

Stock Index Modeling using EDA based Local Linear Wavelet Neural Network

Yuehui Chen
School of Information Science and
Engineering
Jinan University
Jiwei road 106, Jinan 250022,
P.R.China
E-mail: yhchen@ujn.edu.cn

Xiaohui Dong
School of Information Science and
Engineering
Jinan University
Jiwei road 106, Jinan 250022,
P.R.China
E-mail: nicop2@ujn.edu.cn

Yaou Zhao
School of Information Science and
Engineering
Jinan University
Jiwei road 106, Jinan 250022,
P.R.China
E-mail: yaou_zhao@yahoo.com.cn

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 Local Linear Wavelet Neural Network (LLWNN) technique. To this end, we considered the Nasdaq-100 index of Nasdaq Stock MarketSM 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 LLWNN are optimized by using Estimation of Distribution Algorithm (EDA). 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. Experiment results shown that the model considered could represent the stock indices behavior very accurately.

I. 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][2]. Several intelligent systems have in recent years been developed for modeling expertise, decision support and complicated automation tasks [3][4]. In this paper, we analyzed the seemingly chaotic behavior of two well-known stock indices namely the Nasdaq-100 index of NasdaqSM [5] and the S&P CNX NIFTY stock index [6]. The Nasdaq-100 index reflects Nasdaq's largest companies across major industry groups, including computer hardware and software, telecommunications, retail/wholesale trade and biotechnology [5]. 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 [6]. 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.

In our previous work, the neural network (NN) and flexible neural tree (FNT) have been employed for stock index modeling [7][9]. This research is to investigate the performance analysis of LLWNN for modeling the Nasdaq-100 and the NIFTY stock market indices. The parameters of the LLWNN model are optimized by EDA. We analyzed the Nasdaq-100 index value from 11 January 1995 to 11 January 2002 [5] and the NIFTY index from 01 January 1998 to 03 December 2001 [6]. 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 [2].

II. LOCAL LINEAR WAVELET NEURAL NETWORK

In terms of wavelet transformation theory, wavelets in the following form

$$\Psi = \{\Psi_i = |a_i|^{-\frac{1}{2}} \phi(\frac{x - b_i}{a_i}) : a_i, b_i \in \mathbb{R}^n, i \in \mathbb{Z}\} \quad (1)$$

$$x = (x_1, x_2, \dots, x_n)$$

$$a_i = (a_{i1}, a_{i2}, \dots, a_{in})$$

$$b_i = (b_{i1}, b_{i2}, \dots, b_{in})$$

are a family of functions generated from one single function $\psi(x)$ by the operation of dilation and translation. $\psi(x)$,

which is localized in both the time space and the frequency space, is called a mother wavelet and the parameters a_i and b_i are named the scale and translation parameters, respectively. The x represents inputs to the WNN model.

In the standard form of wavelet neural network, the output of a WNN is given by

$$f(x) = \sum_{i=1}^M \omega_i \Psi(x) = \sum_{i=1}^M \omega_i |a_i|^{-\frac{1}{2}} \psi\left(\frac{x-b_i}{a_i}\right) \quad (2)$$

where ψ_i is the wavelet activation function of i th unit of the ω_i is the weight connecting the i th unit of the hidden layer to the output layer unit. Note that for the n -dimensional input space, the multivariate wavelet basis function can be calculated by the tensor product of n single wavelet basis functions as follows

$$\psi(x) = \prod_{i=1}^n \psi(x_i) \quad (3)$$

Obviously, the localization of the i th units of the hidden layer is determined by the scale parameter a_i and the translation parameter b_i . According to the previous researches, the two parameters can either be predetermined based upon the wavelet transformation theory or be determined by a training algorithm. Note that the above wavelet neural network is a kind of basis function neural network in the sense of that the wavelets consists of the basis functions.

Note that an intrinsic feature of the basis function networks is the localized activation of the hidden layer units, so that the connection weights associated with the units can be viewed as locally accurate piecewise constant models whose validity for a given input is indicated by the activation functions. Compared to the multilayer perceptron neural network, this local capacity provides some advantages such as the learning efficiency and the structure transparency. However, the problem of basis function networks is also led by it. Due to the crudeness of the local approximation, a large number of basis function units have to be employed to approximate a given system. A shortcoming of the wavelet neural network is that for higher dimensional problems many hidden layer units are needed.

