

# Flexible neural trees ensemble for stock index modeling

Yuehui Chen<sup>a,\*</sup>, Bo Yang<sup>a,b</sup>, Ajith Abraham<sup>a,c</sup>

<sup>a</sup>*School of Information science and Engineering, Jinan University, 106 Jiwei Road, 250022 Jinan, PR China*

<sup>b</sup>*State Key Laboratory of Advanced Technology for Materials Synthesis and Processing, Wuhan University of Science and Technology, Wuhan, PR China*

<sup>c</sup>*School of Computer Science and Engineering, Chung-Ang University, Seoul, Korea*

Available online 14 October 2006

## Abstract

The use of intelligent systems for stock market predictions has been widely established. In this paper, we investigate how the seemingly chaotic behavior of stock markets could be well represented using flexible neural tree (FNT) ensemble technique. We considered the Nasdaq-100 index of Nasdaq Stock Market<sup>SM</sup> and the S&P CNX NIFTY stock index. We analyzed 7-year Nasdaq-100 main index values and 4-year NIFTY index values. This paper investigates the development of novel reliable and efficient techniques to model the seemingly chaotic behavior of stock markets. The structure and parameters of FNT are optimized using genetic programming (GP) like tree structure-based evolutionary algorithm and particle swarm optimization (PSO) algorithms, respectively. A good ensemble model is formulated by the local weighted polynomial regression (LWPR). This paper investigates whether the proposed method can provide the required level of performance, which is sufficiently good and robust so as to provide a reliable forecast model for stock market indices. Experimental results show that the model considered could represent the stock indices behavior very accurately.

© 2006 Elsevier B.V. All rights reserved.

*Keywords:* Flexible neural tree; GP-like tree structure-based evolutionary algorithm; Particle swarm optimization; Ensemble learning; Stock index

## 1. Introduction

Prediction of stocks is generally believed to be a very difficult task—it behaves like a random walk process and time varying. The obvious complexity of the problem paves the way for the importance of intelligent prediction paradigms. During the last decade, stocks and futures traders have come to rely upon various types of intelligent systems to make trading decisions [1–4,10,15,22,14]. Several intelligent systems have in recent years been developed for modeling expertise, decision support and complicated automation tasks [15,17,23,28,18].

Leigh et al. [16] introduced a method for combining template matching, using pattern recognition and a feed-forward neural network, to forecast stock market activity. The authors evaluated the effectiveness of the method for forecasting increases in the New York Stock Exchange Composite Index at a 5 trading day horizon. Results

indicate that the technique is capable of returning results that are superior to those attained by random choice.

Kim and Chun [13] explored a new architecture for graded forecasting using an arrayed probabilistic network (APN) and used a “mistake chart” to compare the accuracy of learning systems against default performance based on a constant prediction. Authors also evaluated several backpropagation models against a recurrent neural network (RNN) as well as probabilistic neural networks, etc.

Tsaih et al. [26] investigated a hybrid AI (artificial intelligence) approach to the implementation of trading strategies in the S&P 500 stock index futures market. The hybrid AI approach integrates the rule-based systems technique and the neural networks technique to accurately predict the direction of daily price changes in S&P 500 stock index futures. By highlighting the advantages and overcoming the limitations of both the neural networks technique and rule-based systems technique, the hybrid approach can facilitate the development of more reliable intelligent systems to model expert thinking and to support the decision-making processes.

\*Corresponding author.

*E-mail addresses:* [yhchen@ujn.edu.cn](mailto:yhchen@ujn.edu.cn) (Y. Chen), [yangbo@ujn.edu.cn](mailto:yangbo@ujn.edu.cn) (B. Yang), [ajith.abraham@ieee.org](mailto:ajith.abraham@ieee.org) (A. Abraham).

Refenes et al. [24] proposed a simple modification to the error backpropagation procedure which takes into account gradually changing input–output relations. The procedure is based on the principle of discounted least squares whereby learning is biased towards more recent observations with long term effects experiencing exponential decay through time. This is particularly important in systems in which the structural relationship between input and response vectors changes gradually over time but certain elements of long term memory are still retained. The procedure is implemented by a simple modification of the least-squares cost function commonly used in error back-propagation.

