Chaos, Solitons and Fractals 85 (2016) 1–7

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

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

Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market

Mingyue Qiu*, Yu Song, Fumio Akagi

Department of Systems Management, Fukuoka Institute of Technology, 3-30-1 Wajiro-higashi, Higashi-ku, Fukuoka 811-0295 Japan

ARTICLE INFO

Article history:
Received 7 December 2015
Accepted 6 January 2016

Keywords:
Finance
Artificial intelligence
Artificial neural network
Genetic algorithm
Simulated annealing
Japanese stock market

ABSTRACT

Accurate prediction of stock market returns is a very challenging task because of the highly nonlinear nature of the financial time series. In this study, we apply an artificial neural network (ANN) that can map any nonlinear function without a prior assumption to predict the return of the Japanese Nikkei 225 index. (1) To improve the effectiveness of prediction algorithms, we propose a new set of input variables for ANN models. (2) To verify the prediction ability of the selected input variables, we predict returns for the Nikkei 225 index using the classical back propagation (BP) learning algorithm. (3) Global search techniques, i.e., a genetic algorithm (GA) and simulated annealing (SA), are employed to improve the prediction accuracy of the ANN and overcome the local convergence problem of the BP algorithm. It is observed through empirical experiments that the selected input variables were effective to predict stock market returns. A hybrid approach based on GA and SA improve prediction accuracy significantly and outperform the traditional BP training algorithm.

1. Introduction

To revive the Japanese economy, the Japanese government has recently developed many significant economic strategies, and each strategy is closely related to the Japanese stock market. As the most widely used market index for the Tokyo Stock Exchange, the Nikkei 225 index, also known as the Nikkei average or simply Nikkei, is a benchmark that is used to evaluate the Japanese economy. Forecasting the stock return of the Nikkei 225 index is an important financial subject that has attracted significant attention in major financial markets around the world. The purpose of this paper is to apply an artificial neural network (ANN) to forecast the return of the Nikkei 225 index.

It has been widely accepted by many studies that nonlinearity exists in financial markets and that an ANN can be used effectively to uncover this relationship [1]. McCulloch and Pitts [2] created a computational model for neural networks based on mathematics and algorithms, and the application of ANNs to financial and investment decisions has been examined by researchers for many years. Compared to regression or the passive buy-and-hold strategy, Motiwalla and Wahab [3] found that ANN models are more successful in predicting returns. Enke and Thawornwong [1] used neural network models for level estimation and classification. They showed that the trading strategies guided by a neural network classification model can generate higher profits than any other model. Hodnett and Hsieh [4] utilized two ANN learning rules to forecast the cross-section of global equity returns. Their findings support the use of ANNs for financial forecasting. Application of ANNs has become the most popular machine learning method, and it has been proven that such an approach can outperform conventional methods [5-13].

* Corresponding author. Tel.: +81 80 9062 7301; fax: +81 92 606 0756.
E-mail addresses: qmy1175116@126.com, p15x1001@bene.fit.ac.jp (M. Qiu), song@fit.ac.jp (Y. Song), akagi@fit.ac.jp (F. Akagi).

http://dx.doi.org/10.1016/j.chaos.2016.01.004
0960-0779/© 2016 Elsevier Ltd. All rights reserved.

In light of previous studies, it has been hypothesized that various technical indicators may be used as input variables in the construction of prediction models to forecast the return of a stock price index [14]. In most applications, input variables that have been proven effective by previous studies were used to predict stock market returns. Some effective indicators of stock price include lagged returns, interest rate value, foreign exchange rate, consumer price index, industrial production index, and deposit rate [1,15]. In this study, we examined the indicators that were proven valid by prior studies and attempted to determine input variables that have not been previously used to predict stock market returns by assessing the ability of those indicators to predict a stock market index. We collected 71 input variables that cover financial and economic information of the Japanese stock market, most of which have not been examined in previous studies.

Although an ANN can be a very useful tool in the prediction of stock market returns, several studies have shown that ANNs have some limitations because stock market data contain a tremendous amount of noise, non-stationary characteristics, and complex dimensionality [16]. Therefore, we must perform data preprocessing prior to utilizing an ANN to predict stock market returns. This study attempted to implement fuzzy surfaces in the selection of optimal input variables. As a result, 18 valid explanatory variables were selected from the 71 input variables for experimentation.

