Professor Heng-Li YANG
E-mail: yanh@nccu.edu.tw
Han-Chou LIN, PhD Student
E-mail: actuallife@hotmail.com
Department of Management Information Systems
National Cheng-Chi University, Taiwan

INTEGRATING EMD, CHAOS-BASED NEURAL NETWORK AND PSO FOR FINANCIAL TIME SERIES FORECASTING

Abstract: *In capital market research, stock or index prices are notoriously difficult to predict, because of their chaotic nature. For chaotic time series, the prediction techniques of PSR (Phase Space Reconstruction) methods, which are based on attractor reconstruction, can be employed to extract the information and characteristics hidden of the dynamic system from the time series. However, the existence of noise which may mask or mimic the deterministic structure of the time series, can lead to spurious results. In this work, EMD (Empirical Mode Decomposition) is specially developed for analyzing such nonlinear and non-stationary data. Thus, the major of this study is to integrate PSR, EMD and NN techniques optimized by particle swarm optimization to attempts to increase the accuracy for the prediction of stock index. The effectiveness of the methodology was verified by experiments comparing random walk model for Nasdaq Composite Index (NASDAQ). The results show that the proposed PSR-EMD-NNPSO model provides best prediction of stock index.*

Keywords: *Phase space reconstruction, Empirical mode decomposition, Neural network, Particle Swarm Optimization, Financial time series forecasting.*

JEL Classification Code: C32, C45, C53

Introduction

In the business and economic environment, it is very important to accurately predict various kinds of financial variables. In the past, most prediction models were based on conventional statistical methods, such as random walk, exponential smoothing, regression analysis, multivariate analysis and so on. It has been indicated that conventional statistical techniques for forecasting have reached their limitations in applications with nonlinearities in the dataset (Refenes, et al., 1994). Artificial neural network, a computing system containing many simple nonlinear computing units or nodes interconnected by links, is a well-tested method for financial analysis on the stock market (Refenes, et al., 1997). Neural networks have been shown to be able to decode nonlinear time series data which adequately describe the characteristics of the stock markets (Lapedes and Farber, 1987). In past decades, neural networks have been explored by many researchers for financial forecasting (Ecer, 2013). Among these researches, some are conducted particularly on forecasting the value of a stock index (Shavandi and Alizadeh, 2010). Using neural networks for modeling and predicting stock market has been the subject of recent empirical and theoretical investigations by both academics and practitioners.

In capital market research, stock or index prices are notoriously difficult to predict using the traditional forecasting methods, such as least squares regression, because of their chaotic nature. Chaos theory points out that the complicated things which are originally believed unpredictable have predictability (Lu and Lu, 2002). Moreover, chaos is widely found in the fields of physics, other natural sciences and empirical evidence of chaotic behavior in financial time series can also be found (McKenzie, 2001). For chaotic time series, the techniques of prediction based on phase space reconstruction (PSR) can be employed to extract the information and characteristics hidden of the dynamic system from the time series. PSR is an embedding that maps a signal into a sufficiently high dimension. In the new high dimensional space, a structure is formed that is topologically equivalent to the original phase space. This arouses encouragement to apply the chaos theory to the time series forecast.

The vast majority of PSR methods are based on attractor reconstruction from time series and such characteristics as largest Lyapunov exponent, K2 entropy, and correlation dimension calculation (Kantz and Schreiber, 2004). However, there are noise, which may mask or mimic the deterministic structure of the time series, can lead to spurious results (Zhang, Luo, and Small, 2006). In this work, empirical mode decomposition (EMD), which is a new technique in dealing with noise and nonlinearity in stock index prediction, is applied to decompose the stock index time series. The EMD technique is specially developed for analyzing nonlinear and non-stationary data (Huang et al., 1998). By the EMD technique, any complicated data set can be decomposed into a finite and often small number of intrinsic mode functions (IMFs). This decomposition method is adaptive, and, therefore, highly efficient. Since the decomposition is based on the local characteristic time scale of the data, it is applicable to non-linear and non-stationary processes. Thus, the major objective of this study is to integrate PSR, EMD and NN technique optimized by particle swarm optimization to attempts to increase the accuracy for the prediction of stock index.

1. Methodology

2.1. Phase space reconstruction (PSR)

The idea of the chaotic dynamics reconstruction technique stems from the embedding theorem developed by Takens (1981) and Sauer, et al. (1991). The theorem regards a one-dimensional chaotic time series as compressed information from a higher dimension space. According to Taken's theorem, we tried to reconstruct an attractor which preserves the invariant characteristics of the original attractor, since the original attractor is unknown. The dynamics of the time series x_1, x_2, …, x_n is fully captured or embedded in the m-dimensional phase space, $m \geq d$ where d is the dimension of the original attractor. A vector Y_t in the reconstructed phase space is constructed from the time series as follows：

$$Y_t = [\, x_t \,, x_{t-\tau} \,,\ldots, x_{t-(m-1)\tau} \,] \tag{1}$$

where τ is the delay time.