Van den Berg et al. [27] proposed a probabilistic fuzzy systems to develop financial models where one can identify different states of the market for modifying ones actions. Authors developed a Takagi-Sugeno (TS) probabilistic fuzzy systems that combine interpretability of fuzzy systems with the statistical properties of probabilistic systems. The methodology is applied to financial time series analysis and demonstrated how a probabilistic TS fuzzy system can be identified, assuming that a linguistic term set is given.

From the perspective of the agent-based model of stock markets, Chen and Liao [5] examined the possible explanations for the presence of the causal relation between stock returns and trading volume. Using the agent-based approach, the authors found that the explanation for the presence of the stock price volume relation may be more fundamental. Conventional devices such as information asymmetry, reaction asymmetry, noise traders or tax motives are not explicitly required. Authors claimed that a full understanding of the price volume relation cannot be accomplished unless the feedback relation between individual behavior at the bottom and aggregate phenomena at the top is well understood.

In this paper, we analyzed the seemingly chaotic behavior of two well-known stock indices namely the Nasdaq-100 index of Nasdaq<sup>SM</sup> [20] and the S&P CNX NIFTY stock index [21]. The Nasdaq-100 index reflects Nasdaq's largest companies across major industry groups, including computer hardware and software, telecommunications, retail/wholesale trade and biotechnology [20]. The Nasdaq-100 index is a modified capitalization-weighted index, which is designed to limit domination of the Index by a few large stocks while generally retaining the capitalization ranking of companies. Through an investment in the Nasdaq-100 index tracking stock, investors can participate in the collective performance of many of the Nasdaq stocks that are often in the news or have become household names. Similarly, S&P CNX NIFTY is a well-diversified 50 stock index accounting for 25 sectors of the economy [21]. It is used for a variety of purposes such as benchmarking fund portfolios, index-based derivatives and index funds. The CNX Indices are computed using market capitalization weighted method, wherein the level of the Index reflects the total market value of all the stocks in the

index relative to a particular base period. The method also takes into account constituent changes in the index and importantly corporate actions such as stock splits, rights, etc. without affecting the index value.

Our research is to investigate the performance analysis of FNT [7,9,6] ensemble for modeling the Nasdaq-100 and the NIFTY stock market indices. The hierarchical structure of FNT is evolved using GP with specific instructions. The parameters of the FNT model are optimized by PSO algorithm [12]. The proposed method interleaves both optimizations. Starting with random structures and corresponding parameters, it first tries to improve the structure and then as soon as an improved structure is found, it fine tunes its parameters. It then goes back to improving the structure again and, fine tunes the structure and rules' parameters. This loop continues until a satisfactory solution is found or a time limit is reached. The novelty of this paper is in the usage of flexible neural trees ensemble for selecting the important inputs and/or time delays and for forecasting models.

We analyzed the Nasdaq-100 index value from 11 January 1995 to 11 January 2002 [20] and the NIFTY index from 01 January 1998 to 03 December 2001 [21]. For both the indices, we divided the entire data into almost two equal parts. No special rules were used to select the training set other than ensuring a reasonable representation of the parameter space of the problem domain [3].

The rest of the paper is organized as follows. The flexible neural tree model and its design method are given in Section 2. Some simple and FNT ensemble approaches are presented in Section 3. Some simulation results on stock index modeling are shown in Section 4. Finally in Section 5, we present some conclusions and future works.

## 2. The flexible neural tree model

The function set  $F$  and terminal instruction set  $T$  used for generating a FNT model are described as  $S = F \cup T = \{+_2, +_3, \dots, +_N\} \cup \{x_1, \dots, x_n\}$ , where  $+_i (i = 2, 3, \dots, N)$  denote non-leaf nodes' instructions and taking  $i$  arguments.  $x_1, x_2, \dots, x_n$  are leaf nodes' instructions and taking no other arguments. The output of a non-leaf node is calculated as a flexible neuron model (see Fig. 1). From this point of view, the instruction  $+_i$  is also called a flexible neuron operator with  $i$  inputs.