We first applied a back propagation (BP) algorithm to train the neural network in a large number of experiments. The BP algorithm is a widely applied classical learning algorithm for neural networks. Wong, Bodnovich [17] found that many of the studies that have used ANNs relied on gradient techniques for network training, typically some variation of the BP algorithm. Although, researchers have commonly trained ANNs using the gradient technique of the BP algorithm, limitations of gradient search techniques emerge when ANNs are applied to complex nonlinear optimization problems [18]. The BP algorithm has two significant drawbacks; i.e., slowness in convergence and an inability to escape local optima [19]. In view of these limitations, global search techniques, such as genetic algorithms (GA) and simulated annealing (SA), have been proposed to overcome the local convergence problem for nonlinear optimization problems. This study has attempted to determine the optimal set of initial weights and biases to enhance the accuracy of an ANN using GA or SA. The experimental results show that a hybrid approach improves the prediction accuracy of the return of the Nikkei 225 index and outperforms the BP training algorithm. In addition, the effect on prediction of the combined 18 input variables is effective and can therefore be a good alternative for predicting stock market returns.

The remainder of this paper is organized as follows. Section 2 provides an outline of the prediction procedure. Section 3 describes variable selection. Then, we describe experiments that used the BP algorithm in Section 4, and we discuss improving parameter training using a GA or SA in Section 5. Finally, Section 6 provides a discussion of the experimental results and conclusions. Data descriptions are provided in the Appendix.

2. Prediction procedure

2.1. Data description

The Nikkei 225 index is the most widely used market index for the Tokyo Stock Exchange. It includes 225 equally weighted stocks and has been calculated daily since 1950. To predict the returns of the Nikkei 225 using an ANN, we collected 71 variables that include financial indicators and macroeconomic data. The entire data set covers the period from November 1993 to July 2013, providing a total of 237 months of observations. The data set was divided into two periods. The first period covers November 1993 to December 2007 (170 months), and the second period covers January 2008 to July 2013 (67 months). The first period, i.e., the in-sample data, was divided into training (70% of the period) and prediction (30% of the period) sets. The training data was used to determine model specifications and parameters, and the prediction set was reserved for evaluation and comparison of performance among the prediction models. The second period, i.e., the out-of-sample data, was reserved for testing the performance of the prediction models because this data was not utilized to develop the models.

2.2. Model description

Funahashi [20], Hornik, Stinchcombe [21] have shown that neural networks with sufficient complexity could approximate any unknown function to any degree of desired accuracy with only one hidden layer. Therefore, the ANN model in this study consists of an input layer, a hidden layer and an output layer, and each of which is connected to the other. The architecture of the ANN is shown in Fig. 1. The input layer corresponds to the input variables, with one node for each input variable. The hidden layer is used for capturing the nonlinear relationships among variables. Note that an appropriate number of neurons in the hidden layer needs to be determined by repeated training. The output layer consists of only one neuron that represents the predicted value of the output variable.

2.3. Prediction procedures

The architecture of our experimental process is shown in Fig. 2. First, we applied fuzzy surfaces to the selection of effective input variables prior to modeling. Then, we performed BP algorithm experiments 900 times to determine the most appropriate parameter combination for the ANN. We selected the best BP model for predicting the stock returns. Using the BP algorithm, we can obtain the optimized weights and biases of the network by repeated training. We also applied a GA and SA to improve the ANN parameters. We then trained the network using the BP algorithm with the improved weights and biases. Finally, we

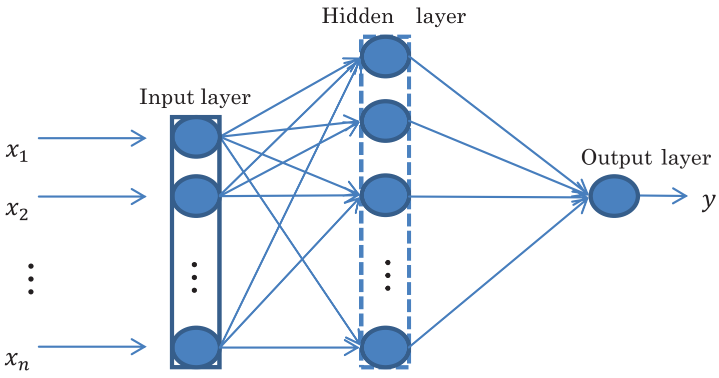

Fig. 1. Architecture of the three-layered ANN.

