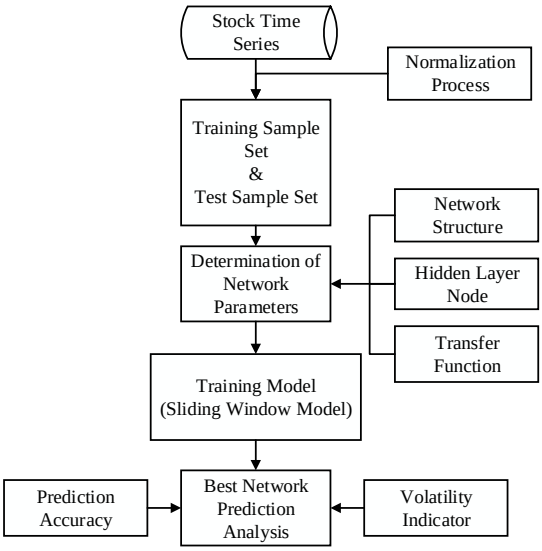
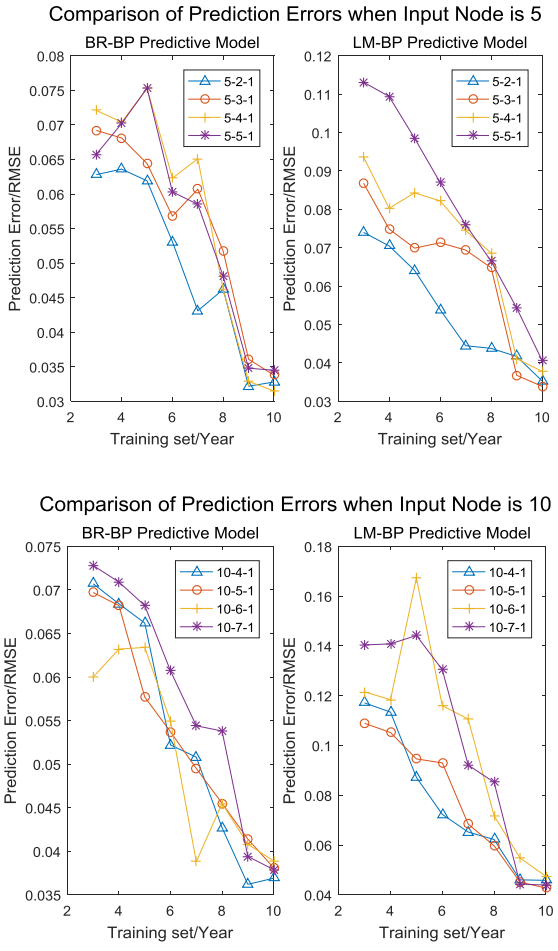


Figure 1. Process of building our prediction model

Figure 2 shows that the prediction error under the structure 10-4-1 is significantly lower than that of other structures. Thus, we can assume that the best structure of the neural network is 10-4-1.



