Prediction of S&P 500 and DJIA Stock Indices using Particle Swarm Optimization Technique

Ritanjali Majhi, G. Panda, *Senior Member IEEE,* G. Sahoo, Abhishek Panda and Arvind Choubey

Abstract— **The present paper introduces the Particle Swarm Optimization (PSO) technique to develop an efficient forecasting model for prediction of various stock indices. The connecting weights of the adaptive linear combiner based model are optimized by the PSO so that its mean square error(MSE) is minimized. The short and long term prediction performance of the model is evaluated with test data and the results obtained are compared with those obtained from the multilayer perceptron (MLP) based model. It is in general observed that the proposed model is computationally more efficient, prediction wise more accurate and takes less training time compared to the standard MLP based model.**

I. Introduction

AVILIABILITY of assets and money can assure a person a secured and comfortable life. With the introduction of online trading the stock market has become one of the forums where the small investors can also earn good profits. In the present scenario large number of transactions are done in stock market for selling and purchasing shares. Presently it is also a burning topic of research due to large applications in different business transactions. As the stock market data is non-stationary and volatile the investors feel insecure during investing. In the recent years lot of attention has been devoted to the analysis and prediction of future values and trends of the financial markets. However, the stock market is affected by many complex events such as business cycles, monetary policies, interest rates and political situations. Due to volatility and non-stationary characteristics involved in the stock market data it is difficult to build an accurate forecasting model. But even then different financial forecasting methods have been proposed in the literature each of which has its own merits and limitations.

Some prior research on stock market prediction particularly using hybrid techniques is reviewed. Initial research work essentially on this topic was based on statistical approach such as regression, correlation and spectral analysis [1,2,3]. During the same time econometric models were also developed [4,5]. For short term forecasting, the regression and econometric models have been reported to be inefficient. However, the correlation and the spectral based models were successful but led to poor prediction quality due to nonrealistic assumption of financial time series such as strong autocorrelation, stationary characteristics and linear structure [6]. Subsequently soft computing tools such as artificial neural networks (ANN) and Fuzzy logic (FL) were employed to develop stock forecasting models. In the ANN based approach the models reported are based on multilayer perception (MLP)[7], recurrent neural network (RNN)[8], radial basis function (RBF)[9] and dynamic recurrent neural net (DRNN)[10]. These adaptive models have distinct advantages of fast universal approximation capabilities and adaptation of their prediction quickly to changed conditions. They also produce robust performance when the training data set is large or missing in between. However these models exhibit inconsistent performance on noisy financial data. In addition, the learning algorithms of these models are derivative based such as back propagation (BP) and recursive least square (RLS) algorithms. Subsequently many research papers appeared in the literature using evolutionary computing tools such as genetic algorithm(GA) and genetic programming(GP) in developing improved forecasting models. However the recent research trend in financial forecasting is to develop hybridized models using soft computing structure with evolutionary computing based learning techniques.

Prediction of stock market data are different from other benchmark time series prediction such as sun spot data or the time sequence generated from the Mackey-Glass equation. The stock data are highly time-invariant and are highly influenced by indeterminate dealing. There have been several applications of GA to portfolio optimization, bankruptcy prediction, financial forecasting, fraud detection and scheduling. Hiemstra has applied GA to the tactical asset allocation [11]. In econometric estimation the GA based model has been proposed in [12]. Similarly GA has been used in developing a forecasting model for portfolio decisions [13]. In a recent paper [14] an efficient investment strategy in portfolio management using GA has been proposed. Genetic programming (GP) [15] also has been applied to find the technical trading rules in the foreign exchange market. Bhattacharya et al [16] have developed a GP based trading model for efficient prediction of exchange rates. In another work Chen et al. [17] used GP to derive option pricing formulas using real data from S&P 500 index options for training and testing purpose. To approximate the relationship between the price of a stock option, the terms of the option contract and the properties of the underlying stock price, GP has been applied by Chidambaran et al [18]. In another publication [19] the application of GP to the prediction of the price data in Japanese stock market has been proposed. Comparative experiments have been conducted using ANN model to exhibit the effectiveness of

Ritanjali Majhi is with Centre of Management Studies, NIT, Warangal, INDIA (email : ritanjalimajhi@gmail.com)
G. Panda is with Department of Electronics & Communication Engineering, National Institute of Technology, Rourkela 769008,India,phone-+91-9437048906(email : ganapati.panda@gmail.com)
G. Sahoo is with the Department of CSE, BIT, Mesra, Ranchi, India.
Abhishek Panda is with IIM, Bangalore.
A. Choubey is with Dept. of Electronics & Communication Engg., NIT Jamshedpur, India.