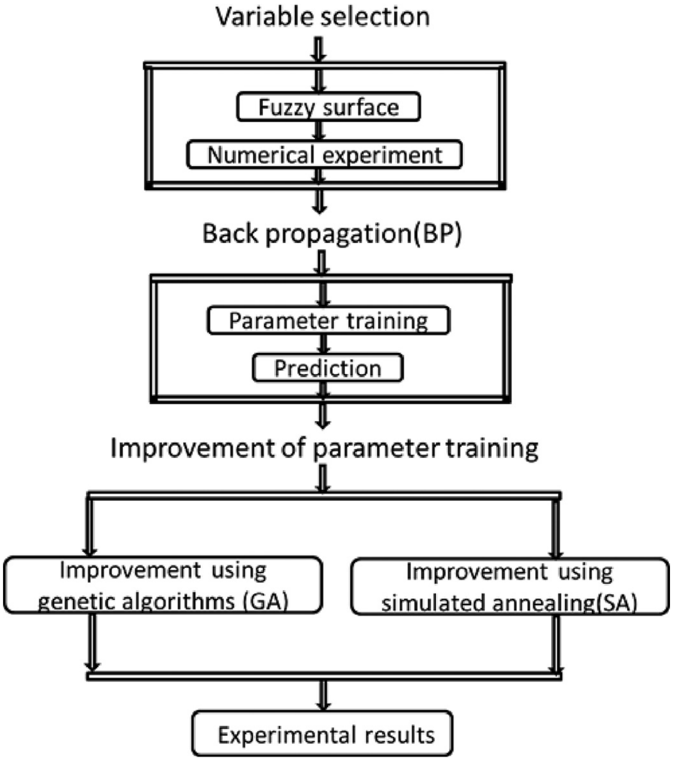

Fig. 2. Architecture of the experimental process.

compared the experimental results of the three forecasting models.

3. Variable selection

3.1. Fuzzy surfaces

In theory, a neural network based on nonlinear modeling techniques does not need to reduce the dimension of the input variable; however, it can easily reach regional minimum convergence. In addition, with the development of the information age, data has become more complex and commonly requires preprocessing. Therefore, in practice, we must reduce the dimension of the input variables prior to modeling.

It should be noted that few studies have attempted to identify significant input variables. Some researchers have selected input variables with no explanation, directly selecting adequate explanatory variables from previous studies which concluded that some variables were effective using the least squares method, stepwise regression, or neural networks.

Table 1
Identifying important input variables.

	Identified input variable	Eliminated variables
Iteration 1	v_{6}	$v_{37}, v_{42}, v_{41}, v_{35}, v_{48}, v_{47}, v_{65}$
Iteration 2	v_{36}	$v_{38}, v_{51}, v_{19}, v_{43}, v_{34}, v_{32}$
Iteration 3	v_{8}	$v_{18}, v_{21}, v_{23}, v_{20}, v_{22}, v_{24}$
...	...	...
Iteration 17	v_{55}	v_{28}
Iteration 18	v_{56}	v_{11}

As there are many factors that affect stock market returns, the data in this study has a high degree of nonlinear characteristics. Thus, we chose fuzzy curve analysis to select effective input variables for the ANN. Fuzzy curve analysis is based on the theory of fuzzy mathematics and does not require complicated mathematical modeling. First, we calculated the correlation between each input and output variable of the ANN. We then sorted all input variables according to importance. A relatively significant correlation exists between each input variable; thus, we excluded relevant variables by fuzzy surfaces and established a simple and optimal subset of input variables. The details of the algorithm are described in [22].

3.2. Numerical experiment

We used the first period runs (November 1993 to December 2007; 170 months of observations) to select optimal input variables using the fuzzy surface technique. The data includes 71 input variables and one output variable. The results are shown in Table 1.

According to the simulation, the significant input variables were identified and ranked in order of importance: v_{6}, v_{36}, v_{8}, v_{50}, v_{52}, v_{49}, v_{7}, v_{14}, v_{12}, v_{17}, v_{44}, v_{30}, v_{10}, v_{33}, v_{54}, v_{53}, v_{55}, and v_{56}. The selected variables were renamed, and the meaning of each variable is shown in the Appendix. The values of the input variables were preprocessed by normalizing within the range of 0 and 1 to minimize the effects of magnitude among the inputs and increase the effectiveness of the learning algorithm.