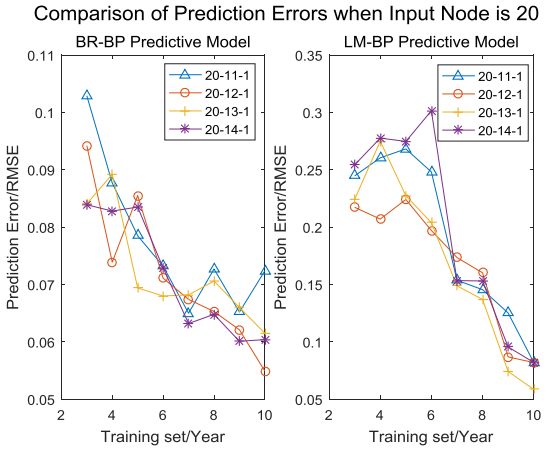


Figure 2. Error curve under different training parameters

3.2. Analysis of the Results

3.2.1. Prediction Model with a Single Feature

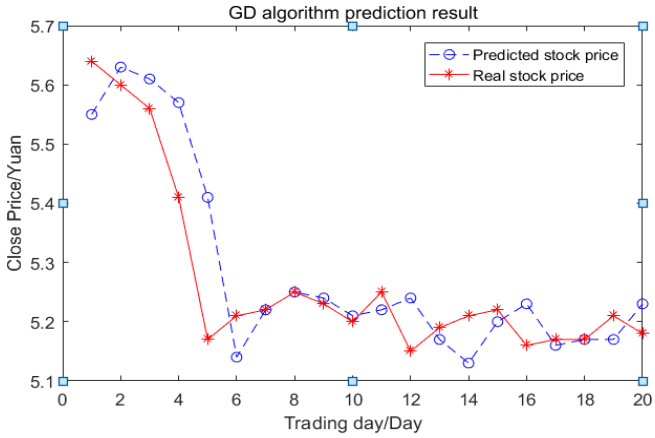
Firstly, we experimented on our stock data with a single feature, the closing price, to test the performance of the BR-BP model. We adopted four different evaluation methods, which are listed below (a detailed description is given in Table 2). $y(n)$ is the actual closing price of the n_{th} day, $\hat{y}(n)$ is the predicted value of the n_{th} day, and N is the total number of predicted days. *CDR* and *C* are used to measure the fluctuations of stock price; the higher they are, the closer the volatility of the predicted value is to the real volatility. *RMSE* and *MAPE* are used to measure the precision; the lower they are, the closer the predicted value is to the real value.

Table 2. Performance metric description

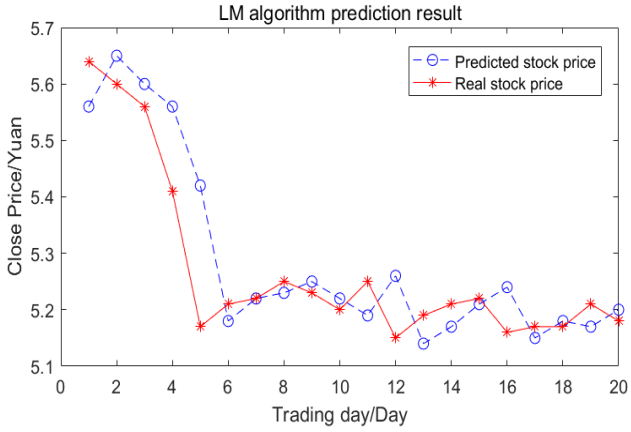
Abbreviation	Name	Formula
RMSE	Root mean squared error	$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y(n)-\hat{y}(n))^2}$
MAPE	Mean absolute percentage error	$MAPE = \sum_{n=1}^{N}\lvert y(n)-\hat{y}(n)\rvert / y(n) \times 100\%$
CDR	Correct direction rate	$\begin{cases} CDR = \frac{1}{N}\sum_{n=1}^{N} CDr_n \times 100\% \\ CDr_n = \begin{cases} 1 & (y(n+1)-y(n)) \times (y(n+1)-\hat{y}(n)) > 0 \\ 0 & \text{else} \end{cases} \end{cases}$
R	Correlation coefficient	$R = \frac{\sum_{n=1}^{N}(\hat{y}(n)-\bar{\hat{y}})(y(n)-\bar{y})}{\sqrt{\sum_{n=1}^{N}(\hat{y}(n)-\bar{\hat{y}})^2 (y(n)-\bar{y})^2}}$

With the above settings, we then trained our model using the GD algorithm, LM algorithm, and BR algorithm to obtain the closing price in twenty trading days. The result is shown in Figure 3.

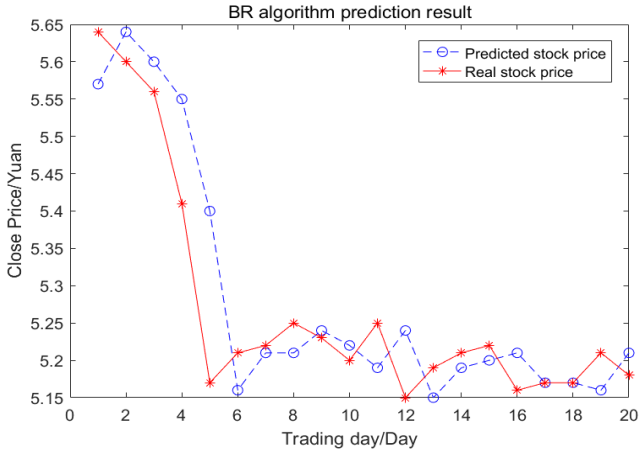
As shown in Figure 3 and Table 3, in terms of RMSE and MAPE, the BR algorithm is superior to the traditional GD algorithm and the LM algorithm, and it has better generalization ability and prediction precision. The CDR and R of the BR algorithm are higher than those of the other two algorithms, indicating that the BR algorithm also performs well in forecasting the fluctuations.