The above approach to reconstruct the attractor is based on Taken's theorem which guarantees that for an infinite, noise free, data series, one can almost always find an embedding dimension m preserving the invariant measures of the chaotic attractor of dimension d (Takens, 1981).

$$m \geq 2d + 1. \tag{2}$$

Takens considered that the sufficient condition for the embedding dimension is $m \geq 2d + 1$. However, a too large embedding dimension needs more observations and complex computation. Moreover, if we choose a too large embedding dimension, noise and other unwanted inputs will be highly embedded with the real source input information, which may corrupt the underlying system dynamic information. Therefore, in accordance with Sauer, et al. (1991), if the dimension of the original attractor is *d* then an embedding dimension of $m = 2d + 1$ will be adequate for reconstructing the attractor.

The delay time τ is an important parameter for controlling the accuracy of time-series prediction. Some of the methods for determining the optimal delay time τ include auto correlation (AC) (Broomhead, 1986), average displacement (AD) (Rosenstein, et al., 1994), and mutual information (Fraser and Swinney, 1986). Each method has its own advantages and drawbacks. Kim, et al. have developed a special technique called the C-C method, which can estimate both the delay time τ and embedding window well. This method is relatively easy to implement with good performance and, therefore, is discussed in this paper.

Since *d* in Eq. (2) is an internal characteristic of the system and usually unknown in real applications, its estimation, namely correlation dimension D_m, is usually calculated and used to determine *m* in place of *D*. The G-P algorithm (Grassberger and Procaccia, 1983) is one of the popular methods for determining D_m. In this method, a correlation integral ln $C_m(r)$ is defined as

$$\mathrm{C}_m(r) = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N} H(r-|x_i - x_j|) \tag{3}$$

where H is the Heaviside function and *m* is the embedding dimension. For *r* sufficiently small and the number of observed values N sufficiently large, then

$$D_m = \lim_{r\to 0}\frac{\ln C_m(r)}{\ln(r)} \tag{4}$$

So the G-P algorithm first plots $\ln C_m(r)$ against $\ln(r)$ curves by increasing the values of *m* until the slope of the curve’s linear part is almost constant, then estimates the slope of the straight portion of the curves with the similar or same linear parts over a certain range of *r*, and further regards the estimated slope as the correlation dimension D_m.

When considering chaos, tools based on PSR have been developed. The most commonly used of these chaos tests are the Lyapunov exponents test and the Grassberger—Procaccia correlation dimension test (Harrison, et al., 1999). Here we concentrate on the latter. For a chaotic attractor, D_m is a non-integer, the value of which determines whether the system is low- or high-dimensional.

1.2. *Empirical mode decomposition (EMD)*

Empirical mode decomposition method based on Hilbert-Huang transformation is developed from the simple assumption that any signal consists of different simple intrinsic mode oscillations. The essence of the method is to identify the intrinsic oscillatory modes (IMFs) (Huang, et al., 1998) by their characteristic time scales in the signal and then decompose the signal accordingly. The characteristics time scale is defined by the time lapse between the successive extremes.

To extract the IMF from a given data set, the sifting process is implemented as follows. First, identify all the local extrema, and then connect all of the local maxima by a cubic spline line as the upper envelope. Then, repeat the procedure for the local minima to produce the lower envelope. The upper and lower envelope. The upper and lower envelopes should cover all the data between them. Their mean is designated $m_1(t)$, and the difference between the data and $m_1(t)$ is $h(t)$, i.e.:

$$x(t) - m_1(t) = h_1(t) \tag{5}$$

Ideally, $h_1(t)$ should be an IMF, for the construction of $h_1(t)$ described above should have forced the result to satisfy all the definitions of and IMF, we demand the following conditions: (i) $h_1(t)$ should be free of riding waves, i.e., the first component should not display under-shots or over-shots riding on the data and producing local extremes without zero crossing. (ii) To display symmetry of the upper and lower envelops with respect to zero. (iii) Obviously the number of zero crossing and extremes should be the same in both functions.

The sifting process has to be repeated as many times as it is required to reduce the extracted signal to an IMF. In the subsequent sifting process steps, h_1(t) is treated as the data; then:

$$h_1(t) - m_{11}(t) = h_{11}(t) \tag{6}$$

Where $m_{11}(t)$ is the mean of the upper and lower envelops of $h_s(t)$. This process can be repeated up to k times; $h_{1k}(t)$ is then given by

$$h_{1(k-1)}(t) - m_{1k}(t) = h_{1k}(t) \tag{7}$$

After each processing step, checking must be done on whether the number of zero crossings equals the number of extrema. The resulting time series is the first IMF, and then it is designated as $c_1(t) = h_{1k}(t)$. The first IMF component from the data contains the highest oscillation frequencies found in the original data $x(t)$.