The output variable is the return of the Nikkei 225 index, which is computed as follows:

$$y_t = \frac{(P_t - P_{t-1})}{P_{t-1}},$$

where P_t is the value of the index for Month t. Note that dividends were not considered in this study.

Among the selected input variables, we found that some variables, e.g., T-bill rate, had been proven effective and have been used frequently by previous studies. However, most input variables have not been previously examined; therefore, we verified the predicted effects of these variables in the following models. In addition, due to the lag associated with the publication of macroeconomic indicators, we applied a one-month time lag to certain data. We consider that using these variables in the forecasting models is similar to real-world practice.

4. Back propagation neural network training

4.1. Parameter training

The BP algorithm is a widely applied classical learning algorithm for neural networks [23]. In the BP algorithm, we enter the in-sample data, and then the algorithm adjusts the weights and bias of the network by repeated training in such a way that the error between the desired output and the actual output is reduced. When the error is less than a specified value or when termination criteria are satisfied, training is completed and the weights and bias of the network are saved.

4.2. Numerical experiment

We used the in-sample data described in Section 2.1 for training in the numerical experiments. To optimize the ANN learning algorithm, we experimented with parameter settings for the network using the BP algorithm to determine the most appropriate parameters rather than choosing effective parameter values from a review of domain experts and prior research. The ANN parameters and their levels are summarized in Table 2. Ten levels of *n*, nine levels of *mc*, and ten levels of *ep* were tested in the experiments. However, it should be noted that, as suggested in the literature, a small value of *lr* (0.1) was selected. We obtained a different *MSE* value for each iteration when the ANN was trained by the same combination of parameters. Therefore, we ran the experiment once for each parameter combination.

The parameter setting experiments were performed with 900 parameter combinations. We selected the parameter combination that resulted in the best performance. Note that we used the mean square error *(MSE)* method to evaluate the performance of the ANN model:

$$MSE = \frac{1}{N}\sum_{t=1}^{N}\left(y_t - \hat{y}_t\right)^2,$$

where y_t denotes the actual return of the Nikkei 225 index, and $\hat{y}_t$ is the predicted return.

Among the 900 parameter combinations, we found that the most appropriate parameter combination was $n = 10$, $ep = 3000$, $mc = 0.4$, and $lr = 0.1$. The *MSE* value for the best BP training algorithm for the ANN (i.e., BPNN) model was 0.0017. The average *MSE* value obtained from the 900 training experiments was 0.1219.

Table 2
Tested ANN parameters and levels.

Parameter	Meaning	Level(s)
n	The number of neurons in the hidden layer	10, 20, … 100
ep	number of iterations	1000, 2000, … 10000
mc	momentum constant	0.1, 0.2, … 0.9
lr	value of learning rate	0.1

During the experiment, we observed the following characteristics.

- Calculation time increased with an increased number of neurons in the hidden layer; however, it was observed that the *MSE* value did not decrease gradually.
- Parameter combinations with relatively small *MSE* values always have relatively fewer neurons in the hidden layer. Note that the *MSE* value is relatively small when the number of neurons in the hidden layer ranges from 10–30.
- When the number of neurons in the hidden layer was large, the computer spent more time capturing nonlinear relationships among the variables. However, for many parameter combinations with a large number of neurons in the hidden layer, the experiment terminated in a short period of time. We speculate that the experiments achieved the best solution in the region of their starting point, which is the local minimum.
- Due to the complex nature and large volume of data, the time required to achieve convergence was significant; i.e., approximately one hour per parameter combination.

5. Improvement of parameter training

5.1. Improvement using genetic algorithms (GA)

As observed in the parameter setting experiments, we found that the BP algorithm has two drawbacks; i.e., trapping into local minima and slow convergence. Note that these drawbacks have been verified by previous studies. To overcome these problems, many studies have used optimal global search techniques rather than gradient search techniques such as the BP algorithm, which is designed for local search. Many studies have used GA-based hybrid models to overcome the drawbacks of the BP approach. The results of these studies support the notion that GAs can enhance the accuracy of ANN models and can reduce the time required for experiments [24-27]. In this section, a GA algorithm was utilized to optimize the initial weights and bias of the ANN model. Then, the ANN model was trained by the BP algorithm using the determined weights and bias.