In the creation process of neural tree, if a nonterminal instruction, i.e.,  $+_i (i = 2, 3, 4, \dots, N)$  is selected,  $i$  real values are randomly generated and used for representing the connection strength between the node  $+_i$  and its children. In addition, two adjustable parameters  $a_i$  and  $b_i$  are randomly created as flexible activation function parameters. For developing the FNT, the following flexible activation function is used,

$$f(a_i, b_i, x) = e^{-((x-a_i)/b_i)^2}. \quad (1)$$

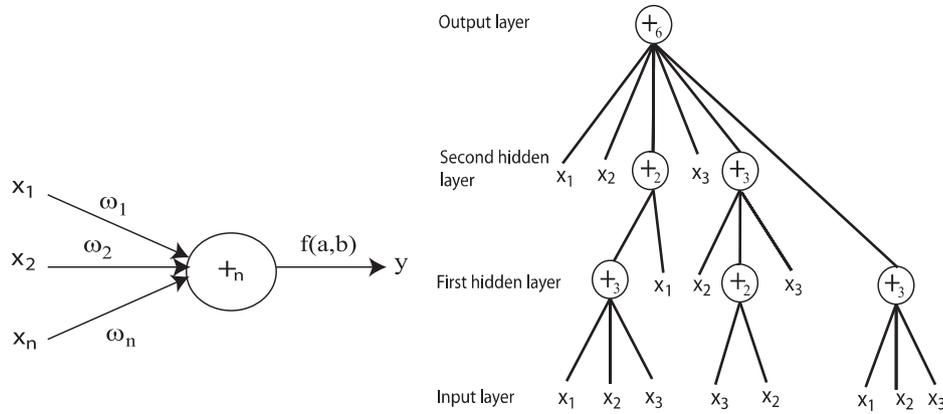


Fig. 1. A flexible neuron operator (left), and a typical representation of the FNT with function instruction set  $F = \{+2, +3, +4, +5, +6\}$ , and terminal instruction set  $T = \{x_1, x_2, x_3\}$  (right).

The total excitation of +<sub>n</sub> is  $net_n = \sum_{j=1}^n w_j * x_j$ , where  $x_j (j = 1, 2, \dots, n)$  are the inputs to node +<sub>n</sub>. The output of the node +<sub>n</sub> is then calculated by,

$$out_n = f(a_n, b_n, net_n) = e^{-((net_n - a_n)/b_n)^2}. \quad (2)$$

The overall output of flexible neural tree can be computed from left to right by depth-first method, recursively.

### 2.1. Tree structure optimization

Finding an optimal or near-optimal neural tree is formulated as a product of evolution. In our previous work, the probabilistic incremental program evolution (PIPE) and ant programming (AP) algorithm have been employed to find a near-optimal neural tree [7,8]. In this study, the crossover and selection operators used are same as those of standard GP. A number of neural tree mutation operators are developed as follows:

- (1) Changing one terminal node: randomly select one terminal node in the neural tree and replace it with another terminal node.
- (2) Changing all the terminal nodes: select each and every terminal node in the neural tree and replace it with another terminal node.
- (3) Growing: select a random leaf in hidden layer of the neural tree and replace it with a newly generated subtree.
- (4) Pruning: randomly select a function node in the neural tree and replace it with a terminal node.

### 2.2. Parameter optimization with PSO

The particle swarm optimization (PSO) conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA). A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector  $\mathbf{x}_i$ . A swarm of particles moves through the problem space, with the moving

velocity of each particle represented by a velocity vector  $\mathbf{v}_i$ . At each time step, a function  $f_i$  representing a quality measure is calculated by using  $\mathbf{x}_i$  as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector  $\mathbf{p}_i$ . Furthermore, the best position among all the particles obtained so far in the population is kept track of as  $\mathbf{p}_g$ . In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle. At each time step  $t$ , by using the individual best position,  $\mathbf{p}_i$ , and the global best position,  $\mathbf{p}_g(t)$ , a new velocity for particle  $i$  is updated by

$$\mathbf{v}_i(\mathbf{t} + 1) = \mathbf{v}_i(\mathbf{t}) + c_1 \phi_1(\mathbf{p}_i(\mathbf{t}) - \mathbf{x}_i(\mathbf{t})) + c_2 \phi_2(\mathbf{p}_g(\mathbf{t}) - \mathbf{x}_i(\mathbf{t})), \quad (3)$$