This first IMF is subtracted from the original data, and this difference, is called a residue $r_1(t)$ by:

$$x(t) - c_1(t) = r_1(t) \tag{8}$$

The residue $r_1(t)$ is taken as if it was the original data and we apply the sifting process to it again. The process of finding more intrinsic modes $c_1(t)$ continues until the last mode is found. The final residue will be a constant or a monotonic function; in this last case it will be the general trend of the data.

$$x(t) = \sum_{j=1}^{n} c_j(t) + r_n(t) \tag{9}$$

Thus, one achieves a decomposition of the data into n-empirical IMF modes, plus a residue, $r_n(t)$, which can be either the mean trend or a constant.

2.3. Artificial Neural Network (ANN)

An ANN is a biologically inspired form of distributed computation. It simulates the functions of the biological nervous system by a composition of interconnected simple elements (artificial neurons) operating in parallel. An element is a simple structure that performs three basic functions: input, processing and output. ANNs can be organized into several different connection topologies and learning algorithms (Lippmann, 1987). The number of inputs to the network is constrained by the problem type, whereas the number of neurons in the output layer is constrained by the number of outputs required by the problem type. Moreover, the number of hidden layers and the sizes of the layers are decided by the designer.

ANNs apply many learning rules, of which BP is one of the most commonly used algorithms in financial research. BP trains multilayer feed-forward networks with differentiable transfer functions to perform function approximation, pattern association and pattern classification. It is the process by which the derivatives of network error, with respect to network weights and biases, are computed to perform computations backwards through the network. Computations are derived using the chain rule of calculus. There are several different BP training algorithms with a variety of different computation and storage requirements. No single algorithm is best suited to all the problems. All the algorithms use the gradient of the performance function to determine how to adjust the weights to minimize the performance. For example, the performance function of feed forward networks is the Mean Square Error (MSE).

The basic BP algorithm adjusts the weights in the steepest descent direction (negative of the gradient); that is, the direction in which the performance function decreases most rapidly. The training process requires a set of examples of proper network inputs and target outputs. During the training, the weights and biases of the network are iteratively adjusted to minimize the network performance function.

2.4. Particle swarm optimization (PSO)

In PSO (Parsopoulos and Vrahatis, 2002) the population is called a swarm, and each individual of the swarm is called a particle. In each iteration step j of the optimization, each particle q yields an updated vector X_q^{j+1} using its best personal vector P_q^j so far and the best global vector P_g^j among all individuals of the population. For the update of X_q^j, firstly a velocity vector V_q^j is updated by using two vector differences, and secondly the velocity vector is added to X_q^j for the calculation of X_q^{j+1} by the following rules:

$$V_q^{j+1} = \omega V_q^j + c_1 r_1 (P_q^j - X_q^j) + c_2 r_2 (P_g^j - X_q^j)\ , \quad X_q^{j+1} = X_q^j + V_q^{j+1} \tag{10}$$

where X_q^j is the vector of the *q*th particle at the *j*th iteration step. In Eq. (10), r_1 and r_2 are random numbers of the interval (0,1), and the other values ω, c_1 and c_2 are real-valued parameters of the PSO. In addition to this standard PSO, a random change of the components in each iteration step is introduced to help avoid suboptimal local minima in the search space. For each component class of X_q^{j+1}, i.e., for the component class of the epoch.

2. Forecasting Model

3.1 Establish the training sets

Phase space reconstruction method provides a way to characterize the chaotic system. Reconstructed data Y*t* Eq. (1) can be used to train the predictor model. The *l*-lead time prediction, *x*(*t*+*l*), is a function of lag vector Y*t* in the reconstructed space. The function can be expressed as follows:

$$x(t+l) = \mathrm{F(Y)} \tag{11}$$

Recently, some forecasting techniques for chaotic time series almost fixed their selected time lag at 1(Liong and Sivapragasam, 2002), thus we also fix time lag = 1 in this study. A vector of m embedding dimension is put into the model, the data are predicted, in the function

$$\hat{x}(t+1) = f(x(t), x(t-\tau), \ldots, x(t-(m-1)\cdot\tau)) \tag{12}$$

where $\hat{x}(t+1)$ stands for the value predicted, and $x(t), x(t-\tau),..., x(t-(m-1)\cdot\tau)$ stand for input.