We encoded all weights and the bias in a string and generated the initial population. Each solution generated by a GA is referred to as a chromosome (or individual). The collection of chromosomes is called a population. Here, each chromosome represents an ANN with a certain set of weights and bias. We evaluated each chromosome of the population using a fitness function that is based on *MSE*. Chromosomes with high fitness values participate in reproduction and yield new strings by the GA (e.g., crossover and mutation). Thus, we obtain a new population. Through iterative progression, and after many generations, the population with the best fitness values was found.

5.2. Improvement using simulated annealing

SA, which was first presented by Kirkpatrick, Gelatt and Vecchi [28], is a famous optimization method that can be widely and successfully employed to solve global optimization problems in many fields. The major advantage of SA is that it accepts both better and worse neighboring solutions that have a certain probability so as to jump out of a local optimum to search for the global optimum. Many prior studies have successfully applied the SA algorithm to optimize the structure of an ANN model in various applications [29-31]. In this section, the SA technique is also used to optimize the BP-trained weights and bias of an ANN model.

It should be noted that implementation of the SA algorithm is remarkably easier than a GA. In our study, the SA algorithm initiates with a relatively high temperature value to avoid being prematurely trapped in a local optimum. There are two loops in the SA algorithm. In the outer loop, temperature is changed, and the inner loop determines how many neighborhood moves will be attempted for each temperature. The algorithm proceeds by attempting a certain number of neighborhood moves for each temperature as the temperature parameter is gradually reduced [32].

5.3. Numerical results

Each model described above was estimated and validated by the in-sample data. The empirical evaluation of each model was based on the untouched out-of-sample data (January 2008 to July 2013; 67 months of observations) because the superior in-sample performance did not always guarantee validity of forecasting accuracy.

The *MSE* and CPU time (for training and prediction) results of the performance evaluations for each algorithm are shown in Table 3. The models were tested using the Windows 7 operating system. In addition, MathWorks MATLAB R2011a was employed in all experiments.

Nine hundred BP training experiments were executed, and we selected the best combination of parameters ($n=10$, $ep=3000$, $mc=0.4$, $lr=0.1$) for the BPNN model as a baseline for comparison with the other models. The average performance of the BP algorithm is shown in

Table 3
Error analyses of different forecasting models.

Models	BPNN best	BPNN average	GABPNN best	GABPNN average	SABPNN best	SABPNN average
MSE	0.0044	0.1077	0.0043	0.0090	0.0725	0.0862
CPU Time[*]	68		1080	1322	28	40

[*] CPU Time for a 2.66 GHz Intel Core 2 Duo E6750.

Table 3. Note that GABPNN denotes the hybrid GA and BP training algorithm used in the neural network. The cross probability and mutation probability values of the experiments were changed 81 times. This resulted in the best and average performances of the GABPNN model. SABPNN is a hybrid SA and BP training algorithm. The SABPNN experiments were performed 10 times, and the temperature of each experiment was 100. The best and average ability of the SABPNN to estimate stock returns are also shown in Table 3.

Here, we exclude computing time and give a simple comparison of the error indicator values. Note that smaller criterion values indicate better prediction effects. Compared to the BPNN average, GABPNN and SABPNN overcame the local minimum weakness and greatly improved prediction accuracy. SABPNN models with lower *MSE* values are superior to the average level of the BPNN experiments, especially for the hybrid GA and BP approach. Note that the *MSE* value of the best model is 0.0043. Compared to the BPNN average, the average *MSE* of the GABPNN model was also effective, demonstrating a value of 0.0090. The best BPNN model also demonstrated effective performance with an *MSE* value of 0.0044.

In terms of run time for the three models, the BPNN models may demonstrate the shortest time because they easily fall into the local minimum with bad performance. If not, the time required to reach convergence was very long; i.e., approximately one hour for each parameter combination with a large number of hidden neurons. There was no sense to provide the average CPU time for the BPNN. Computing time is 68 s when the best BPNN model is not caught in the local minimum. Although the run time of the best BPNN was short, excessive time was required to search for the most appropriate parameter combination for the BPNN models. The SABPNN required only 28 s, which helped the BP algorithm jump out of the local search. The average time for the SABPNN was less than the other models. The GABPNN required more time than the SABPNN; 18 min of run time was required, but it reduced the *MSE* value significantly.

