International Journal of Academic Research in Accounting, Finance and Management Sciences
Vol. 4, No.3, July 2014, pp. 144–154
E-ISSN: 2225-8329, P-ISSN: 2308-0337

www.hrmars.com

Stock Market Index Prediction via Hybrid. Inertia Factor PSO and Constriction Coefficient PSO

Morteza ASHHAR[1]
Amin Rostami MOTAMED[2]

[1]Department of Economic, Finance and accounting, College of management, Yzad University, Iran, E-mail: morteza.ashhar@yahoo.com
[2]Department of Accounting, Islamic Azad University Central Tehran Branch, Iran

Abstract *Conventional statistical techniques for forecasting are constrained by the underlying seasonality, non-stationary and other factors. Increasingly over the past decade, Artificial intelligence (AI) methods including Artificial Neural network (ANN), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) etc. have been used successfully to perform predictions in financial markets and other areas. This study presents a hybrid inertia factor and constriction coefficient PSO-based methodology to deal with the Stock market index problem. We will demonstrate the superiority an applicability of the proposed approach by using Tehran Stock Exchange Index (TSEI) data and comparing the outcomes with other PSO methods such as: standard PSO, Inertia Factor PSO and Constriction Coefficient PSO. Experimental results clearly show that a hybrid PSO approach meaningfully outperforms all of the other PSO methods in terms of MAD, MSE, RMSE and MAPE Evaluation statistics also, the proposed approach can be considered as a suitable AI model for stock market index forecasting problem.*

Key words Stock market forecasting, Hybrid PSO (hybrid intelligence model), Standard PSO, Inertia Factor PSO, Constriction Coefficient PSO

DOI: 10.6007/IJARAFMS/v4-i3/1108 URL: http://dx.doi.org/10.6007/IJARAFMS/v4-i3/1108

1. Introduction

Financial Forecasting or specifically Stock Market prediction has recently turned into one of the hottest fields of research due to its commercial applications owing to the high stakes and the attractive benefits that can be drawn from it.

Forecasting the index in stock markets has been a major challenge for common investors, businesses, brokers and speculators. The technical investors assume that the future trends in the stock market index are based, at least in part, on present and past events and data. However, financial time-series are one of the 'noisiest' and most 'non-stationary' signals present and are hence very difficult to forecast (Oh and Kim, 2002; Wang, 2003).

Financial time-series have high volatility and the time-series changes with time. In addition, movements in stock market index are affected by many macro-economical factors such as political events, firms' policies, general economic conditions, investors' expectations, institutional investors' choices, movement of other stock markets, psychology of investors etc.

Hence, stock market prediction is regarded as a challenging task in financial time-series forecasting (Atsalakis George and Valavanis Kimon, 2009).Using hybrid models or combining several models has become a common practice to improve forecasting accuracy and the literature on this topic has expanded dramatically over the past years.

Genetic Algorithm (GA) was incorporated to train Feed Forward Neural Network parameters (FNN-GA) (Ullah Khan *et al.,* 2008), optimum feature selection was applied to train the network parameters (Kim and Lee, 2004). Polynomial Neural Network based Genetic Algorithm (PNN-GA) was used to search between all possible input variables and to select the order of polynomial and Local Linear Wavelet Neural Network (LLWNN) optimized by Estimation of Distribution Algorithm (EDA) was proposed to train the network parameters (Chen *et al.*, 2005).On the other hand, researchers proved that ensemble neural networks and their training for the same task can produce more accurate results than using individual neural network (Chen *et al.*, 2006). Thus Particle Swarm Optimization (PSO) was used in training neural networks and was applied successfully in time series forecasting (Chaouachi *et al.,* 2009), moreover it was shown that it is better suited for real time series prediction applications than GA because it has fewer parameters to tune and will not follow the rule of the survival of the fittest (Sivanagaraju and Viswanatha, 2008). Based on this recognition, (PSO) algorithm was used to train the selective neural network ensemble(PSOSEN) (Zhang *et al.,* 2007) and Flexible Neural Tree (FNT) with its structure and parameters optimized using (PSO)incorporated with (GA) were applied in both Nasdaq100 and S&P 500 indices (Chen *et al.,* 2007).

Emad et al. (2005) presented a comparison among five recent evolutionary-based optimization algorithms: genetic algorithms, memetic algorithms, particle swarm, ant-colony systems, and shuffled frog leaping. The comparative results show the PSO method was generally found to outperform other algorithms in terms of success rate and solution quality (Emad *et al.,* 2005).

Also, some comparative research works for the real problems presented that the PSO based results have better performance than the based on GA (Gaing, 2004; Panda and Padhy, 2007).

Also, Ricardo de A. Araújo (2010) presented the swarm-based translation-invariant morphological prediction (STMP) method to overcome the random walk dilemma for financial time series forecasting. The proposed STMP method is inspired by the Takens theorem and consists of a hybrid model composed of a MMNN (Araújo *et al.,* 2006) combined with a PSO (Vandenbergh and Engelbrecht, 2004), which searches for the particular time lags capable of a fine-tuned characterization of the time series and estimates the initial (sub-optimal) parameters of the MMNN (weights, architecture and number of modules) (Araújo, 2010).

According to what was previously discussed, the present research aims at combining Inertia Factor PSO with Constriction Coefficient PSO in an attempt to reduce forecast errors.