978-1-4244-1823-7/08/$25.00©2008 IEEE

the GP based approach. An integrated approach using Hidden Markov Model (HMM), ANN [20] and GA has also been reported for better forecasting of financial market behavior particularly for a number of securities in the IT sector.

The present paper is organized as follows. Section II deals with the basic principle of the particle swarm optimization (PSO). The application of PSO for stock market prediction is discussed in Section III. The formula used for technical indicators and algorithm for developing the PSO based forecasting model are discussed here. The results and discussion is described in Section IV. Finally conclusion is given in Section V.

II. BASICS OF PARTICLE SWARM OPTIMIZATION

The Particle swarm optimization(PSO)is an evolutionary computation technique developed by Kennedy and Eberhrt [21]. It is motivated from the simulation of social behavior. In evolutionary computational algorithms evolutionary operators to manipulate the individuals are used. But in PSO these individual are evolved by cooperation and competition among the individuals themselves through generations. Each individual in PSO flies in the search space with a velocity which is dynamically adjusted according to its own as well as its companions' flying experiences. Each individual is treated as a volume-less particle in a D-dimensional space. The ith particle is represented as $X_i = [x_{i1}, x_{i2}, ...x_{iD}]^T$. The position giving the best previous value is called the best previous position of any particle and is represented as $P_i = [p_{i1}, p_{i2}, ...p_{iD}]^T$. The rate of change of position for ith particle is represented as $V_i = [v_{i1}, v_{i2}, ...v_{iD}]^T$. The global best position at a generation is given by $P_g = [p_{g1}, p_{g2}, ..., p_{gD}]^T$. The velocity and position of the dth element of ith particle are changed according to (1) and (2) respectively.

$$v_{id} = v_{id} + C_1 * rand_1 * (p_{id} - x_{id}) + C_2 * rand_2 * (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where C_1 and C_2 are two positive constants where as $rand_1$ and $rand_2$ are two random functions in the range [0, 1]. The second part in (1) is the cognition part which represents the private thinking of the particle itself. The third part is the social part which represents the collaboration among the particles. The new velocity of a particle is calculated using (1) which is according to its previous velocity and the distances of current position from its own best experience(position) and the group's best experience. After that the particle flies towards a new position according to (2). The performance of each particle is evaluated based on a predefined fitness function which is problem dependent.

The PSO is similar to a genetic algorithm (GA) in that the system is initialized with a population of random solution. But unlike in GA, each potential solution in PSO is assigned a randomized velocity and the potential solutions, called particles are than flown through the problem space. Each particle keeps track of its coordinates in the problem space which are associated with the best solution(fitness) it has achieved so far and is denoted by P_i. The overall best location (global) obtained so far by the particles is also tracked by the optimizer and is represented by $P_g = [p_{g1}, p_{g2}, ..., p_{gD}]^T$.

The steps in implementing the PSO algorithms are:

(a) Initialize a population of particles with random positions and velocities on D-dimensions in the problem space (the problem consists of D variables)
(b) For each particle, evaluate the fitness function in D variables.
(c) Compare the fitness value of each particle with that obtained from the personal best value. If the current position, X_i provides better fitness value than provided by P_i, then it is made equal to the current value and the previous location is updated to the current location in D-dimensional space.
(d) Compare each fitness value with that given by overall best value, P_g of the population. If the current value is better than that offered by P_g then reset P_g to the current best position vector.
(e) Update the velocity and position of each particle according to (1) and (2) respectively.
(f) Loop to step (b) until a pre-specified criterion is met for example the minimum mean square error (MMSE) is achieved or all the particles attain a common best position.