The run time of the SABPNN was faster than any other model, and the prediction accuracy was higher than the normal BP model levels. By synthesizing the performance of the accuracy of prediction and the run time of the experiments, we consider that GABPNN demonstrates better market return prediction performance and higher accuracy than the other models.

The experimental results indicate that, even though most of the proposed 18 input variables have not been used in previous studies, their effect on prediction was remarkable. Thus, these input variables are considered a good choice for prediction of stock market returns. The optimization afforded by the GA or SA has demonstrated strong potential for obtaining globally optimal solutions. The GA can quickly achieve the best prediction accuracy for the BP models while the BP algorithm requires to test a large number of parameter combinations.

6. Conclusion

In this study, we have examined methods to predict the return of Nikkei 225 index using an ANN. To search for new and effective input variables for an ANN, we collected 71 variables with respect to different aspects of the Japanese stock market. We selected new combinations of input variables of 18 explanatory variables by fuzzy surfaces and utilized the combination to predict market returns. We employed monthly data obtained using the 18 variables to predict the return of the Nikkei 225 index, and then compared the prediction performance of the different models. The empirical results showed that the proposed 18 input variables can successfully predict the stock market returns.

For the classic BP training algorithm, we conducted an experiment with 900 parameter combinations. We then selected the best model and the average level of the parameter settings to compare other prediction models. To overcome the limitations of BP gradient search, we applied global search techniques (i.e., a GA and SA) to optimize the weights and bias of the ANN. Compared to the average level of BP model, the hybrid SA and BP approach overcame the weakness of local minima and greatly improved prediction accuracy. In addition, we synthesized the performance of the accuracy of prediction and run time. We consider that the hybrid GA and BP approach provides more accurate forecasting of future values than other prediction models.

In addition, it is possible that multi-collinearity problem and correlation are existed among the selected input variables. In the future, we consider these problems under the applications of various methods for reducing the dimension of the input variables. Future research also focus on the prediction of the direction of stock market index movement using an ANN and other models.

Appendix

The selected variables and the meaning of each variable are shown in Table A.1.

Table A.1
Input variables.

Input variables	Meaning
x_1	Average amounts outstanding of monetary base
x_2	Banknotes in circulation of average amounts outstanding of monetary base
x_3	Coins in circulation of average amounts outstanding of monetary base
x_4	Uncollateralized overnight of call rates at the end of month
x_5	Yen spot rate at the end of month of Tokyo market
x_6	Yen central rate at the end of month of Tokyo market
x_7	Yen lowest in the month of Tokyo market
x_8	Percent changes from the previous year in average amounts outstanding of money stock
x_9	Percentage changes in average amounts outstanding from the previous year of loans and discounts for total of major and regional banks
x_{10}	Loans and discounts of regional banks
x_{11}	Import price index of all commodities
x_{12}	Real exports
x_{13}	Real imports
x_{14}	Indices of industrial production
x_{15}	1-year T-bill rate
x_{16}	2-years T-bill rate
x_{17}	3-years T-bill rate
x_{18}	4-years T-bill rate

References

[1] Enke D, Thawornwong S. The use of data mining and neural networks for forecasting stock market returns. Expert Syst Appl 2005;29:927–40.

[2] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943;5:115–33.

[3] Motiwalla L, Wahab M. Predictable variation and profitable trading of US equities: a trading simulation using neural networks. Comput Oper Res 2000;27:1111–29.

[4] Hodnett K, Hsieh HH. Application of cascade-correlation neural networks in developing stock selection models for global equities. Int Bus Eco Res J 2012;11:375–96.

[5] Fernandez Rodrıguez F, Gonzalez Martel C, Sosvilla Rivero S. On the profitability of technical trading rules based on artificial neural networks: evidence from the Madrid stock market. Eco Lett 2000;69:89–94.

[6] Versace M, Bhatt R, Hinds O, Shiffer M. Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks. Expert Syst Appl 2004;27:417–25.

[7] Lu CJ. Integrating independent component analysis-based denoising scheme with neural network for stock price prediction. Expert Syst Appl 2010;37:7056–64.