2. Particle swarm optimizers

2.1. Standard PSO of Kennedy and Eberhart

PSO is a robust stochastic optimization technique based on the movement and intelligence of swarms. PSO applies the concept of social interaction to problem solving. It was developed in 1995 by James Kennedy (social-psychologist) and Russell Eberhart (electrical engineer) (Kennedy and Eberhart, 1995). It uses a number of agents (particles) that constitute a swarm moving around in the search space, looking for the best solution. Suppose that the i-th particle is flying over a hyper plane space, its position and velocity being denoted by $\vec{x_i}$ and $\vec{v_i}$.

The best previous position of the i-the particle is recorded and represented as pbest. The best previous position of the i-the particle is recorded and represented to serve as the index of the best particle among all the particles (N particle) using the symbol gbest. Consequently, the next flying velocity and position of the particle is updated at iteration k+1 using the following heuristic equations:

$$\vec{v}_i^{k+1} = \vec{v}_i^{k} + c_1 rand_1() \times (pbest_i - \vec{x}_i^{k}) + c_2 rand_2() \times (gbest - \vec{x}_i^{k}) \quad (1)$$

$$\vec{x}_i^{k+1} = \vec{x}_i^{k} + \vec{v}_i^{k+1} \qquad i = 1,2,3,\ldots,N_{Particle} \quad (2)$$

Where c_1 and c_2 are the cognitive and social learning rates, respectively. These two parameters control the relative importance of the memory (position) of the particle itself to the memory of the neighborhood, and are often both set to the same value to give each component equal weight. The variable $rand_1$ () and $rand_2$ () are two random functions that are uniformly distributed in the range [0,1]. As shown in Eq. (1), the two random values are generated independently, and the velocity of the particle is updated in relation to the variations on its current position, its previous best position, and the previous best position of its neighbors. After updating the velocity of the particle from Eq. (1), position is updates by adding the velocity vector to the current position to locate the next position. The stability and convergence of the algorithm have been

Vol. 4 (3), pp. 144–154, © 2014 HRMARS

analyzed theoretically by Clerc and Kennedy (Clerc and Kennedy, 2002), and using a dynamic system theory by Trelea (Trelea, 2003).

Fig.1 shows flowchart depicting a Standard PSO algorithm and Fig.2 shows the cognitive component search space contribution for two dimensions problem.