3.2 Optimization

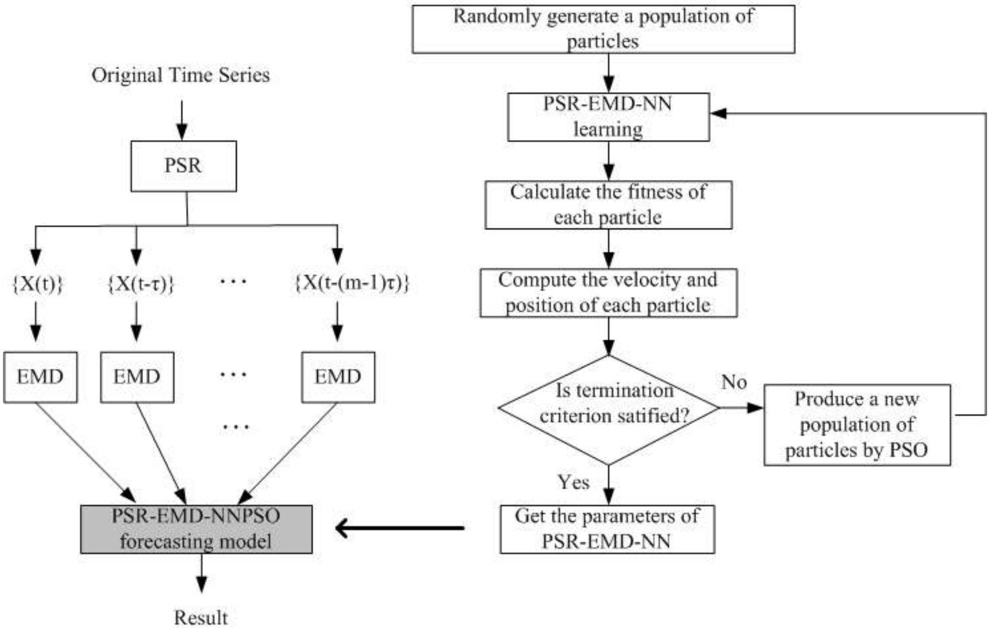

Randomly generate a population of particles
Original Time Series
PSR-EMD-NN learning
PSR
Calculate the fitness of each particle
{X(t)}
{X(t-τ)}
···
{X(t-(m-1)τ)}
Compute the velocity and position of each particle
EMD
EMD
···
EMD
Is termination criterion satified?
No
Produce a new population of particles by PSO
···
Yes
PSR-EMD-NNPSO forecasting model
Get the parameters of PSR-EMD-NN
Result

Figure 1. Schematic representation of the proposed PSR-EMD-NNPSO model

Consider a great influence of the parameters on generalization performance of NN, particle swarm optimization is applied to search the parameters of NN. The procedure of the PSR-EMD-NNPSO model is interpreted in Fig. 1. The optimal process by particle swarm optimization is described as follows:

Step1. Initialize parameters of particle swarm optimization including the number of evolutionary generation, population size, inertia weight, and randomly generate a population of particles composed of epoch.

Step2. Use the selected parameters to train a NN model. The testing samples are used to measure forecasting ability of the BPNN model. Applicability of the model is measured by fitness as mean absolute percentage error (MAPE).

Step3. Compute the velocity and position of each particle with Eq.(10), respectively.

Step4. Stop the algorithm if termination criterion is satisfied, and the best NN is gained. Otherwise, produce the new particle according to Eq.(10).

3. Experimental Results

4.1 Research data and evaluation criteria

In an experiment, we evaluated our proposed model by the *Nasdaq Composite Index (NASDAQ)*. Since closing prices have been used in many previous studies, in the same way, the empirical data used are also the daily closing prices of the stock index, which are extracted from the TEJ (Taiwan Economic Journal) Database (January 1997 – August 2008). In addition, the data are divided into training and testing sets. Many studies have applied a convenient ratio to separate training (in-sample) from testing (out-of-sample) data between the ratios 7：3 and 9：1 (Zhang, 2004). This study follows the choice: the data from Jan. 1997 to Apr. 2006 are used for training, while the data from May 2006 to Aug. 2008 are used for testing. Hence, the ratio adopted is 8：2, which is a ratio that lies in-between.

Each data point was scaled by Eq. (13) within the range of (0,1). This scaling for original data points helps to improve the forecasting accuracy (Chang and Lin, 2001):

$$\frac{X_t - X_{\min}}{X_{\max} - X_{\min}} 0.7 + 0.15 \qquad (13)$$

where x_t is the exchange rate at time t, $x_{\max}$ is the maximum of exchange rates during the period of data source and $x_{\max}$ is the minimum of exchange rates during the period of data source.

The prediction performance is evaluated using the following statistics: MSE (mean-squared error) 、RMSE (root-mean-square error) 、MAPE (mean-absolute percentage error)、MAE (mean-absolute error) and DS (Directional Symmetry). These formulas are shown in Table 1. The former four criteria measure the correctness of a prediction in terms of levels and the deviation between the actual and predicted values. The smaller the values, the closer the predicted time-series values will be to the actual values. Although predicting the levels of price changes (or first differences) is desirable, in many cases the sign of the change is equally important. Most investment analysts are usually far more accurate at predicting directional changes in an asset price than predicting the actual level. DS provides an indication of the correctness of the predicted direction given in the form of percentages (a large value suggesting a better predictor).