In order to take advantage of the local capacity of the wavelet basis functions while not having too many hidden units, here we propose an alternative type of wavelet neural network. The architecture of the proposed LLWNN [8] is shown in Fig.1. Its output in the output layer is given by

$$y = \sum_{i=1}^M (\omega_{i0} + \omega_{i1}x_1 + \dots + \omega_{in}x_n) \Psi_i(x)$$

$$= \sum_{i=1}^M (\omega_{i0} + \omega_{i1}x_1 + \dots + \omega_{in}x_n) |a_i|^{-\frac{1}{2}} \psi\left(\frac{x-b_i}{a_i}\right) \quad (4)$$

where $x = [x_1, x_2, \dots, x_n]$. Instead of the straightforward weight ω_i (piecewise constant model), a linear model

$$v_i = \omega_{i0} + \omega_{i1}x_1 + \dots + \omega_{in}x_n \quad (5)$$

is introduced. The activities of the linear models $v_i (i = 1, 2, \dots, M)$ are determined by the associated locally active wavelet functions $\psi_i(x) (i = 1, 2, \dots, M)$, thus v_i is only locally significant. The motivations for introducing the local linear models into a WNN are as follows: (1) Local linear models have been studied in some neurofuzzy systems and shown good performances [18], [19]; and (2) Local linear models should provide a more parsimonious interpolation in high-dimension spaces when modeling samples are sparse.

The scale and translation parameters and local linear model parameters are randomly initialized at the beginning and are optimized by a EDA discussed in the following section.

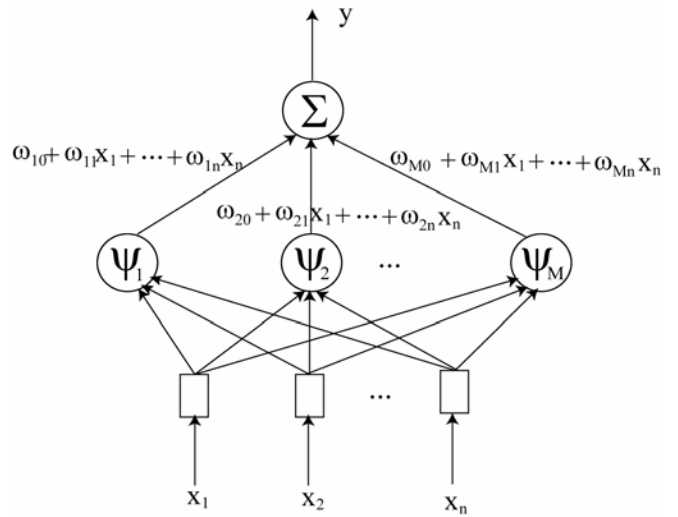


Fig. 1. A local linear wavelet neural network

III. LLWNN TRAINING

Estimation of distribution algorithms (EDAs) [11] [12] [14] [16] [17] are a new class of evolutionary algorithms. Like other evolutionary algorithms, EDAs maintain and successively improve a population of potential solutions until some stopping condition is met. However, EDAs do not use crossover or mutation. Instead, they select the best solutions from the current population and explicitly extract global statistical information from the selected solutions. A

posterior probability distribution model of promising solutions is built, based on the extracted information. Then new solutions are sampled from the model thus built and fully or in part replace solutions in the current population. More precisely, EDAs work as follows:

- S0 Randomly pick a set of solutions to form the initial population.
- S1 Select some solutions from the current population according to a selection method. Build the probability model of the selected solutions.
- S2 Replace some or all of the members of the current population by new solutions sampled from the probability model.
- S3 If the stopping condition are not met, go to Step 1.

Several EDAs have been proposed for solving global optimization problems. In these existing algorithms, the probability distribution of the promising solutions are modeled by a Gaussian distribution, a Gaussian mixture or a histogram. Since many points are needed to build a good probability model, these algorithms are often very time-consuming in practice.

One of the major issues in EDAs is how to select parents. A widely used selection method in EDA is the truncation selection. In the truncation selection, individuals are sorted according to their objective function values. Only the best individuals are selected as parents.

Another major issue in EDAs is how to build a probability distribution model $p(x)$. In EDAs for the global continuous optimization problem, the probabilistic model $p(x)$ can be a Gaussian distribution [13], a Gaussian mixture [14][11], a histogram [15], or a Gaussian model with diagonal covariance matrix (GM/DCM) [14].