where  $c_1$  and  $c_2$  are positive constant and  $\phi_1$  and  $\phi_2$  are uniformly distributed random number in  $[0,1]$ . The term  $\mathbf{v}_i$  is limited to the range of  $\pm \mathbf{v}_{max}$ . If the velocity violates this limit, it is set to its proper limit. Changing velocity this way enables the particle  $i$  to search around its individual best position,  $\mathbf{p}_i$ , and global best position,  $\mathbf{p}_g$ . Based on the updated velocities, each particle changes its position according to the following equation:

$$\mathbf{x}_i(\mathbf{t} + 1) = \mathbf{x}_i(\mathbf{t}) + \mathbf{v}_i(\mathbf{t} + 1). \quad (4)$$

### 2.3. Feature/input selection with FNT

It is often a difficult task to select important variables for a forecasting or classification problem, especially when the feature space is large. A fully connected NN classifier usually cannot do this. In the perspective of FNT framework, the nature of model construction procedure allows the FNT to identify important input features in building a forecasting model that is computationally efficient and effective. The mechanisms of input selection in the FNT constructing procedure are as follows. (1) Initially the input variables are selected to formulate the FNT model with same probabilities; (2) The variables which have more contribution to the objective function will be enhanced and

have high opportunity to survive in the next generation by a evolutionary procedure; (3) The evolutionary operators i.e., crossover and mutation, provide a input selection method by which the FNT should select appropriate variables automatically.

#### 2.4. Procedure of the general learning algorithm

The general learning procedure for constructing the FNT model can be described as follows:

- (1) Create an initial population randomly (FNT trees and its corresponding parameters);
- (2) Structure optimization is achieved by the neural tree variation operators as described in Section 2.
- (3) If a better structure is found, then go to step (4), otherwise go to step (2);
- (4) Parameter optimization is achieved by the PSO algorithm as described in Section 2. In this stage, the architecture of FNT model is fixed, and it is the best tree developed during the end of run of the structure search. The parameters (weights and flexible activation function parameters) encoded in the best tree formulate a particle.
- (5) If the maximum number of local search is reached, or no better parameter vector is found for a significantly long time then go to step (6); otherwise go to step (4);
- (6) If satisfactory solution is found, then the algorithm is stopped; otherwise go to step (2).

### 3. The FNT ensemble

For most regression and classification problems, combining the outputs of several predictors improves on the performance of a single generic one [25]. Formal support to this property is provided by the so-called bias/variance dilemma [11], based on a suitable decomposition of the prediction error. According to these ideas, good ensemble members must be both accurate and diverse, which poses the problem of generating a set of predictors with reasonably good individual performances and independently distributed predictions for the test points. Diverse individual predictors can be obtained in several ways. These include: (i) using different algorithms to learn from the data (classification and regression trees, artificial neural networks (ANNs), support vector machines, etc.), (ii) changing the internal structure of a given algorithm (for instance, number of nodes/depth in trees or architecture in neural networks), and (iii) learning from different adequately-chosen subsets of the data set. The probability of success in strategy (iii), the most frequently used, is directly tied to the instability of the learning algorithm [3]. That is, the method must be very sensitive to small changes in the structure of the data and/or in the parameters defining the learning process. Again, classical examples in this sense are classification and regression trees and ANNs. In particular, in the case of ANNs the instability comes naturally from

the inherent data and training process randomness, and also from the intrinsic non-identifiability of the model. In what follows, three ensemble methods are employed for the stock index forecasting problems.

#### 3.1. The basic ensemble method

A simple approach to combining network outputs is to simply average them together. The basic ensemble method (BEM) output is defined:

$$f_{\text{BEM}} = \frac{1}{n} \sum_{i=1}^n f_i(x). \quad (5)$$

This approach by itself can lead to improved performance, but does not take into account the fact that some FNTs may be more accurate than others. It has the advantage that it is easy to understand and implement and can be shown not to increase the expected error.

#### 3.2. The generalized ensemble method

A generalization to the BEM method is to find weights for each output that minimize the positive and negative classification rates of the ensemble. The general ensemble method (GEM) is defined:

$$f_{\text{BEM}} = \sum_{i=1}^n \alpha_i f_i(x), \quad (6)$$

where the  $\alpha_i$ 's are chosen to minimize the root mean square error between the FNT outputs and the desired values. For comparison purpose, the optimal weights of the ensemble predictor are optimized by using PSO algorithm.