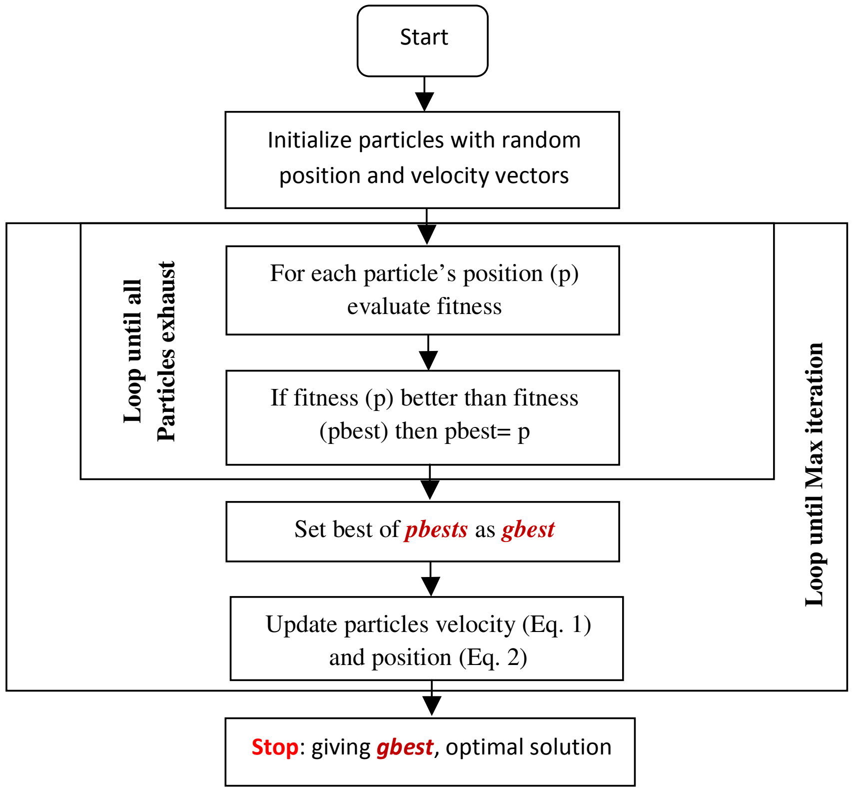

Figure 1. Flow chart depicting the Standard PSO algorithm

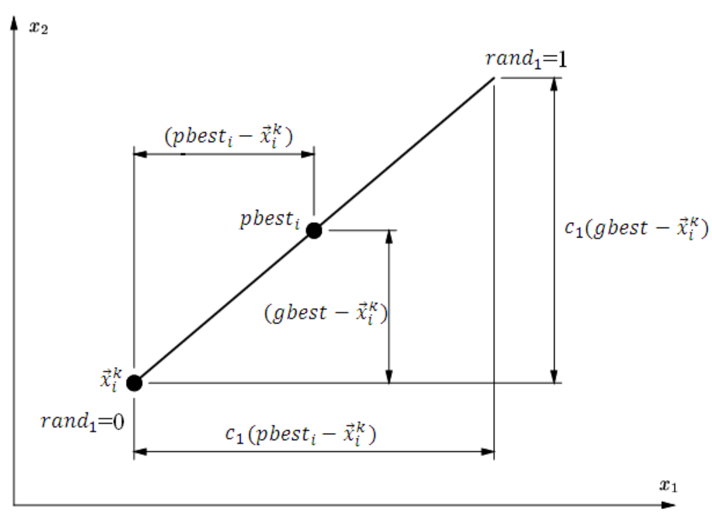

Figure 2. Cognitive component search space contribution for 2-D problem

2.2. PSO via Inertia factor of Shi and Eberhart

The original PSO of Kennedy and Eberhart are effective in determining optimal solutions in static environments, but it suffered from poor performance in locating a changing extreme. It was also necessary to impose a maximum value Vmax to avoid the particle exploded because there was no existing mechanism for controlling the velocity of a particle. In 1998a, Shi and Eberhart showed that PSO searches wide areas effectively, but tends to lack local search precision (Shi and Eberhart, 1998a). They introduced a control parameter called the inertia weight, w, to damp the velocities over time, allowing the swarm to converge more accurately and efficiently (Shi and Eberhart, 1998b). The modified PSO for updating the velocity vector is reformulated as follows:

$$\vec{v}_i^{k+1} = w\,\vec{v}_i^{k} + c_1 rand_1() \times (pbest_i - \vec{x}_i^{k}) + c_2 rand_2() \times (gbest - \vec{x}_i^{k}) \quad (3)$$

$$\vec{x}_i^{k+1} = \vec{x}_i^{k} + \vec{v}_i^{k+1} \qquad i = 1,2,3,\ldots,N_{Particle} \quad (4)$$

$$w = w_{Max} - \frac{\lfloor (w_{Max} - w_{Min}) \times iter \rfloor}{Max\ iter} \quad (5)$$

Where

$w_{Max} = initial\ weight$,

$w_{Min} = final\ weight$,

$iter = current\ iteration\ number$,

$Max\ iter = maximum\ iteration\ number$

Equation (3) represents a dynamically adapted formulation for velocity resulting in better fine tuning ability.

Looking at Eq. (3) reveals that the large inertia weight facilitates a global exploration while the small value facilitates a local search. Consequently, a dynamically adjust able formulation for inertia weight should be suitable for achieving a balance between global and local exploration and thus fastening search results. By introducing a linearly decreasing inertia weight into the original version of PSO, the performance of PSO has been significantly improved through parameter study of inertia weight (Shi and Eberhart, 1998a; Naka et al., 2001). Fig. 3 shows the concept of modification of a searching point by PSO via Inertia factor.

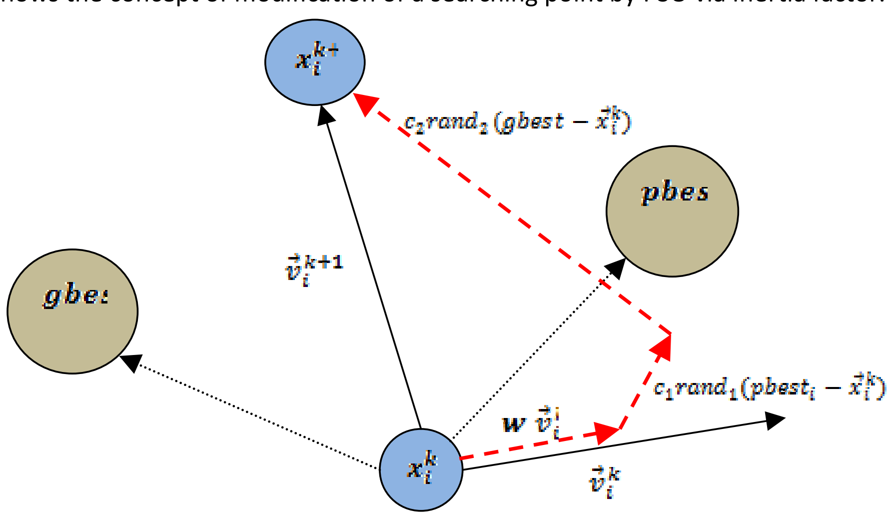

Figure 3. Concept of modification of a searching point by PSO via Inertia factor (Chan and Kumar Tiwari, 2007)

2.3. PSO via constriction coefficient of Clerc

In 1999, Maurice Clerc proposed the use of a constriction coefficient (Factor), K that improves PSO's ability to constrain and control velocities (Clerc, 1999) in the original PSO. Later, Eberhart and Shi found that K, combined with constraints on the maximum allowable velocity vector (V_{Max}), significantly improved the PSO performance (Eberhart and Shi, 2000). Also, when using the PSO with constriction coefficient (Factor), the setting of (X_{max}) on each dimension is the best approach. The constriction coefficient, K, implements a velocity control, effectively eliminating the tendency of some particles to spiral into ever increasing velocity oscillations. The formulation of K is expressed as follows:

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \quad (6)$$

Where $\varphi = c_1 + c_2$ and $\varphi > 4$, then the Kennedy and Eberhart's original PSO for velocity updating become:

$$\vec{v}_i^{k+1} = K\,[\vec{v}_i^{k} + c_1 rand_1() \times (pbest_i - \vec{x}_i^{k}) + c_2 rand_2() \times (gbest - \vec{x}_i^{k})] \quad (7)$$

Clearly, the constriction coefficient (Factor) K in Eq. (7) can be seen as a damping factor that controls the magnitude of the flying velocity of a particle. From the experiments in the literature, the Clerc's PSO has a potential ability to avoid particles being trapped into local optima effectively while possessing a fast convergence capability and was shown to have superior performance than the standard and modified PSOs. As shown in Eq. (6), the value of φ, defined as the sum of the cognitive and social learning rates, highly affects the constriction coefficient (Factor) K, and is an important parameter for achieving a good PSO with high performance. In general, when Clerc's constriction PSO is used, the common value for φ is set to4.1 and the constriction coefficient (Factor) K is approximately 0.729.

A hybrid of inertia factor PSO and constriction coefficient PSO

In order to keep apart from local solutions, and to prevent noisy algorithm output, both Inertia factor and constriction coefficient are applied in PSO relations.

For particle i in repeat cycle k of the algorithm the following relations are defined:

$$\vec{v}_i^{k+1} = K_i^k\,[w\,\vec{v}_i^{k} + c_1 rand_1() \times (pbest_i - \vec{x}_i^{k}) + c_2 rand_2() \times (gbest - \vec{x}_i^{k})] \quad (8)$$

In the above relation, w is calculated according to equation 5 and the amounts are reduced at every repeat cycle. As a result,$w_{Min} = 0.398$ and $w_{Max} = 0.975$ are calculated with trial and error.

In the implemented algorithm constriction coefficient is reduced as the following relation:

$$K_i^{k+1} = K_e \times K_i^k \qquad k \geq 1\ , K_e < 1 \quad (9)$$

Proper amount of K_e coefficient is calculated with trial and error and $K_e = 0.95$ has been chosen. Furthermore, if $c_1 = c_2 = 2.1$ the amount of $K_i^1 = 0.642$.

3. Model design and implementation

Daily data of Tehran stock exchange index from (2000-2013) has been used for designing and implementing the suggested model. Due to the importance of the issue, and in order to make an accurate forecast for the design and implementation of the model a forecast framework has been designed. This framework is demonstrated in figure (4). According to the framework, after data collection, these data are divided in two categories: train data and test data. 80% of the data are used for training the model and the remaining 20% are used for testing the model.

The, Mean Absolute Deviations(MAD), Mean Square Errors(MSE), Root Mean Square Errors(RMSE) and Mean Absolute Percentage Error (MAPE) are used to gauge the performance of the trained prediction model for the test data. The effort is to minimize the MAD, MSE, RMSE and MAPE for testing patterns in the quest for finding a better model for forecasting stock index movements. The MAD, MSE, RMSE and MAPE are given as:

1-$MAD = \frac{1}{n}\sum_{t=1}^{n}\left|X_t - \hat{X}_t\right|$ (10)

2-*MSE*= $\frac{1}{n}\sum_{t=1}^{n}(X_t-\hat{X}_t)^2$ (11)

3-*RMSE*= $\sqrt{\frac{1}{n}\sum_{t=1}^{n}(X_t-\hat{X}_t)^2}$ (12)

4-*MAPE*= $\frac{1}{n}\sum_{t=1}^{n}\left|\frac{X_t-\hat{X}_t}{X_t}\right|(100\%)$ (13)