The acceleration constants C_1 and C_2 in (1) represent the weighting of the stochastic acceleration terms that pull each particle towards particle's best and global best positions. Low values allow particles to roam far from target regions while high values results in abrupt movement towards or past target regions. Each acceleration constant is usually taken to be 2.0 for almost all applications. The selection of population size is problem dependent but the most common population size selected is 20-50. In PSO a parameter called inertia weight is introduced for balancing the global and local search. The corresponding equation (1) for velocity is modified as

$$v_{id} = W * v_{id} + C_1 * rand_1() * (p_{id} - x_{id}) + C_2 * rand_2() * (p_{gd} - x_{id}) \quad (3)$$

$$x_{id} = x_{id} + v_{id} \quad (4)$$

The inertia weight, W has characteristics that are reminiscent of temperature parameter in the simulated annealing. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. By linearly decreasing the inertia weight from a large value(close to unity) to a small value through the course of PSO run, the PSO tends to have more global search ability at the beginning of the run while possessing more local search ability near the end of the run.

In this paper the performance of PSO algorithm with early inertia weight has been investigated by extensive experimental studies of forecasting of two important stock prices.

III. Application of PSO to Stock Market Prediction

Referring to Fig. 1, the adaptive linear combiner is essentially an adaptive finite impulse response (FIR) filter having number of inputs equal to the number of features in the input pattern derived from the stock market series. The weights of the combiner are considered as the particles and initially their values are set to random numbers. A population of such random particle is chosen. Each particle updates its values using the PSO principle by way of minimizing the mean square error (MSE) as the cost function. The details of optimization is discussed in sequence.

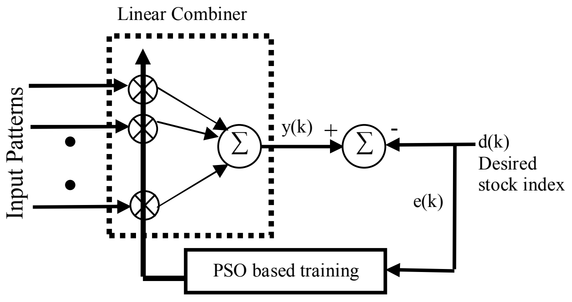

Fig. 1 A stock market forecasting model using adaptive linear combiner with PSO based training

Steps involved in PSO based training of the forecasting model :

i. The coefficients of the model are initially chosen from a swarm of I particles (birds). Each particle constitutes D number of parameters and each parameter represents a weight of the linear combiner.

ii. K number of patterns each containing ten features are obtained from the past stock indices.

iii. Each of the K patterns are passed through the linear combiner, multiplied with the weights and the partial sums are added together to give $y(k)$.

iv. The output of the linear combiner, $y(k)$ is then compared with corresponding normalized desired stock index, $d(k)$ to produce the error, $e(k)$. At the completion of all patterns K errors are produced. The mean square error (MSE) for a set of parameters (corresponding to ith particle) is determined by using the relation.

$$MSE(i) = \frac{\sum_{k=1}^{K} e^2(k)}{K}$$

This is repeated for I times.

v. Since the objective is to minimize MSE (i), $i = 1$ to I the PSO based optimization is used.

vi. In PSO the velocity and position of each bird is updated using (3) and (4) given in Section II.

vii. In each generation the minimum MSE (MMSE) is obtained and plotted against generation to show the learning characteristics of the model

viii. The learning process is stopped when MMSE reaches the minimum possible floor level.

IV. Simulation Study

A. Experimental Data

The data for the stock market prediction experiments has been collected for Standard's & Poor's 500 (S&P 500), USA and Dow Jones Industrial Average (DJIA), USA. The experimental data used consists of technical indicators and daily close price of the indices. The total number of samples for the stock indices is 3228 trading days, from 3rd January 1994 to 23rd October 2006. Each sample consists of the closing price, opening price, lowest price, highest price and the total volume of stocks traded for the day. Ten Technical indicators are selected as feature subsets by the review of domain experts and prior research. These are computed from the raw data as indicated in the Table 1.

The data is divided into two sets – training and testing sets. The training set consists of 2510 samples and the rest is set aside for testing. All the inputs are normalized to values between -1 to +1. The normalization is carried out by expressing the data in terms of the maximum and minimum value of the dataset.