(a) GD algorithm



(b) LM algorithm



(c) BR algorithm

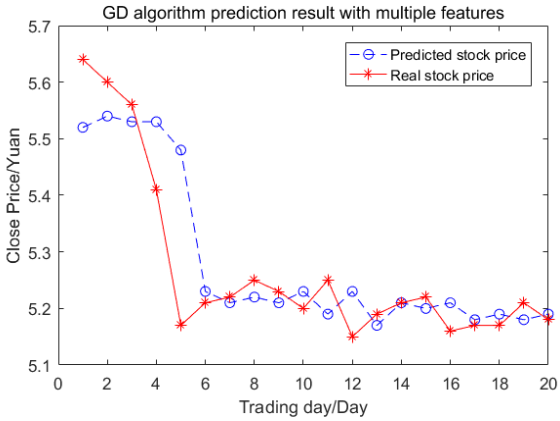
Figure 3. Prediction results of different models under single feature

Table 3. Comparison of different training algorithms with a single feature

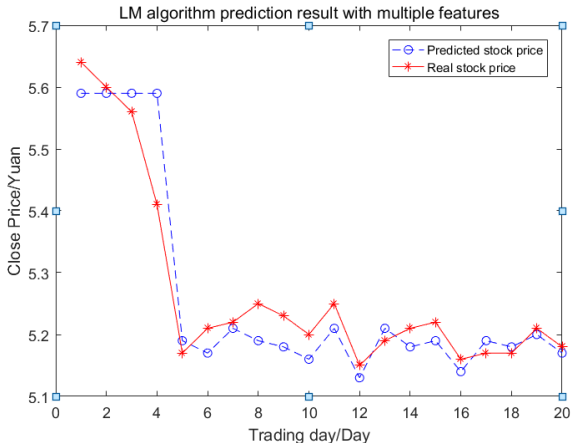
Training Algorithm	RMSE	MAPE	CDR	C
GD	0.0773	1.04%	35%	0.8918
LM	0.0762	1.02%	40%	0.8930
BR	0.0732	0.95%	40%	0.8986

3.2.2. Prediction Model with Multiple Features

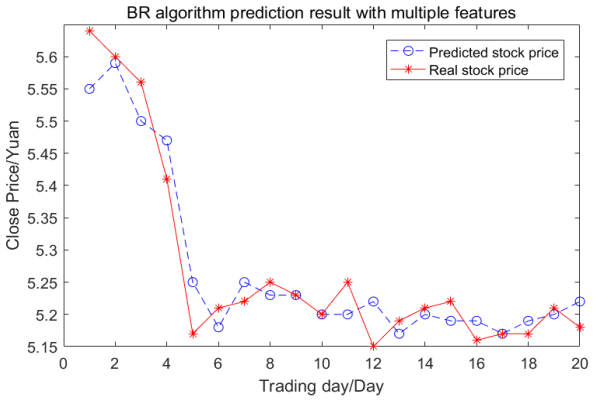
With multiple features to train the prediction model, it can learn the non-linear characteristic of the stock data more effectively. Thus, we tried to train the model with seven new features besides the closing price to observe the preformance. The comparison with the model trained with a single feature is shown in Figure 4.



(a) GD algorithm



(b) LM algorithm



(c) BR algorithm

Figure 4. Prediction results of different models with multiple features

The comparison is shown in Table 4 for different multiple feature models. Note that the performance metric of most multiple feature models is prominently higher than that of single feature models, especially the BR-BP model. This implies that the prediction model under multiple features generates stronger non-linear characteristics of the stock data.

Table 4. Comparison of different training algorithms with multiple features

Training Algorithm	RMSE	MAPE	CDR	C
GD-BP	0.0959	1.28%	65%	0.7877
LM-BP	0.0414	0.62%	70%	0.9656
BR-BP	0.0423	0.62%	75%	0.9616

4. Conclusions

In this work, we adopt Bayesian regularization to constrain the loss function of BP neural networks. By optmizing the

solution space of the weight vector, the BR algorithm can control the network structure and decrease the risk of overfitting. During the experiments, we used the stock time series as the objective of study. After repeated trials, we found the best parameters of the model, and then we trained three different models with single and multiple features and obtained the predicted closing price in future days. The experimental results showed that the neural network can be used to model time series prediction. Compared with the traditional BP neural network, the BR-BP network performed better in predicting both trends and volatility. Furthermore, the prediction results under multiple features were superior in each evaluation metric, which indicated that the closing price was affected by many factors. Our work in this paper was mainly focused on the closing price of stock data, and we will consider taking more variables as the output of the prediction model in future work.

Acknowledgements

This work is supported by the Gansu Provincial Communications Department Project (No. 2016B-027).