3.1. Model selection using AIC and BIC

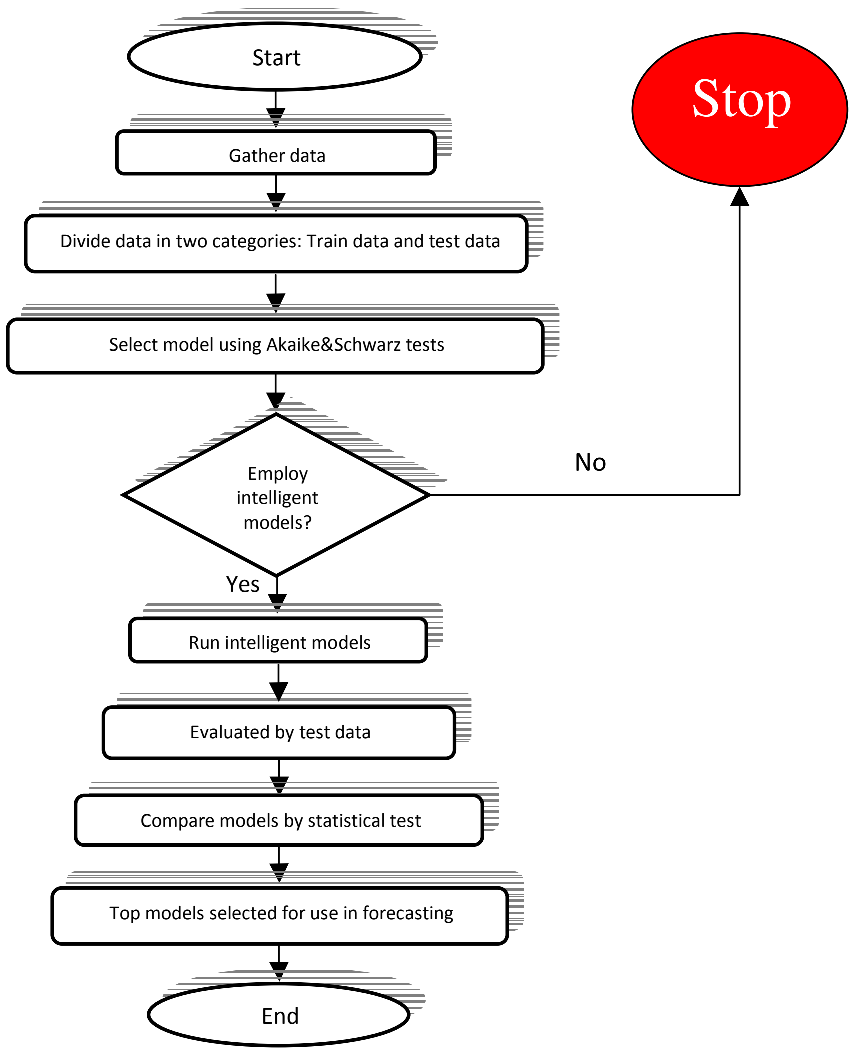

Figure 4. Flowchart designed to model predictions presented

The Akaike information criterion (AIC) is a measure of the goodness of fit of a statistical model. It was developed by Hirotsugu Akaike, under the name of "an information criterion" (AIC), and was first published by Akaike in 1974 (Hirotugu, 1974). It is grounded in the concept of information entropy, in effect offering a relative measure of the information lost when a given model is used to describe reality. It can be said to describe the tradeoff between bias and variance in model construction, or loosely speaking, that of the accuracy and complexity of the model.

The AIC is not a test of the model in the sense of hypothesis testing; rather, it provides a means for comparison among models a tool for model selection. Given a data set, several candidate models may be ranked according to their AIC, with the model having the minimum AIC being the best. In the general case, the AIC is:

$$AIC = 2k - 2\ln(L) \tag{14}$$

Where k is the number of parameters in the statistical model, and L is the maximized value of the likelihood function for the estimated model (Bozdogan, 2000).

Also, the Bayesian information criterion (BIC) or Schwarz criterion (SBIC) is a criterion for model selection among a class of parametric models with different numbers of parameters. Choosing a model to optimize BIC is a form of regularization. The BIC was developed by Gideon E. Schwarz, who gave a Bayesian argument for adopting it. It is closely related to the Akaike information criterion (AIC). In fact, Akaike was so impressed with Schwarz's Bayesian formalism that he developed his own Bayesian formalism (Schwarz, 1978). The BIC is an asymptotic result derived under the assumptions that the data distribution is in the exponential family (Kass and Wasserman, 1995). The formula for the BIC is:

$$BIC = n.\ln(\hat{\sigma}_e^2) + k.\ln(n) \tag{15}$$

Let:

- x = the observed data;
- n = the number of data points in x, the number of observations, or equivalently, the sample size;
- k = the number of free parameters to be estimated. If the estimated model is a linear regression, k is the number of regressors, including the intercept;
- $\hat{\sigma}_e^2$ is the error variance.

The error variance in this case is defined as:

$$\hat{\sigma}_e^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{16}$$

According to the above explanations, AIC and BIC Tests was performed and the results are shown in table 1.

Table 1. Lag estimation using AIC and BIC

Lag	LL	LR	Df	p-value	AIC	SBIC
0	-29019.2	-	-	-	19.5028	19.5048
1	-15659.5	26719	1	0.000	10.5252	10.5292
2	-15370.9	577.08	1	0.000	10.3319	10.338
3	-15351.3	39.224	1	0.000	10.3194	10.3275
4	-15350.1	2.4139	1	0.000	10.3193	10.3294
5	-15337.8	24.554*	1	0.000	10.3117*	10.3238*

Based on AIC and BIC Tests results, 5 lags are estimated for index time series.

3.2. Forecasting using Particle swarm optimization

Forecast was done using Standard PSO and Inertia factor PSO and Constriction coefficient PSO and Hybrid PSO and the target function is defined as below:

$$Min\ f(x) = \frac{1}{n}\sum_{i=1}^{n}|E_{Actual} - E_{Forecasted}|\ /E_{Actual} \tag{17}$$

In which, n is the total number of observations, E_{actual} is the actual amount and $E_{forecasted}$ is the forecasted amount of index. As demonstrated by AIC and BIC Tests, each day’s index is related to the last five days. Thus, $E_{forecasted}$ is calculated by using the following algorithm:

$$Model: E_{forecasted} = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \alpha_3 X_{t-3} + \alpha_4 X_{t-4} + \alpha_5 X_{t-5} + \alpha_6 \tag{18}$$

Therefore, PSO moves toward reducing the difference between forecasted and actual amounts. After analyzing various combinations of PSO parameters and adjusting them, eventually a proper system for the model was suggested which is depicted in table 2.