TABLE 1
SELECTED TECHNICAL INDICATORS AND THEIR FORMULA

Technical Indicators used	Formula
Exponential Moving Average (EMA) (3 numbers)	$(P \times A) + (\text{Previous EMA} \times (1 - A))$; A=2/(N+1) P – Current Price, A- Smoothing factor, N-Time Period (EMA10, EMA20 and EMA30 are calculated using the given formula)
Accumulation/ Distribution Oscillator (ADO)	$\frac{\text{(C.P - L.P) - (H.P - C.P))}}{\text{(H.P - L.P)} \times \text{(Period's Volume)}}$ C.P – Closing Price, H.P – Highest price, L.P – Lowest price
Stochastic Indicator (STI)	$\%K = \frac{\text{(Today's Close - Lowest Low in K period)}}{\text{(Highest High in K period - Lowest Low in K period)}} \times 100$ $\%D$ = SMA of $\%K$ for the Period.
Relative Strength Index (RSI) (2 numbers)	$\text{RSI} = 100 - \frac{100}{1 + (U/D)}$ U= total gain/n, D= total losses/n, n = number of RSI period (RSI9 and RSI14 are calculated using the given formula)
Price Rate Of Change (PROC)	$\frac{\text{(Today's Close - Close X-period ago)}}{\text{(Close X-period ago)}} \times 100$ (PROC27 is calculated using the formula)
Closing Price Acceleration (CPACC)	$\frac{\text{(Close Price - Close Price N-period ago)}}{\text{(Close Price N-period ago)}} \times 100$
High Price Acceleration (HPACC)	$\frac{\text{(High Price - High Price N-period ago)}}{\text{(High Price N-period ago)}} \times 100$

B. Training and testing of the forecasting model

Training of the forecasting models are carried out using the PSO algorithm as given in Section III and the optimized weight values are obtained. Then using these weights the same forecasting models are again used for testing purpose.
The experiments are carried out to test the performance of the model for predicting the close price of the index one day, three, five, seven and fifteen days in advance.
The Mean Absolute Percentage Error (MAPE) is used to gauge the performance of the trained prediction model for the test data. The MAPE is defined as

$$MAPE = \frac{1}{N} \sum_{k=1}^{N} \left| \frac{d(k) - y(k)}{d(k)} \right| \times 100 \qquad (5)$$

where N is the number of test patterns.

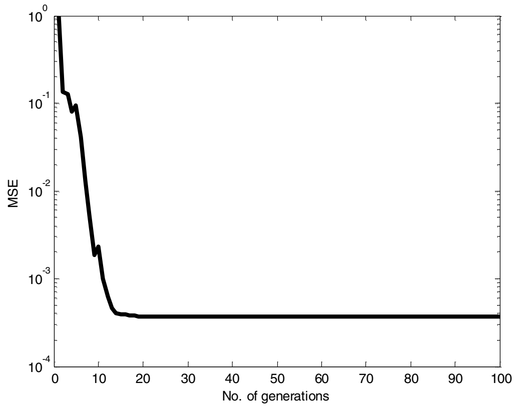

Fig. 2(a). Learning characteristics of S&P 500 for one day advance using PSO

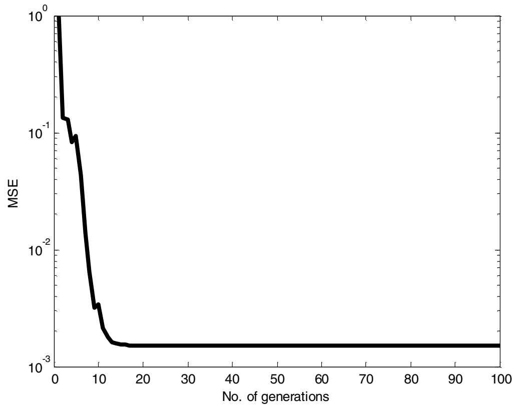

Fig. 2(b). Learning characteristics of DJIA for seven days advance using PSO

V. RESULTS AND DISCUSSION

The simulated, PSO model is used to predict the S&P500 and DJIA stock indices closing price one, three, five and seven days in advance. Fig. 2(a) and (b) show the learning characteristics of the model obtained through simulation for one day advance for S&P and seven days advance for DJIA respectively. These indicate that the mean square error (MSE) falls substantially during training and then settles at a minimum value indicating the convergence of weights .
The results provided here are the best evaluations obtained using the testing set. Various experiments are carried out, varying the selection of technical indicators as input to the network. This is done in an attempt to identify and eliminate less important statistical parameters and rogue parameters which adversely affect the network prediction performance. The ten technical indicators used for this simulation are EMA10, EMA20, EMA30, ADO, STI, RS19, RSI14, PROC27, CPACC and HPACC. Various parameters used in the simulation study for PSO are :

No. of particle =30, $W = 0.3$, C_1=1.49445, C_2 =1.49445 ; To compare the performance of the proposed model a best possible MLP model with {9-3-1} neuron is also simulated using same technical indicators. The value of converging factor, η is choosen as 0.01. The training is carried out for 10000 iterations.