#### 3.3. The LWPR method

To investigate more efficient ensemble method, a LWPR approximation approach is employed in this work [19]. In this framework, the final output of FNT ensemble is approximated by a local polynomial model, i.e.,

$$f_{\text{LWPR}} = \sum_{i=1}^M \beta_i t_i(x), \quad (7)$$

where  $t_i$  is a function that produces the  $i$ th term in the polynomial. For example, with two inputs and a quadratic local model we would have  $t_1(x) = 1$ ,  $t_2(x) = x_1$ ,  $t_3(x) = x_2$ ,  $t_4(x) = x_1^2$ ,  $t_5(x) = x_1 x_2$ ,  $t_6(x) = x_2^2$ . Eq. (5) can be written more compactly as

$$f_{\text{LWPR}} = \beta^T t(x), \quad (8)$$

where  $t(x)$  is the vector of polynomial terms of the input  $x$  and  $\beta$  is the vector of weight terms. The weight of the  $i$ th datapoint is computed as a decaying function of Euclidean distance between  $x_k$  and  $x_{\text{query}}$ .  $\beta$  is chosen to minimize

$$\sum_{i=1}^N \omega_i^2 (f_{\text{LWPR}} - \beta^T t(x)), \quad (9)$$

where  $\omega_i$  is a Gaussian weight function with kernel width  $K$ :

$$\omega_i = \exp(-\text{Distance}^2(x_i, x_{\text{query}})/2K^2). \quad (10)$$

For this problem, an algorithm based on a multiresolution search of a quickly constructible augmented kd-tree without needing to rebuild the tree, has been proposed for fast predictions with arbitrary local weighting functions [19].

#### 4. Experiments

We considered 7-year stock data for the Nasdaq-100 Index and 4-year for the NIFTY index. Our target is to develop efficient forecast models that could predict the index value of the following trade day based on the opening, closing and maximum values of the same on a given day. The assessment of the prediction performance of the different ensemble paradigms were done by quantifying

Table 1  
Empirical comparison of RMSE results for four learning methods

	Best-FNT	BEM	GEM	LWPR
Nasdaq-100	0.01854	0.01824	0.01635	$4.41 \times 10^{-5}$
NIFTY	0.01315	0.01258	0.01222	$1.96 \times 10^{-7}$

Table 2  
Statistical analysis of four learning methods (test data)

	Best-FNT	BEM	GEM	LWPR
<i>Nasdaq-100</i>				
CC	0.997542	0.997610	0.997757	0.999999
MAP	98.1298	98.3320	97.3347	0.4709
MAPE	6.1090	6.3370	5.7830	0.0040
<i>NIFTY</i>				
CC	0.996908	0.997001	0.997109	0.999999
MAP	28.0064	34.3687	26.8188	$7.65 \times 10^{-4}$
MAPE	3.2049	2.9303	2.6570	$1.92 \times 10^{-5}$

the prediction obtained on an independent data set. The root mean squared error (RMSE), maximum absolute percentage error (MAP) and mean absolute percentage error (MAPE) and correlation coefficient (CC) were used to study the performance of the trained forecasting model for the test data. *MAP* is defined as follows:

$$MAP = \max \left( \frac{|P_{\text{actual},i} - P_{\text{predicted},i}|}{P_{\text{predicted},i}} \times 100 \right), \quad (11)$$

where  $P_{\text{actual},i}$  is the actual index value on day  $i$  and  $P_{\text{predicted},i}$  is the forecast value of the index on that day. Similarly *MAPE* is given as

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left( \frac{|P_{\text{actual},i} - P_{\text{predicted},i}|}{P_{\text{predicted},i}} \right) \times 100, \quad (12)$$

where  $N$  represents the total number of days.

We used instruction set  $I = \{+2, +3, \dots, +6, x_0, x_1, x_2\}$  for modeling the Nasdaq-100 index and instruction set  $I = \{+2, +3, \dots, +8, x_0, x_1, x_2, x_3, x_4\}$  for modeling the NIFTY index. We have conducted 10 FNT models for predicting the Nasdaq-100 index and the NIFTY index, respectively. And then three ensemble methods discussed in Section 3 are employed to predict the both index.