Table 1. Performance criteria and formulas

MSE = $\frac{1}{N}\sum\left\lvert x_t - \hat{x}_t\right\rvert^2$	RMSE = $\sqrt{\frac{1}{N}\sum\left\lvert x_t - \hat{x}_t\right\rvert^2}$	MAE = $\frac{1}{N}\sum_{t=1}^{N}\left\lvert x_t - \hat{x}_t\right\rvert$
MAPE = $\frac{\sum\left\lvert\frac{x_t - \hat{x}_t}{x_t}\right\rvert}{N}\times 100$	DS = $\frac{100}{N}\sum_{t=1}^{N} d_t$,	$d_t = \begin{cases} 1 & (x_t - x_{t-1})(\hat{x}_t - \hat{x}_{t-1}) \geq 0 \\ 0 & otherwise \end{cases}$

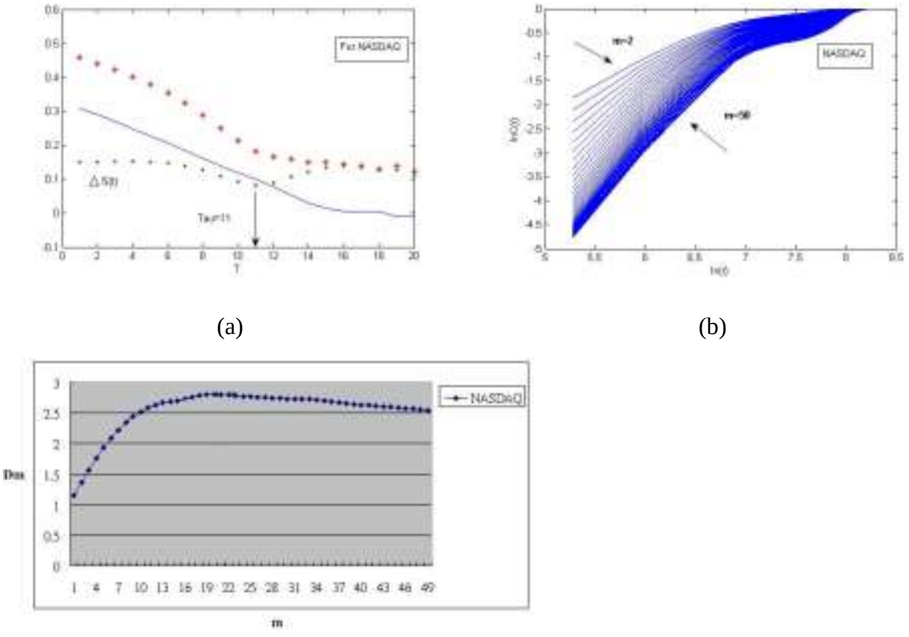
Note: x_t denotes the real stock index on the "t"th day, $\hat{x}_t$ represents the predicted stock index, and N is the number of days.

In order to compare the forecasting capability of the proposed PSR-EMD-NNPSO methodology with traditional approaches, a random walk model is used as the benchmark model.

4.2 Implementation of PSR

We apply reconstruction phase space in time series of stock index, $\{x_i\}$. C-C method was used to determine optimal delay time τ in this study, and the results, the first local minimum of $\Delta\bar{S}(t)$, were illustrated in Fig. 2.(a) The value of the first local minimum $\Delta\bar{S}(t)$ on x-axis for NASDAQ is 11, which is the first locally optimal delay time τ for independence of the data (Kim, et al., 1999).

After the optimal delay time τ was found, the correlation integrals and the exponents for the stock index series are computed using the Grassberger–Procaccia algorithm for phase-spaces reconstructed with embedding dimensions from 2 to 50. Fig. 2.(b) and (c) present the results of the correlation integral analysis. The relationship between C(r) and r, shown in Fig. 2.(b), indicates clear scaling regions (between log r=5.5 and 7.0) for all the embedding dimensions used, allowing fairly accurate estimates of the correlation exponents, m, which are presented in Fig. 2.(c) against the corresponding m values. Fig. 2.(c) shows an increase in the correlation exponent with the embedding dimension up to a certain point, and saturation beyond this point. Such saturation may be an indication of the deterministic dynamics in the runoff phenomenon. The saturation value of the correlation exponent (or correlation dimension) is about 2.806, suggesting that the number of variables dominantly influencing the runoff dynamics (or the minimum number of phase-spaces required) is 3.



(a) (b)

(c)

Figure 2. (a) Optimal delay time τ with C-C method. (b) Optimal embedding dimension with G-P method. (c) Correlation dimension

4.3 Data preprocessing using EMD

After one-dimensional time series of the NASDAQ index was reconstructed by PSR, 3 time series served as inputs for the predictor model would be determined. Before feeding those input data into predictor model for forecasting, we pre-processed each input data using EMD.