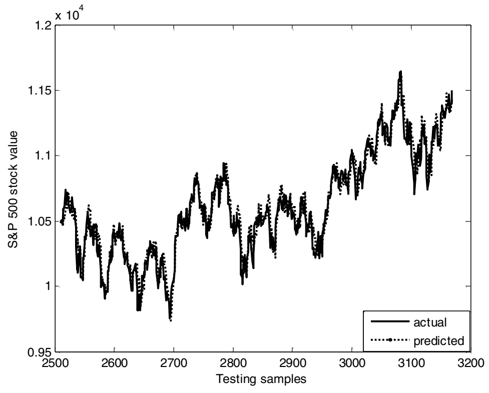

Fig. 3 Comparison of actual and one Day ahead predicted value during testing for S&P500 using PSO

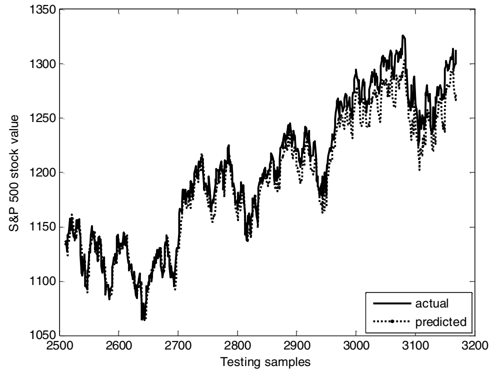

Fig. 4 Comparison of actual and one day ahead predicted value during testing for S&P500 using MLP

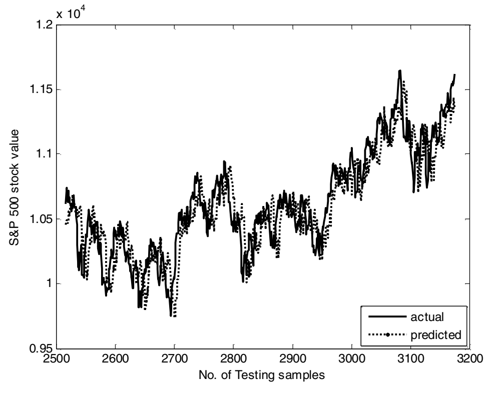

Fig. 5 Comparison of actual and seven days ahead predicted value during testing for S&P500 using PSO

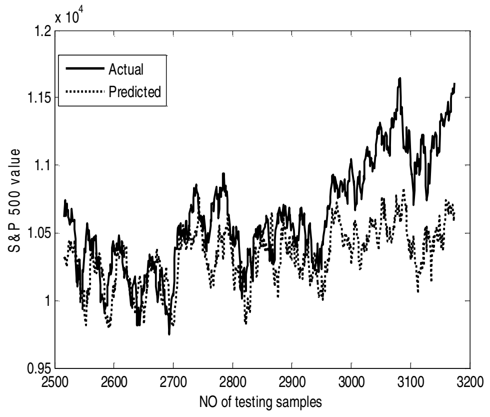

Fig. 6 Comparison of actual and seven days ahead predicted value during testing for S&P500 using MLP

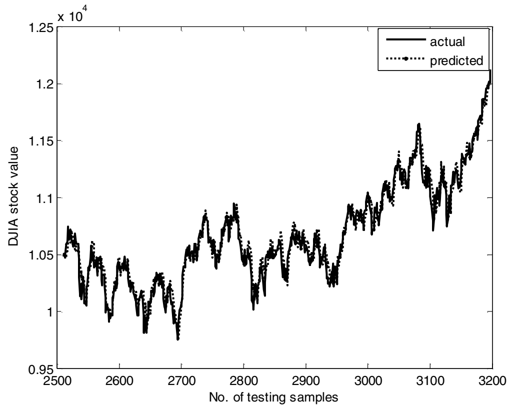

Fig. 7 Comparison of actual and one Day ahead predicted value during testing for DJIA using PSO