GM/DCM is used in our algorithm. In GM/DCM, the joint density function of the k -th generation is written as follows:

$$p_k(x) = \prod_{i=1}^n N(x_i; \mu_i^k, \sigma_i^k) \quad (6)$$

where

$$N(x_i; \mu_i^k, \sigma_i^k) = \frac{1}{\sqrt{2\pi\sigma_i^k}} \exp\left(-\frac{1}{2}\left(\frac{x_i - \mu_i^k}{\sigma_i^k}\right)^2\right) \quad (7)$$

In (2), the n -dimensional joint probability distribution is factorized as a product of n univariate and independent normal distributions. There are two parameters for each variable required to be estimated in the k -th generation: the mean, μ_i^k , and the standard deviation, σ_i^k . They can be estimated as follows:

$$\hat{\mu}_i^k = \bar{x}_i^k = \frac{1}{M} \sum_{j=1}^M x_{ji}^k \quad (8)$$

$$\hat{\sigma}_i^k = \sqrt{\frac{1}{M} \sum_{j=1}^M (x_{ji}^k - \bar{x}_j^k)^2} \quad (9)$$

Before describing details of the algorithm for training LLWNN, the issue of coding is presented. Coding concerns the way the weights, dilation and translation parameters of LLWNN are represented by individuals or particles. A float point coding scheme is adopted here. For LLWNN coding, suppose there are M nodes in hidden layer and n input variables, then the total number of parameters to be coded is $(2n+n+1) * M = (3n+1)M$. The coding of a LLWNN into an individual or particle is as follows:

$$|a_{11}b_{11} \dots a_{1n}b_{1n} \omega_{10} \omega_{11} \dots \omega_{1n} | a_{21}b_{21} \dots a_{2n}b_{2n} \omega_{20} \omega_{21} \dots \omega_{2n} | \dots | a_{n1}b_{n1} \dots a_{nn}b_{nn} \omega_{n0} \omega_{n1} \dots \omega_{nn} |$$

The simple loop of the proposed training algorithm for local linear wavelet neural network is as follows.

- S1 Initial population is generated randomly.
- S2 Parameter optimization with EDA;
- S3 If the satisfactory solution is found or maximum number of generations is reached then stop; otherwise goto step S2.

TABLE I

THE RMSE RESULTS OF LLWNN AND WNN MODELS FOR TEST DATA SETS

Index	LLWNN	WNN
Nasdaq-100	0.01804	0.01925
NIFTY	0.01235	0.014321

TABLE II

STATISTICAL ANALYSIS OF THE LEARNING METHODS (TEST DATA)

	LLWNN	WNN
Nasdaq-100		
CC	0.997542	0.987610
MAP	99.1298	98.3320
MAPE	6.1090	6.3370
NIFTY		
CC	0.998908	0.992001
MAP	21.0064	32.3687
MAPE	1.2049	2.9303

IV. 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 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:

$$MAP = \max\left(\frac{P_{actual,i} - P_{predicted,i}}{P_{predicted,i}} \times 100\right) \quad (10)$$

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

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left(\frac{P_{actual,i} - P_{predicted,i}}{P_{predicted,i}} \times 100 \right) \quad (11)$$

where N represents the total number of days.

We used LLWNN with architecture {3-8-1} for modeling the Nasdaq-100 index and a LLWNN with architecture {5-8-1} for modeling the NIFTY index. For comparison purpose, two WNNs trained by EDA are also employed to predict the same stock indices.

Table 1 summarizes the test results achieved for the two stock indices using the proposed approach. Performance analysis of the trained forecasting models for the test data was shown in Table 2. Figures 2 and 3 depict the test results for the one day ahead prediction of the Nasdaq/100 index and the NIFTY index respectively.

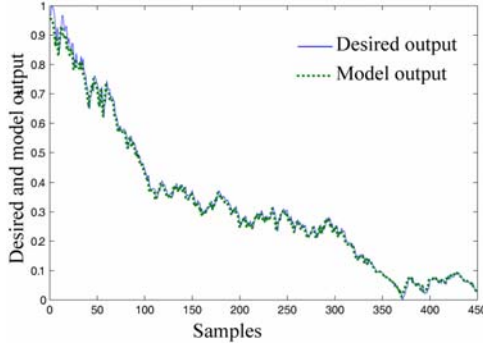


Fig. 2. Test results showing the performance of the LLWNN for modeling the Nasdaq-100 index

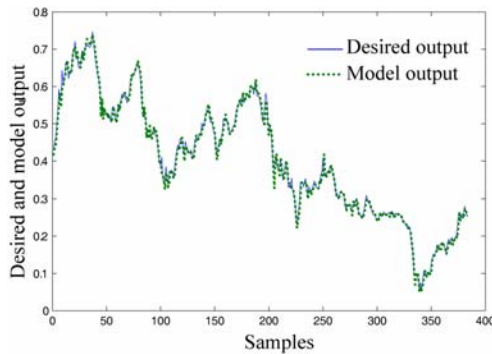


Fig. 3. Test results showing the performance of the LLWNN for modeling the NIFTY index

V. CONCLUSIONS

In this paper, we have demonstrated how the chaotic behavior of stock indices could be well represented by local linear wavelet neural networks. Empirical results on the two data sets using LLWNN models clearly reveal the efficiency of the proposed techniques. In terms of *RMSE* values, for the Nasdaq-100 index and the NIFTY index, LLWNN performed marginally better than other models. For both index (test data), LLWNN 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 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. Our experiment 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.