Take first input data $\{x(t)\}$ for example, the analytic process as shown in the following (Huang et al., 2003): the data set shown in the sub-figure “Original” in Fig. 3, after subjected to the EMD, yields eleven IMF components shown in Fig. 7. The decomposition identifies eleven modes: Mode c_{11} is the residue, mode c_1 contains the highest signal frequencies and mode c_2 the next higher frequency band and so on. Here we can immediately observe many interesting features of the data from just the IMF components. To begin with, there is an obvious change in the data quality starting around 800, when the amplitudes of the short period IMF components (i.e. c_1 and c_2) suddenly increase. And then, our major work for data preprocessing is to reconstitute the data from the components (reconstruction procedure) which illustrates the use of the EMD process as a filter. The sequence of steps is shown as follows in Fig. 3: In each of the sub-panels, we plot the data as a dotted line and the partial sum of the IMFs as a solid line. In Fig. 4(a), we plot the data and component c_{11}, the residue of the sifting. If we add the oscillatory component c_{10}, we have the result in Fig. 4(b). When adding the component c_7, in the result shown in Fig. 4(e), this smooth line clearly gives the smoothest trend of the data variation. With step by step adding of the IMF components, we finally arrived at the sum of all the IMF components shown in Fig. 4(k). Due to the $c_1$and c_2 belong to high frequency components, we eliminated the two components from this data set and adopted Fig. 4(i) as the final result of data preprocessing.