Table 2. PSO parameters for estimation of coefficients

Parameter		Value
Learning Parameters	C_1	2.1
	C_2	2.1
# of Particles		70
# of Iterations		200
# of Parameters		6
#Lower Bound		-5
#Upper Bound		5
w_{Max}		0.975
w_{Min}		0.389
K_i^1		0.642
K_e		0.95

According to the aforementioned Table 2, model parameters were estimated and these parameters are shown in table 3 respectively.

Table 3. Parameters estimated using PSO

Coefficient	Constant	α_1	α_2	α_3	α_4	α_5
Standard PSO	5	2.825682	-1.74161	0.930147	-1.55139	0.555024
Inertia PSO	-8.9456	2.825682	-1.74161	0.930147	-1.55139	0.545024
Constriction PSO	-9.564	2.994744	-2.59383	1.698763	-1.6898	0.585794
Hybrid PSO	15	2.67497	-2.4028	0.750568	-0.68985	0.66579

As shown in the above table3, forecasting was performed using estimated coefficients and suggested models. The following figures demonstrate both forecasted and actual amounts for all the models.

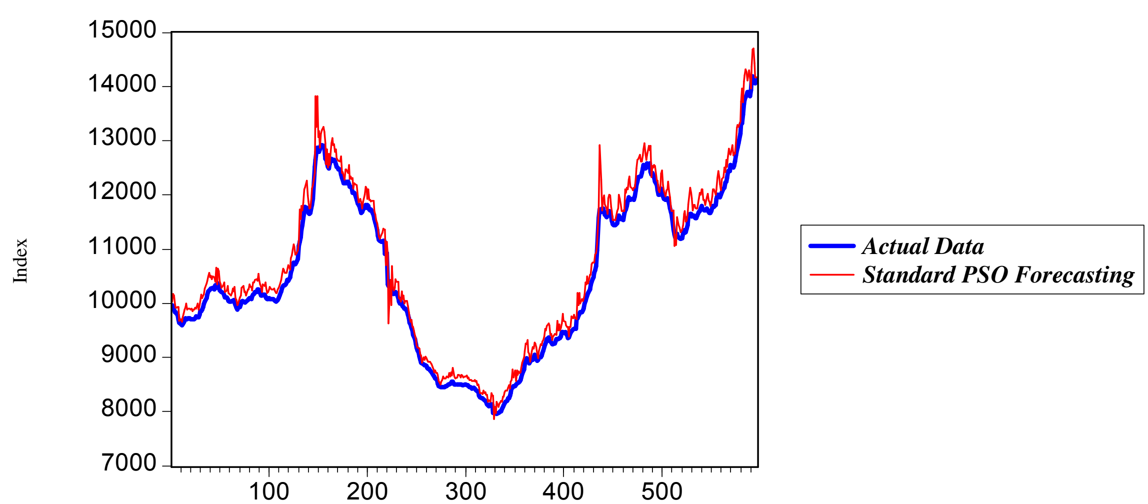

Figure 5. Comparison of data test and Standard PSO forecasting Model

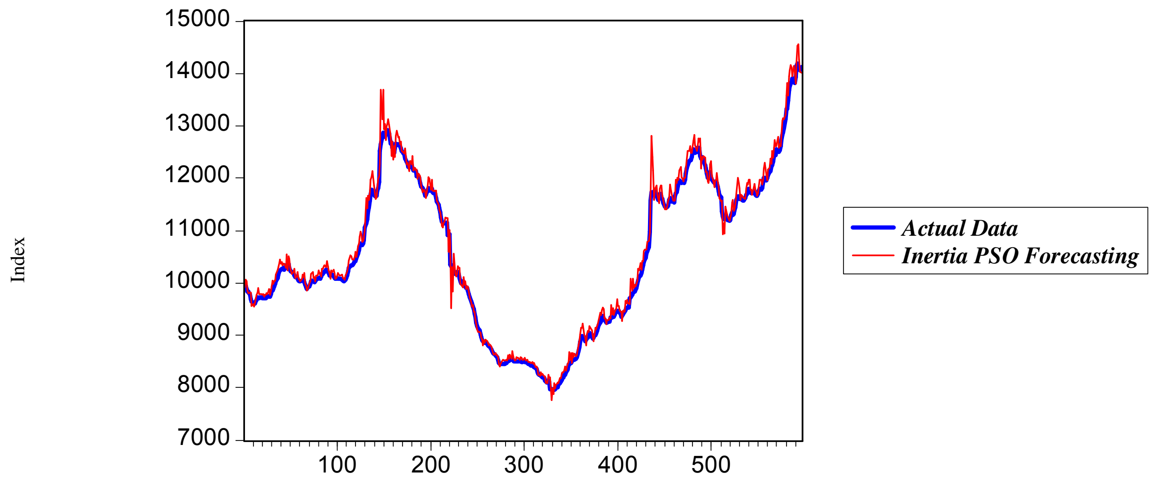

Figure 6. Comparison of data test and Inertia factor PSO forecasting model

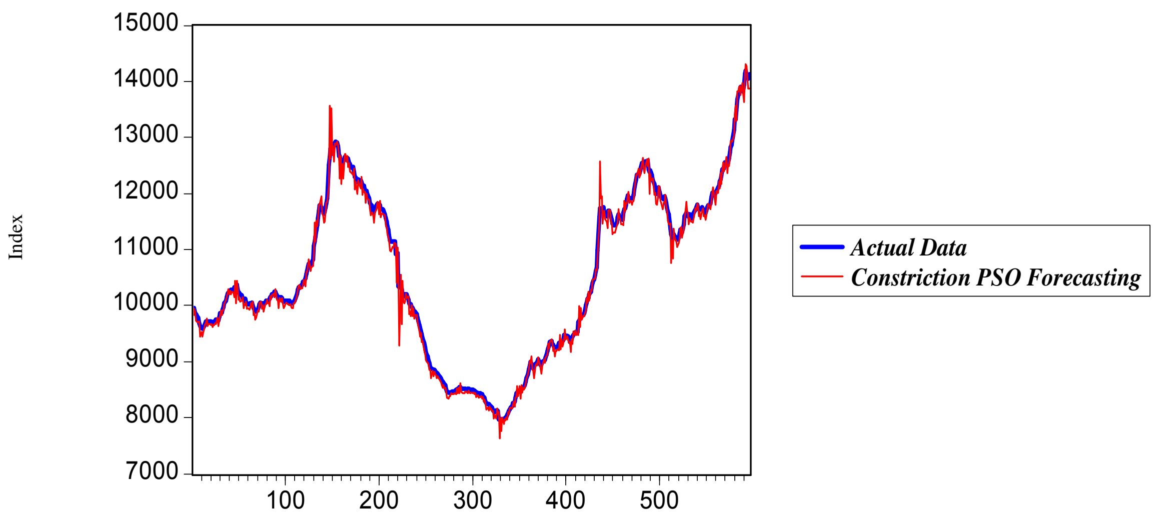

Figure 7. Comparison of data test and Constriction coefficient PSO forecasting model

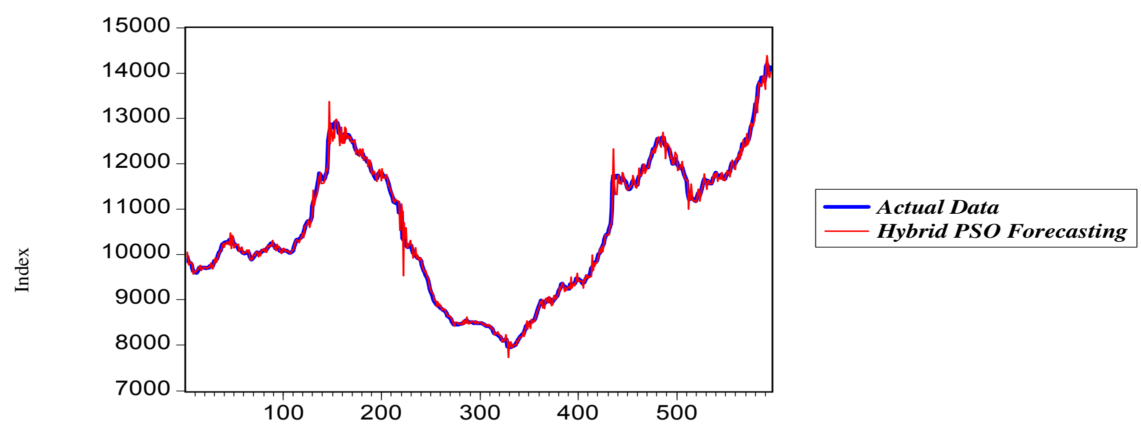

Figure 8. Comparison of data test and Hybrid PSO forecasting model

3.3. Comparison of evolutionary PSO models

By using Standard PSO, Inertia PSO, Constriction PSO and Hybrid PSO and comparing forecast errors, the best model could be selected and used for further forecasts. Table 4 shows the forecast models based on calculated errors.