ACKNOWLEDGMENT

This research was partially supported by the National High Technology Development Program of China (863 Program) under contract number 2002AA4Z3240, and The Provincial Science and Technology Development Program of Shandong under contract number SDSP2004-0720-03.

REFERENCES

- [1] Abraham A., Nath B. and Mahanti P.K., "Hybrid Intelligent Systems for Stock Market Analysis", *Computational Science*, Springer-Verlag Germany, Vassil N Alexandrov et al (Editors), USA, (2001)337-345.
- [2] Abraham A., Philip N.S., and Saratchandran P., "Modeling Chaotic Behavior of Stock Indices Using Intelligent Paradigms", *International Journal of Neural, Parallel and Scientific Computations*, USA, Volume 11, Issue (1,2), (2003)143-160.
- [3] Leigh W., Modani N., Purvis R. and Roberts T., "Stock market trading rule discovery using technical charting heuristics", *Expert Systems with Applications* 23(2), (2002)155-159.
- [4] Leigh W., Purvis R. and Ragusa J.M., "Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support", *Decision Support Systems* 32(4), (2002)361-377.
- [5] Nasdaq Stock MarketsM, <http://www.nasdaq.com>
- [6] National Stock Exchange of India Limited, <http://www.nse-india.com>
- [7] Chen Y. and A. Abraham, "Hybrid Learning Methods for Stock Index

- Modeling", *Book chapter, Artificial Neural Networks in Finance, Health and Manufacturing: Potential and Challenges*, J. Kamruzzaman, R. K. Begg and R. A. Sarker (Eds.), Idea Group Inc. Publishers, USA, 2005. (In Press)
- [8] Chen, Y., Yang, B., Dong, J., "Time-Series Prediction using a Local Linear Wavelet Neural Network", *Neurocomputing*, 2005 (In press).
- [9] Chen, Y., Yang, B., Dong, J., Abraham A., "Time-series forecasting using flexible neural tree model", *Information Science*, Vol.174, Issues 3/4, pp.219-235, 2005.
- [10] Moore A. and Schneider J. and Deng K., "Efficient Locally Weighted Polynomial Regression Predictions", *Proceedings of the Fourteenth International Conference on Machine Learning*, (1997)236-244.
- [11] Bosman, P. A. N. and Thierens, D. (2000), "Expanding from Discrete to Continuous EDAs: The IEDA", *Proceedings of Parallel Problem Solving from Nature*, PPSN-VI, pp. 767-776
- [12] Muehlenbein, H. and Paa⁻; G. (1996), "From Recombination of Genes to the Estimation of Distribution", Part 1, Binary Parameter, *Lecture Notes in Computer Science 1141*, Parallel Problem Solving from Nature, pp. 178-187.
- [13] Rudlof, S. and Koppen M., "Stochastic Hill-Climbing with Learning by Vectors of Normal Distributions", Nagoya, Japan, 1996.
- [14] Larraenaga, P. and Lozano, J. A. (2001), "Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation", *Kluwer Academic Publishers*.
- [15] S. Tsutsui, M. Pelikan, and D. E. Goldberg, "Evolutionary Algorithm Using Marginal Histogram Models in Continuous Domain", *In Proceedings of the 2001 Genetic and Evolutionary Computation Conference Workshop*, 230-233, 2001.
- [16] Pelikan, M. and Goldberg, D. E. and Cantu-Paz, E. (1999), "BOA: The Bayesian optimization algorithm", *In Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, W. Banzhaf and J. Daida and A. E. Eiben and M. H. Garzon and V. Honavar and M. Jakiela and R. E. Smith (Eds), Morgan Kaufmann Publishers, San Francisco, CA, Orlando, FL, vol. 1, pp. 525-532.
- [17] Zhang, B. T. (1999), "A Bayesian Framework for Evolutionary Computation", *Proceedings of the 1999 Congress on Evolutionary Computation*, vol.1, pp. 722-228.
- [18] Foss, B., T.A. Johansen, On local and fuzzy modeling. *Proc. of 3rd Int. Industrial Fuzzy Control and Intelligent Systems*, 134-139, 1993.
- [19] Fischer, B., O. Nelles and R. Isermann, "Adaptive predictive control of a heat exchanger based on a fuzzy model", *Control Engineering Practice*, 6: 259-269, 1998.