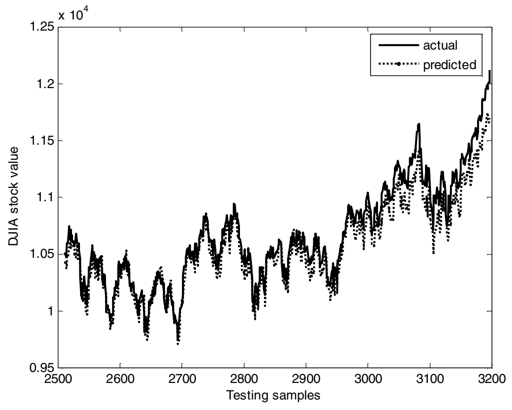

Fig. 8 Comparison of actual and one day ahead predicted value during testing for DJIA using MLP

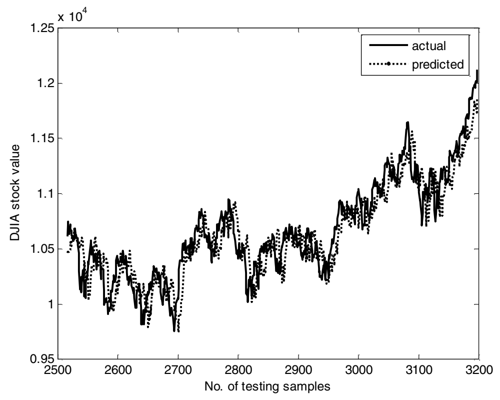

Fig. 9 Comparison of actual and seven days ahead predicted value during testing for DJIA using PSO

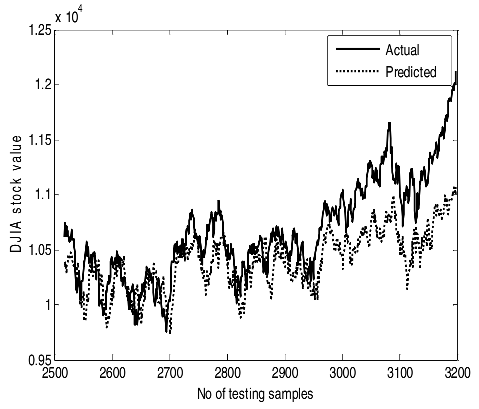

Fig. 10 Comparison of actual and seven days ahead predicted value during testing for DJIA using MLP

Figs. 3 and 5 display the actual vs. predicted graph for S&P500 index for one day and seven days ahead prediction respectively using BFO. The actual vs. predicted graph for S&P500 index for one day and seven days ahead prediction respectively using MLP are presented in Figs. 4 and 6. Similarly Figs. 7 and 9 display the actual and predicted DJIA stock value using test inputs and PSO model for one day and seven days respectively. The corresponding results obtained from the MLP model is shown in Figs. 8 and 10. Table-2 shows the MAPE and computation time of the PSO based model in comparison to the MLP based model for one day, three, five and seven days in advance prediction for S&P 500. The respective values for DJIA index is shown in Table-3.

The plots in Figs. 3-10 indicate that using test data the prediction performance of the PSO based model is superior to its MLP counterpart. This is true for both short term and long term prediction. Observations of Tables 2 and 3 indicate that the MAPE of PSO model is less than the MLP model for both the stock indices and for all days ahead prediction. In addition the proposed PSO model takes less time for training compared to its MLP counterpart.

TABLE 2
COMPARISON OF MAPE AND COMPUTATION TIME BETWEEN BFO AND MLP BASED FORECASTING MODELS for S&P 500

Days ahead	MAPE		Computation time in second	
	PSO	MLP	PSO	MLP
1	0.6661	1.0049	6.1237	26.4081
3	1.0801	2.0922	6.0143	26.6443
5	1.2451	2.5097	6.2016	26.7836
7	1.4664	3.1170	6.1643	26.8925

TABLE 3
COMPARISON OF MAPE AND COMPUTATION TIME BETWEEN BFO AND MLP BASED FORECASTING MODELS for DJIA

Days ahead	MAPE		Computation time in second	
	PSO	MLP	PSO	MLP
1	0.6558	1.0648	6.1417	26.9130
3	1.0782	2.0068	6.1276	27.3456
5	1.2492	3.7658	6.1219	27.9007
7	1.4767	5.6478	6.4229	28.5708