Table 4. Forecast errors of all models for performance assessment

Errors / Forecasting Models	MAD	MSE	RMSE	MAPE
Standard PSO	208.9449	59016.13	242.9324	1.966316
Inertia PSO	108.9302	24488.23	156.4872	1.010697
Constriction PSO	90.88642	17712.5	133.0883	0.85229
Hybrid PSO	64.24397	10970.44	104.7398	0.592004

The calculated error of Hybrid PSO is lower than other PSO models and this shows the accuracy of this intelligent hybrid model. As shown in table 4, MAPE error for Hybrid PSO is 0.592004 which in comparison to Standard PSO, Inertia PSO, and Constriction PSO is respectively 60.893%, 41.426% and 30.54% lower. Therefore, the suggested hybrid model is more accurate and has less errors compared to other PSO models and is used for further forecasts.

4. Conclusions

Since the performance of stock market is assessed by the total index of stock price, it has a significant influence on economic development. In the present research Standard PSO, Inertia factor PSO, Coefficient PSO, and Hybrid PSO models are used for forecasting the total index of Tehran stock market. AIC and BIC tests are implemented to calculate Lag. Research results showed that Hybrid PSO decreases forecast errors more than other PSO models do. Therefore, this intelligent model is very accurate in forecasting.

References

1. Araújo, R.D. (2010). Swarm-based translation-invariant morphological prediction method for financial time series forecasting. Information Sciences. 180, 4784–4805.

2. Araújo, R. d., Madeiro, F., Sousa, R. d., Pessoa, L., Ferreira, T. (2006). An evolutionary morphological approach for financial time series forecasting. Proceedings of the IEEE Congress on Evolutionary Computation, 2467–2474.