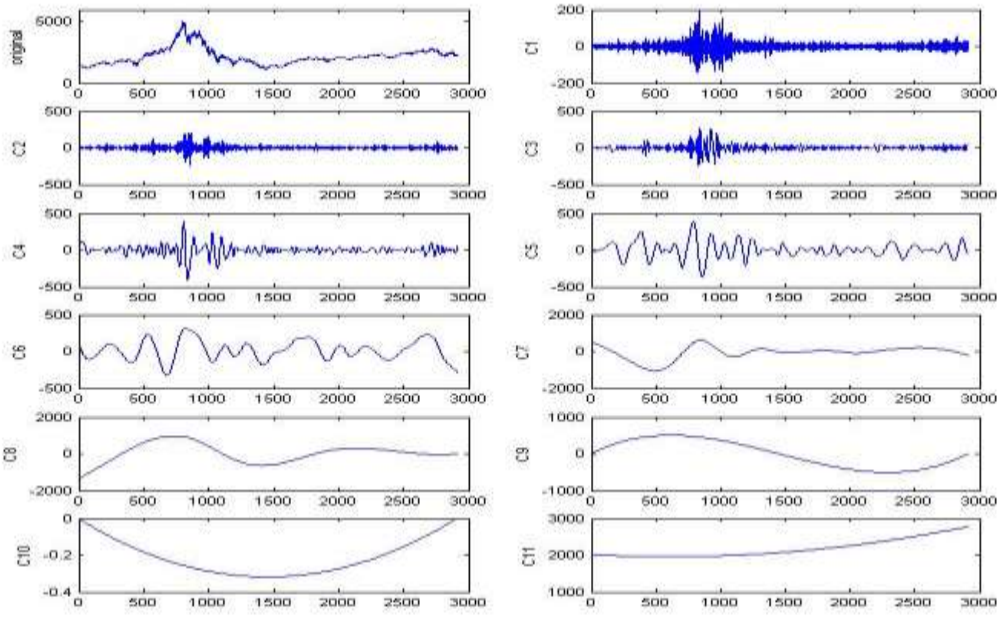


Figure 3. The IMF for the first data set x(t) of NASDAQ index through the EMD

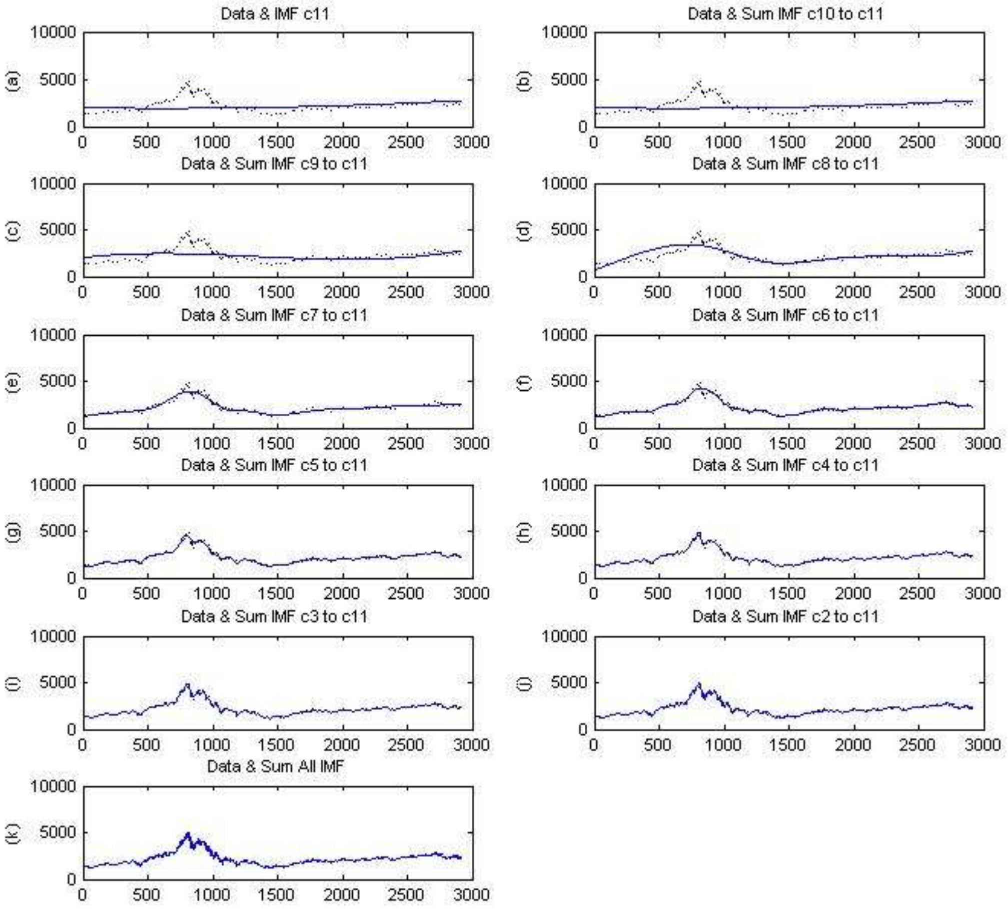


Figure 4. Re-construction of the data from the IMF components. This is also a demonstration for using the EMD technique as a filter.

4.4 Implementation of BPNN optimized by PSO

In this paper, standard three-layer BPNNs are used as benchmarks. The BPNN technique in this experiment is implemented using the Matlab 7.3's ANN toolbox. Several learning techniques, such as the quasi-Newton method, Levenberg-Marquardt algorithm and conjugate gradient methods could also be used. For efficiency, however, we use the Levenberg–Marquardt algorithm. This study follows Kim (2003) to choose the number of nodes (n/2, n, 2n) in the hidden layer and stopping criteria for training. 2, 3, 6 hidden nodes are employed for each stopping criteria and the epochs are optimized by PSO (http://psotoolbox.sourceforge.net/). The learning rate is 0.001(default). The activation function of the hidden layer is sigmoid and the output node uses the linear transfer function. This study allows 3 input nodes because 3 input variables (i.e. three embedding dimensions). Table 2. shows the results based on learning epochs optimized by PSO for specific hidden nodes. 2 hidden nodes and 114 learning epochs are optimal, and the testing MAPE is 0.008506.

Table 2. PSR-EMD-NNPSO model selection result

Hidden nodes	**Epochs optimized by PSO**	**Testing MAPE(%)**
2	114	0.8505
3	129	0.8533
6	274	0.8510

Table 3. Comparison of the forecasting results from the two models

Model	**MSE**	**RMSE**	**MAE**	**MAPE(%)**	**DS**
PSR-EMD-NNPSO	686.2062	26.1955	20.5017	0.8506	0.6000
Random Walk	809.2262	28.4469	21.4588	0.8880	0.4983

4.4 Performance criteria and results

The forecasting results of the PSR-EMD-NNPSO and Random Walk models for the testing data are collated in Table 3. From Table 3, it can be found that the MSE, RMSE, MAE and MAPE of the PSR-EMD-NNPSO model are, 686.206, 26.1955, 19.68 and 0.8506, respectively. It can be observed that these values are smaller than those of the Random Walk model. It indicates that there is a smaller deviation between the actual and predicated values using the proposed PSR-EMD-NNPSO model. Moreover, the PSR-EMD-NNPSO model has higher DS which is 0.6000. DS provides a good measure of the consistency in prediction of the index direction. Thus, it can be concluded that the proposed PSR-EMD-NNPSO model provides a better forecasting result than the Random Walk model in terms of prediction error and prediction accuracy.

4. Conclusions

This study applies phase space reconstruction (PSR) method and empirical mode decomposition (EMD) to time series of the NASDAQ index and forecasts stock index with NN approach optimized by particle swarm optimization (PSO), namely PSR-EMD-NNPSO. Wes have evaluated the feasibility of the proposed model compared to the Random Walk model. In this paper, the daily closing prices of NASDAQ COMPOSITE stock indices extracted from the TEJ (Taiwan Economic Journal) Database (January 1997 – August 2008) are used as experimental data. Moreover, G-P correlation dimension method is also used for chaos test, for a chaotic attractor, correlation dimension D_m is a non-integer (equal to 2.806 from the time series of the stock indices).

We evaluate performance of the three models by MSE、RMSE、MAPE、MAE and DS. It is clear from empirical results, the proposed model PSR-EMD-NNPSO gives better accuracy not only in the derivation performance, but also in the direction performance. Out experiments also show that the AI technique can assist the stock market trading and the development of the financial decision support systems.

Future research may apply other AI methods and data analytic techniques to the forecasting of Nasdaq Composite Index. Direction prediction criteria are important signals in the trading strategies of investors. In the current models, we only selected the daily closing prices of the stock index as input variables. Future research may include other efficient input variables (such as some macroeconomic variables) and try to use diverse data for testing feasibility.