VI. CONCLUSION

The PSO based adaptive model for short and long term forecasting of stock indices is developed in this paper. The structure of the model is basically an adaptive linear combiner whose weights are updated using the PSO tool. To demonstrate the performance of the proposed model simulation study is carried out using known stock indices and its prediction performance is compared with MLP based standard forecasting model. The comparison indicates that the proposed model offers lesser computational complexity, better prediction accuracy and lesser training time compared to those obtained from the MLP model. Thus the proposed model is a new promising forecasting model for stock market prediction.

REFERENCES

[1] D. R. Brillinger, *Time series data analysis and theory*, Rinehart and Winston Inc., NY 1975.
[2] E. J. Hannan, *Multiple time series*, John Wiley and Sons, Inc., NY 1979.
[3] M. G. Kendall and A. Stuart, *The advanced theory of statistics: Design and Analysis and Time Serie,* Charles Griffin and Company Ltd, London 1968.
[4] Z. Griliches and M. D. Intriligator (Eds.), *Handbook of Econometrics*, Elsevier Science, 1994.
[5] T. H. Naylor, *Computer simulation experiments with models of economic systems*, John Wiley and Sons, Inc., NY 1971.
[6]Y. Bodyanskiy and S. Popov, "Neural network approach to forecasting of quasi-periodic financial time series", European journal on Operational Research, vol. 32, pp. 2513-2522, 2005.
[7] J. S. Zirilli, *Financial prediction using Neural Networks*, International Thompson Computer Press, London, 1997.
[8] S. Haykin, Neural Networks, *Pearson Edition*, 2004, pp. 732-785.
[9] S. Haykin, Neural Networks, *Pearson Edition*, 2004, pp. 256-312.
[10] A. Delgado, C. Kambhampati and K. Warwick, "Dynamic recurrent neural network for system identification and control", IEE Proc. on Control Theory and Applications, vol. 142, pp. 307-314, July 1995.
[11]Y. Hiemstra, "Applying neural networks and Genetic algorithms to Tactical asset allocation", Neuro-vest journal, May/June 1996.

[12] R. E. Dorsey and W. J. Mayer, "Genetic algorithms for estimation problems with multiple optima, non-differentiability and other irregular features", Journal of Business & Economics Statistics, vol. 13, no. 1, 1996.
[13] M. Lettau, "Explaining the facts with adaptive agents: the case of Mutual Fund Flows", Journal of Economic Dynamics and Control, vol. 21, 1997.
[14] R. Jiang and K. Y. Szeto, "Discovering investment strategies in Portfolio management" A Genetic Algorithm approach", Proc. Of 9th International conference on Neural Information Processing (ICONP-02), 2002, vol. 3, pp. 1206-1210.
[15] C. Neely, P. Weller, R. Dittnar, "Is technical analysis in the Foreign Exchange Market Profitable? A Genetic Programming approach", Journal of Financial and Quantitative analysis, vol 32, no. 4, 1997.
[16] S. Bhattacharya, O. Pictet and G. Zumbach, "Semantics for Genetic Programming based learning in High-frequency financial data", Proc. Of the 3rd Annual Genetic Programming Conference, 1998.
[17] S. H. Chen, C. H. Yeh and W. C. Lee, "Operation pricing with genetic Programming", Pro. of the 3rd Annual Genetic Programming Conference, 1998.
[18]N. K. Chidambaran, C. H. J. Lee and J. R. Trogucros, "An adaptive evolutionary approach to option pricing with Genetic Programming", Proc. Of the 3rd Annual Genetic Programming conference, 1998.
[19] H. Iba and Takashi Sasaki, " Using Genetic Programming to predict financial data", Proc. of IEEE Congress on Evolutionary Computation, , 6-9 July 1999, vol. 1, pp. 244-251.
[20] Md. R. Hassan, Baikunth Nath and Michael Kirley, " A fusion model of HMM, ANN and GA for stock market forecasting", Expert systems with applications, vol. 33, issue 1, pp. 171-180, July 2007.
[21] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", in Proc. IEEE Int. Conf. Neural Networks, Perth, Australia, Dec. 1995, pp. 1942-1948.