3. Atsalakis George, S., Valavanis Kimon, P. (2009). Forecasting stock market short-term trends using a neuro-fuzzy based methodology. Expert System with Application. 36, 10696–10707.

4. Bozdogan, H. (2000). Akaike's information criterion and recent developments in information complexity. Journal of mathematical psychology. 44 (1), 62-91.

5. Chan, F. T., Kumar Tiwari, M. (2007). Swarm Intelligence Focus on Ant and Particle Swarm Optimization. Vienna, Austria: I-TECH Education and Publishing.

6. Chaouachi, A., Kamel, R. M., Ichikawa, R., Hayashi, H., Nagasaka, K. (2009). Neural Network Ensemble-based Solar Power Generation Short-Term Forecasting. World Academy of Science, Engineering and Technology, 54, 54-59.

7. Chen, H., Yuan, S., Jiang, K. (2006). Selective neural network ensemble based on clustering. Lecture Notes in Computer Science,Springer Verlag, 3971, 545-550.

8. Chen, Y., Dong, X., Zhao, Y. (2005). Stock Index Modeling using EDA based Local Linear Wavelet Neural Network. International Conference on Neural Networks and Brain (ICNN&B), 1646-1650.

9. Chen, Y., Yang, B., Abraham, A. (2007). Flexible neural trees ensemble for stock index modeling. Neurocomputing. 70, 697–703.

10.Clerc, M., Kennedy, J. (2002). The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space. IEEE Transactions on Evolutionary Computation, 658-73.

11.Clerc, M. (1999). The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. Proc. of ICEC, Washington D.C., 1951-1957.

12.Eberhart, R., Shi, Y. (2000). Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. Congress on Evolutionary Computing. 1, 84-88.

13.Emad, E., Tarek, H., Donald, G. (2005). Comparison among five evolutionary-based optimization algorithm. Advanced Engineering Informatics. 19, 43–53.

14.Farzi, S. (2008). A New Approach to Polynomial Neural Networks based on Genetic Algorithm. International Journal of Electrical and Computer Engineering, 3-7.

15.Gaing, Z. L. (2004). A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Transactions on Energy Conversion, 384–391.

16.Hirotugu, Akaike (1974). A new look at the statistical model identification. IEEE Transactions on Automatic Control, 716–723.

17.Kass, R. E., Wasserman, L. (1995). A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. Journal of the American Statistical Association, 90, 377–395.

18.Kim and, K., Lee, W. (2004). Stock Market Prediction using ANN with Optimal Feature Transformation. *Neural Computing & Applications. 13* (3), 225-260.

19.Naka, S., Genji, T., Fukuyama, Y. (2001). Practical Distribution State Estimation Using Hybrid Particle Swarm Optimization. Proc. of IEEE Power Engineering Society Winter Meeting. Columbus, Ohio, 2, 815-820.

20.Oh, K., Kim, K. (2002). Analyzing stock market tick data using piecewise non linear model. Expert System with Applications. 22, 249-255.

21.Panda, S., Padhy, N.P. (2007). Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design. Applied Soft Computing, ASOC-417, 1–10.

22.Kennedy, J., Eberhart, R.C. (1995). Particle Swarm Optimization. IEEE: International Conference on Neural Network, 1942-1948.

23.Schutte, J. F.,Groenwold, Albert A. (2005). The Particle Swarm Optimization Algorithm. 31(1), 93-108.

24.Schwarz, G. E. (1978). Estimating the dimension of a model. Annals of Statistics. 6 (2), 461–464.

25.Shi, Y., Eberhart, R. (1998a). A modified particle swarm optimizer. Proceedings of the IEEE Congress on Evolutionary Computation, Anchorage, AK, 69–73.

26.Shi, Y., Eberhart, R.C. (1998b). Parameter Selection in Particle Swarm Optimization. Annual Conf. on Evolutionary Programming, San Diego, 7, 591-600.

27.Sivanagaraju, S., Viswanatha, J. (2008). Capacitor Placement in Balanced and Unbalanced Radial Distribution System by Discrete Particle SwarmOptimization. ICGST-ACSE, 8 (1), 7-15.

28.Takens, F. (1980). Detecting strange attractor in turbulence, in: A. Dold, B. Eckmann (Eds.), Dynamical Systems and Turbulence. Springer-Verlag, New York.

29.Trelea, I.C. (2003). The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection. Information Processing Letters, 85, 317-325.

30.Ullah Khan, A., Bandopadhyaya, T.K., Sharma, S. (2008). Genetic Algorithm Based Backpropagation Neural Network Performs better than Backpropagation Neural Network in Stock Rates Prediction. International Journal of Computer Science and Network Security, 8 (7), 162-166.

31.Vandenbergh, F., Engelbrecht, A. (2004). A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation, 3, 225–239.

32.Wang, Y. (2003). Mining stock prices using fuzzy rough set system. Expert System with Applications, 24, 13-23.

33.Wang, Y. (2002). Predicting stock price using fuzzy grey prediction system. Expert System with Applications, 22, 33-39.

34.Zhang, X., Chen, Y., Yang, J.Y. (2007). Stock Index Forecasting Using PSO Bases Selective Neural Network Ensemble. International Conference on Artificial Intelligence (ICAI07), 260-264.