Table 1 summarizes the test results achieved for the two stock indices using the four different approaches. Performance analysis of the trained forecasting models for the test data was shown in Table 2. Figs. 2 and 3 depict the test results for the one day ahead prediction of the Nasdaq-100 index and the NIFTY index, respectively.

#### 5. Conclusions

In this paper, we have demonstrated how the chaotic behavior of stock indices could be well represented by FNT ensemble learning paradigm. Empirical results on the two data sets using FNT ensemble models clearly reveal the efficiency of the proposed techniques. In terms of RMSE values, for the Nasdaq-100 index and the NIFTY index,

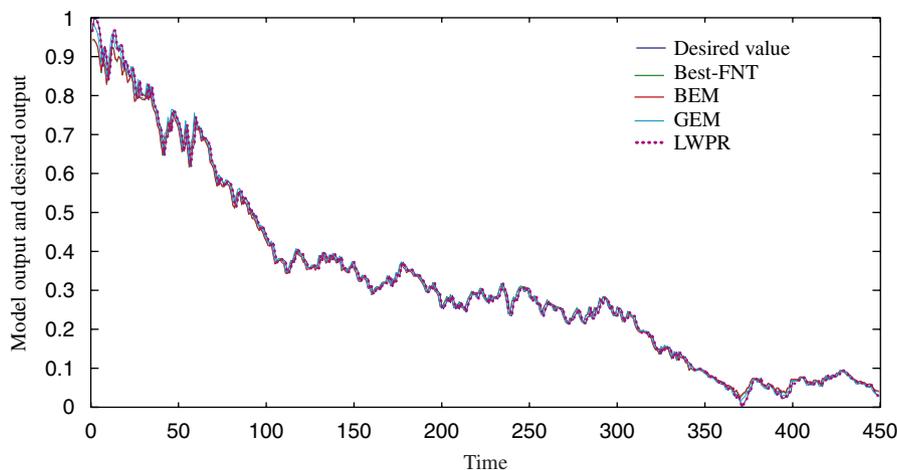


Fig. 2. Test results showing the performance of the different methods for modeling the Nasdaq-100 index.

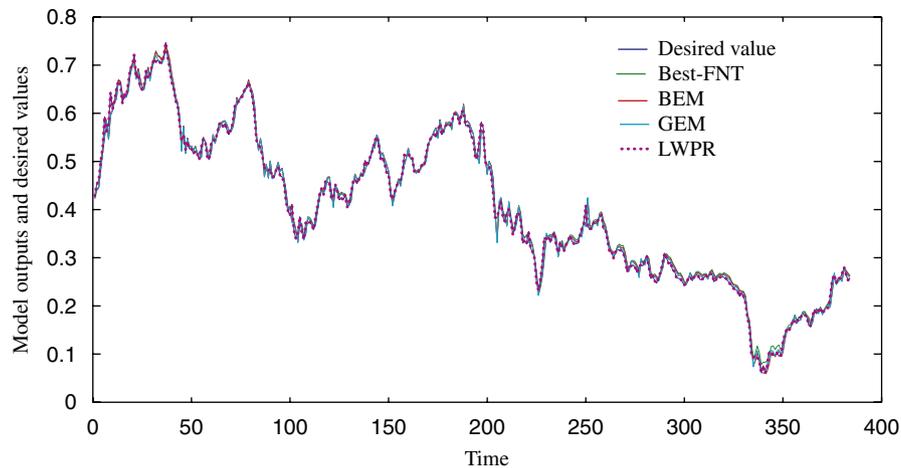


Fig. 3. Test results showing the performance of the different methods for modeling the NIFTY index.

LWPR performed marginally better than other models. For both indices (test data), LWPR also has the highest correlation coefficient and the lowest value of MAPE and MAP values. A low MAP value is a crucial indicator for evaluating the stability of a market under unforeseen fluctuations. In the present example, the predictability assures the fact that the decrease in trade off is only a temporary cyclic variation that is perfectly under control.