References

1. H. Darryll and B. Hirtle, "Bank Capital Requirements for Market Risk: The Internal Models Approach," *Social Science Electronic Publishing*, Vol. 3, No. 4, pp. 1-12, December 1997
2. Y. T. Wan, Z. G. Zhang, L. T. Zhao, and D. M. Ai, "An Improved Historical Simulation Method to Estimate the Amount of Refined Oil Retail Value at Risk VaR," *Advanced Materials Research*, Vol. 734, pp. 1711-1718, 2013
3. J. Schaumburg, "Predicting Extreme Value at Risk: Nonparametric Quantile Regression with Refinements from Extreme Value Theory," *Computational Statistics & Data Analysis*, Vol. 56, No. 12, pp. 4081-4096, 2012
4. C. S. Chen, M. A. Golberg, and Y. C. Hon, "The Method of Fundamental Solutions and Quasi - Monte - Carlo Method for Diffusion Equations," *International Journal for Numerical Methods in Engineering*, Vol. 43, No. 8, pp. 1421-1435, 1998
5. A. Suharsono, A. Aziza, and W. Pramesti, "Comparison of Vector Autoregressive (VAR) and Vector Error Correction Models (VECM) for Index of ASEAN Stock Price," in *Proceedings of AIP Conference*, Vol. 1913, No. 1, pp. 020032, 2017
6. Yen, Gili and C. F. Lee, "Efficient Market Hypothesis (EMH): Past, Present and Future," *Review of Pacific Basin Financial Markets and Policies*, Vol. 11, No. 12, pp. 305-329, 2008
7. A. Varfis and C. Versino, "Univariate Economic Time Series Forecasting by Connectionist Methods," *Proc. of INNC*, Vol. 56, pp. 342-345, 1990
8. S. O. Olatunji and M. S. Al-Ahmadi, "Saudi Arabia Stock Prices Forecasting using Artificial Neural Networks," *Applications of Digital Information and Web Technologies* (ICADIWT), pp. 81-86, Wisconsin, USA, August 2011
9. K. J. Kim and I. Han, "Genetic Algorithms Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index," *Expert Systems with Applications*, Vol. 19, No. 12, pp. 125-132, August 2000
10. W. M. Ma, Y. Y Wang, and N. F. Dong, "Study on Stock Price Prediction based on BP Neural Network," *Emergency Management and Management Sciences* (*ICEMMS*), pp. 57-60, Beijing, China, August 2010
11. Q. Sun, W. G. Che, and H. L. Wang, "Bayesian Regularization BP Neural Network Model for the Stock Price Prediction," *Advances in Intelligent Systems and Computing*, pp. 521-531, November 2013
12. A. V. D. Vaart, "Bayesian Regularization," Hindustan Book Agency, New Delhi, 2010
13. J. S. Chou and T. K. Nguyen, "Forward Forecast of Stock Price using Sliding-Window Metaheuristic-Optimized Machine-Learning Regression," *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 7, pp. 3132-3142, July 2018
14. Z. Q. Guo, H. Q. Wang, J. Yang, and D. J. Miller, "A Stock Market Forecasting Model Combining Two-Directional Two-Dimensional Principal Component Analysis and Radial Basis Function Neural Network," *PLOS ONE*, Vol. 10, No. 4, pp. e0122385, April 2015
15. L. Mao, D. Y. Qi, and X. X. Li, "Improved GA Combined with GDBP Algorithm for Forecasting Releasing Behaviors of Drug Carrier," *Progress in Informatics and Computing* (*PIC*), pp. 35-39, Shanghai, China, December 2010
16. Q. Xue, F. Yun, C. W. Zheng, and Y. H. Liu, "Improved LMBP Algorithm in the Analysis and Application of Simulation Data," *Computer Application and System Modeling* (*ICCASM*), pp. V6-545-V6-547, Taiyuan, China, October 2010
17. Ticknor and L. Jonathan, "A Bayesian Regularized Artificial Neural Network for Stock Market Forecasting," *Expert Systems with Applications*, Vol. 40, No. 14, pp. 5501-5506, October 2013
18. Q. Sun, W. G. Che, and H. L. Wang, "Bayesian Regularization BP Neural Network Model for the Stock Price Prediction," *Foundations and Applications of Intelligent Systems*, pp. 521-531, 2014

Yue Hou is an associate professor and received her master's degree from Lanzhou Jiaotong University. Her research areas include neural networks and intelligent transportation.

Bin Xie is a master's student in the School of Electronic and Information Engineering at Lanzhou Jiaotong University. His research areas are big data and recommendation systems.

Heng Liu is a master's student in the School of Electronic and Information Engineering at Lanzhou Jiaotong University. His research areas are time series analysis and neural networks.