REFERENCES

[1] **Broomhead, D. S. (1986),** ***Extracting Qualitative Dynamics from Experimental Data***. *Physica D, 20*, 217;

[2] **Chang, C. C., & Lin, C. J. (2001), LIBSVM:** ***A Library for Support Vector Machines***. Retrieved from http://www.csie.ntu.edu.tw/cjlin/papers/libsvm.pdf website:

[3] **Ecer, Fatih (2013),** ***Artificial Neural Networks in Predicting Financial Performance: An Application for Turkey's Top 500 Companies***. *Economic Computation and Economic Cybernetics Studies and Research*, 47(2), *ASE Publishing;* 103-114;

[4] **Fraser, M. & Swinney, H. L. (1986),** ***Independent Coordinates for Strange Attractors from Mutual Information***. *Physical Review A, 33,* 1134;

[5] **Grassberger, P. and Procaccia, I. (1983),** ***Measuring the Strangeness of Strange Attractors***. *Physica D*, 9, 189-208;

[6] **Harrison, R. G., Yu, D., Oxley, L., Lu, W. & George, D. (1999),** ***Non-linear Noise Reduction and Detecting Chaos: Some Evidence from the S&P Composite Price Index***. *Mathematics and Computers in Simulation, 48,* 497-502;

[7] **Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., . . Liu, H. H. (1998),** ***The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-stationary Time Series Analysis***. Paper presented at the Proceeding of Royal Society London, Series A;

[8] **Huang, N. E., Wu, M.-L., Qu, W., Long, S. R., Shen, S. S. P., & Zhang, J. E. (2003),** ***Applications of Hilbert-Huang Transform to Non-stationary Financial Time Series Analysis***. *Applied Stochastic Models in Business and Industry, 19*, 245-268;

[9] **Kantz, H. & Schreiber, T. (2004),** ***Nonlinear Time Series Analysis*****.** Cambridge*: Cambridge University Press*;

[10] **Kim, H. S., Eykholt, R. & Salas, J. D. (1999),** ***Nonlinear Dynamics, Delay Times and Embedding Windows***. *Physica D, 127*, 48-60;

[11] **Kim, K.-J. (2003),** ***Financial Time Series Forecasting Using Support Vector Machines***. *Neurocomputing, 55*(1-2), 307-319. doi: 10.1016/s0925-2312(03)00372-2;

[12] **Lapedes, A. & Farber, R. (1987),** ***Nonlinear Signal Processing Using Neural Networks: Prediction and System Modelling***;

[13] **Liong, S. Y. & Sivapragasam, C. (2002),** ***Flood Stage Forecasting with SVM.*** *Journal of the American Water Resources Association, 38*(1), 173-186;

[14] **Lippmann, R. P. (1987)*, An Introduction to Computing with Neural Nets*.** *IEEE ASSP Magazine, 4*(2), 4-22;

[15] **Lu, J. H. & Lu, J. A. (2002),** ***Chaos Time Series Analysis and its Applications*:** Wuhan;

[16] **McKenzie, M. D. (2001),** ***Chaotic Behavior in National Stock Market Indices - New Evidence from the Close Returns Test***; *Global Finance Journal, 12*, 35-53;

[17] **Parsopoulos, K. E. & Vrahatis, M. N. (2002),** ***Particle Swarm Optimization Method for Constrained Optimization Problems.*** Paper presented at the Proceedings of the 2002 ACM Symposium on Applied Computing;

[18] **Refenes, A. N., Zapranis, A. & Francis, G. (1994),** ***Stock Performance Modeling Using Neural Networks: A Comparative Study with Regression Models***. *Neural Networks, 7*(2), 375-388. doi: http://dx.doi.org/10.1016/0893-6080(94)90030-2;

[19] **Refenes, A. P. N., Burgess, A. N. & Bentz, Y. (1997*), Neural Networks in Financial Engineering: A Study in Methodology*.** *IEEE Transactions on Neural Networks, 8*(6), 1222-1267;

[20] **Rosenstein, M. T., Collins, J. J. & Luca, C. J. D. (1994),** ***Reconstruction Expansion as a Geometry-based Framework for Choosing Proper Delay Time***. *Physica D, 73*, 82-89;

[21] **Sauer, T., Yorke, J. A. & Casdagli, M. (1991),** ***Embedology***. *Journal of Statistical Physics, 65*(3/4), 579-616;

[22] **Shavandi, H. & Alizadeh, P. (2010*), A Hybrid Intelligent Model Using Technical and Fundamental Analysis to Forecasting Stock Price Index*.** *Economic Computation and Economic Cybernetics Studies and Research, ASE Publishing; 44*(3), 95-112;

[23] **Takens, F. (1981),** ***Detecting Strange Attractors in Turbulence***. *Dynamical systems and turbulence., 898*, 366-381;

[24] **Zhang, G. P. (2004),** ***Business Forecasting with Artificial Neural Networks: An Overview, Neural Networks in Business Forecasting***. USA: *Idea Group Publishing;*

[25] **Zhang, J., Luo, X. & Small, M. (2006),** ***Detecting Chaos in Pseudoperiodic Time Series without Embedding***. *Physical Review E, 73*(1), 016216.