[8] Wang Z, Andrews MA, Wu Z, Wang L, Bauch CT. Coupled disease–behavior dynamics on complex networks: a review. Phys Life Rev 2015;15:1–29.

[9] Wang Z, Kokubo S, Jusup M, Tanimoto J. Universal scaling for the dilemma strength in evolutionary games. Phys Life Rev 2015;14:1–30.

[10] Wang Z, Wang L, Szolnoki A, Perc M. Evolutionary games on multilayer networks: a colloquium. Eur Phys J B 2015;88:1–15.

[11] Wang Z, Xia C, Meloni S, Zhou C, Moreno Y. Impact of social punishment on cooperative behavior in complex networks. Sci Rep 2013;3:3055–62.

[12] Xia C, Meng X, Wang Z. Heterogeneous coupling between interdependent lattices promotes the cooperation in the prisoner's dilemma game. Plos One 2015;10:e0129542.

[13] Wang Z, Wang L, Yin Z, Xia C. Inferring reputation promotes the evolution of cooperation in spatial social dilemma games. Plos One 2012;7:40218.

[14] Chen AS, Leung MT, Daouk H. Application of neural networks to an emerging financial market: forecasting and trading the Taiwan stock index. Comput Oper Res 2003;30:901–23.

[15] Leung MT, Daouk H, Chen A-S. Forecasting stock indices: a comparison of classification and level estimation models. Int J Forecast 2000;16:173–90.

[16] Kara Y, Boyacioglu MA, Baykan ÖK. Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange. Expert Syst Appl 2011;38:5311–19.

[17] Wong BK, Bodnovich TA, Selvi Y. A bibliography of neural network business applications research: 1988–September 1994. Expert Syst 1995;12:253–61.

[18] Salchenberger LM, Cinar E, Lash NA. Neural networks: a new tool for predicting thrift failures. Decis Sci 1992;23:899–916.

[19] Lee Y, Oh SH, Kim MW. The effect of initial weights on premature saturation in back-propagation learning. Int Jt Conf Neural Netw 1991;1:765–70.

[20] Funahashi KI. On the approximate realization of continuous mappings by neural networks. Neural Netw 1989;2:183–92.

[21] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw 1989;2:359–66.

[22] Lin Y, Cunningham GA III, Coggeshall SV. Input variable identification—fuzzy curves and fuzzy surfaces. Fuzzy Set Syst 1996;82:65–71.

[23] Werbos PJ. The roots of backpropagation: from ordered derivatives to neural networks and political forecasting. New York: Wiley; 1994.

[24] Jadav K, Panchal M. Optimizing weights of artificial neural networks using genetic algorithms. Int J Adv Res Comput Sci Electron Eng 2012;1:47–51.

[25] Chang Y, Lin J, Shieh J, Abbod MF. Optimization the initial weights of artificial neural networks via genetic algorithm applied to hip bone fracture prediction. Adv Fuzzy Syst 2012;2012:1–9.

[26] Ishigami H, Fukuda T, Shibata T, Arai F. Structure optimization of fuzzy neural network by genetic algorithm. Fuzzy Set Syst 1995;71:257–64.

[27] Patra SR, Jehadeesan R, Rajeswari S, Murty SS, Baba MS. Development of genetic algorithm based neural network model for parameter estimation of fast breeder reactor subsystem. Int J Soft Comput Eng 2012;2:87–90.

[28] Kirkpatrick S, Gelatt CD Jr, Vecchi MP. Optimization by simmulated annealing. Sci 1983;220:671–80.

[29] Yamazaki A, De Souto M, Ludermir T. Optimization of neural network weights and architectures for odor recognition using simulated annealing. Proc Int Jt Conf Neural Netw 2002;1:547–52.

[30] Abbasi B, Mahlooji H. Improving response surface methodology by using artificial neural network and simulated annealing. Expert Syst Appl 2012;39:3461–8.

[31] Sudhakaran R, Murugan VV, Sivasakthivel P, Balaji M. Prediction and optimization of depth of penetration for stainless steel gas tungsten arc welded plates using artificial neural networks and simulated annealing algorithm. Neural Comput Appl 2013;22:637–49.

[32] Eglese R. Simulated annealing: a tool for operational research. Eur J Oper Res 1990;46:271–81.