Our research was to predict the share price for the following trade day based on the opening, closing and maximum values of the same on a given day. The experimental results indicate that the most prominent parameters that affect share prices are their immediate opening and closing values. The fluctuations in the share market are chaotic in the sense that they heavily depend on the values of their immediate forerunning fluctuations. Long-term trends exist, but are slow variations and this information is useful for long-term investment strategies.

Our study focus on short term, on floor trades, in which the risk is higher. However, the results of our study show that even in the seemingly random fluctuations, there is an underlying deterministic feature that is directly enciphered in the opening, closing and maximum values of the index of any day making predictability possible. Empirical results also show that LWPR is a distinguished candidate for the FNT ensemble or neural networks ensemble.

### Acknowledgments

This research was partially supported by the Natural Science Foundation of China under grant number 60573065, and The Provincial Science and Technology Development Program of Shandong under Grant number SDSP2004-0720-03.

### References

[1] A. Abraham, B. Nath, P.K. Mahanti, in: V.N. Alexandrov, et al. (Eds.), *Hybrid Intelligent Systems for Stock Market Analysis*,

Computational Science, Springer-Verlag, Germany, USA, 2001, pp. 337–345.

- [2] A. Abraham, N.S. Philip, B. Nath, P. Saratchandran, Performance analysis of connectionist paradigms for modeling chaotic behavior of stock indices, *Second International Workshop on Intelligent Systems Design and Applications, Computational Intelligence and Applications*, Dynamic Publishers Inc., USA, 2002, pp. 181–186.
- [3] A. Abraham, N.S. Philip, P. Saratchandran, Modeling chaotic behavior of stock indices using intelligent paradigms, *Int. J. Neural Parallel Sci. Comput.* 11 (1–2) (2003) 143–160.
- [4] W.S. Chan, W.N. Liu, Diagnosing shocks in stock markets of southeast Asia, Australia, and New Zealand, *Math. Comput. Simulation* 59 (1–3) (2002) 223–232.
- [5] S.-H. Chen, C.-C. Liao, Agent-based computational modeling of the stock price volume relation, *Inf. Sci.* 170 (1) (2005) 75–100.
- [6] Y. Chen, A. Abraham, Feature selection and intrusion detection using hybrid flexible neural tree, *Lect. Notes Comput. Sci.* 3498 (2005) 439–444.
- [7] Y. Chen, B. Yang, J. Dong, Nonlinear system modeling via optimal design of neural trees, *Int. J. Neural Syst.* 14 (2) (2004) 125–137.
- [8] Y. Chen, B. Yang, J. Dong, Evolving flexible neural networks using ant programming and PSO algorithm, *International Symposium on Neural Networks (ISNN'04)*, Lecture Notes on Computer Science, vol. 3173, 2004, pp. 211–216.
- [9] Y. Chen, B. Yang, J. Dong, A. Abraham, Time-series forecasting using flexible neural tree model, *Inf. Sci.* 174 (3/4) (2005) 219–235.
- [10] E.S. Francis Tay, L.J. Cao, Modified support vector machines in financial time series forecasting, *Neurocomputing* 48 (1–4) (2002) 847–861.
- [11] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, *Neural Comput.* 4 (1992) 1–58.
- [12] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.
- [13] S.H. Kim, S.H. Chun, Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index, *Int. J. Forecast.* 14 (3) (1998) 323–337.
- [14] K.J. Kim, I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, *Expert Syst. Appl.* 19 (2) (2000) 125–132.
- [15] W. Leigh, N. Modani, S. Purvis, T. Roberts, Stock market trading rule discovery using technical charting heuristics, *Expert Syst. Appl.* 23 (2) (2002) 155–159.
- [16] W. Leigh, M. Paz, R. Purvis, An analysis of a hybrid neural network and pattern recognition technique for predicting short-term increases in the NYSE composite index, *Omega* 30 (2) (2002) 69–76.

- [17] W. Leigh, R. Purvis, J.M. Ragusa, Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network and genetic algorithm: a case study in romantic decision support, *Decision Support Syst.* 32 (4) (2002) 361–377.
- [18] I. Maqsood, M.R. Khan, A. Araham, Neural network ensemble method for weather forecasting, *Neural Comput. Appl.* 13 (2) (2004) 112–122.
- [19] A. Moore, J. Schneider, K. Deng, Efficient locally weighted polynomial regression predictions, in: *Proceedings of the 14th International Conference on Machine Learning*, 1997, pp. 236–244.
- [20] Nasdaq Stock Market<sup>SM</sup>, (<http://www.nasdaq.com>).
- [21] National Stock Exchange of India Limited, (<http://www.nse-india.com>).
- [22] K.J. Oh, K.J. Kim, Analyzing stock market tick data using piecewise nonlinear model, *Expert Syst. Appl.* 22 (3) (2002) 249–255.
- [23] T.S. Quah, B. Srinivasan, Improving returns on stock investment through neural network selection, *Expert Syst. Appl.* 17 (4) (1999) 295–301.
- [24] A.N. Refenes, Y. Bentz, D.W. Bunn, A.N. Burgess, A.D. Zaprani, Financial time series modelling with discounted least squares back-propagation, *Neurocomputing* 14 (2) (1997) 123–138.
- [25] A.J.C. Sharkey (Ed.), *Combining Artificial Neural Nets*, Springer, London, 1999.
- [26] R. Tsaih, Y. Hsu, C.C. Lai, Forecasting S&P 500 stock index futures with a hybrid AI system, *Decision Support Syst.* 23 (2) (1998) 161–174.
- [27] J. Van den Berg, U. Kaymak, W.-M. Van den Bergh, Financial markets analysis by using a probabilistic fuzzy modelling approach, *Int. J. Approximate Reasoning* 35 (3) (2004) 291–305.
- [28] Y.F. Wang, Mining stock price using fuzzy rough set system, *Expert Syst. Appl.* 24 (1) (2002) 13–23.



**Yuehui Chen** was born in 1964 in Shandong Province of China. He received his B.Sc. degree in mathematics/automatics from the Shandong University of China in 1985, and Master and Ph.D. degree in electrical engineering from the Kumamoto University of Japan in 1999 and 2001. During 2001C2003, he had worked as the Senior Researcher of the Memory-Tech Corporation at Tokyo. Since 2003 he has been a member at the Faculty of Electrical Engineering in Jinan University, where he is currently head of the Laboratory of Computational Intelligence. His

research interests include Evolutionary Computation, Neural Networks, Fuzzy Logic Systems, Hybrid Computational Intelligence and their

applications in time-series prediction, system identification, intelligent control, intrusion detection systems, web intelligence and bioinformatics. He is the author and co-author of more than 70 technique papers. Professor Yuehui Chen is a member of IEEE, the IEEE Systems, Man and Cybernetics Society and the Computational Intelligence Society, a member of Young Researchers Committee of the World Federation on Soft Computing, and a member of CCF Young Computer Science and Engineering Forum of China. More information at: <http://cilab.ujn.edu.cn>



**Bo Yang** is a professor and vice-president of Jinan University, Jinan, China. He is the Director of the Provincial Key Laboratory of Information and Control Engineering, and also acts as the Associate Director of Shandong Computer Federation, and Member of the Technical Committee of Intelligent Control of Chinese Association of Automation. His main research interests include computer networks, artificial intelligence, machine learning, knowledge discovery, and data mining. He has published numerous papers and gotten some of important scientific awards in this area.



**Ajith Abraham** currently works as a Professor under the South Korean Government's Institute of Information Technology Assessment (IITA) Professorship program at Chung-Ang University, Korea. He is also a visiting researcher of Rovira i Virgili University, Spain and an Adjunct Professor of Jinan University, China and Dalian Maritime University, China. His primary research interests are in computational intelligence with a focus on using evolutionary computation techniques for designing intelligent paradigms.

Application areas include Web services, information security, Web intelligence, financial modeling, multi-criteria decision-making, data mining, etc. He has authored/co-authored over 200 research publications in peer reviewed reputed journals, book chapters and conference proceedings of which three have won "best paper" awards. He is serving the Editorial board of over a dozen International Journals and has also guest edited 15 special issues for reputed International Journals. Since 2001, he is actively involved in the Hybrid Intelligent Systems (HIS) and the Intelligent Systems Design and Applications (ISDA) series of annual International conferences. He received PhD degree from Monash University, Australia. More information at: <http://www.softcomputing.net>