[Volume 3 issue 7 July 2015]
PageNo.1097-1105 ISSN : 2320-7167

INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH

Analysis and Prediction of Nifty Fifty Stocks Based On Best Replacement Optimization Technique

[1]Dr. T.Chitra kalarani, [2]S.Indrakala
[1,2]Associate Professor, Kunthavai Naacchiyar Govt. Arts College(W) Autonomous
Thanjavur, Tamilnadu
[1]E-mail-chitrajendran@yahoo.co.in
[2]E-mail-s.indirakala@yahoo.com

ABSTRACT
There is huge amount of data with complex uncertainty in stock market. Meanwhile. Efficient stock prediction is important in financial investment. Today, financial data analysis is becoming increasingly important in the stock market. As per companies gather more and more data from daily operations, they expect to extract useful information from existing collected data to help make reasonable decisions for different customer requirements. But this data values keeps on fluctuating day by day. So it is very difficult to predict the future value of the market. Although there are various techniques implemented for the prediction of stock market values, but the predicted values are not very accurate and error rate is more. The present paper introduces the best replacement Optimization technique to develop an efficient forecasting model for prediction of niftyfifty stock index. Results presented in this paper show that the proposed model has fast convergencespeed, and it also achieves better accuracy than compared techniques in most cases.
Key Wards: Stock Market, Swarm Initialization, BRO Algorithm

INTRODUCTION
The process of making assumptions of future Changes based on existing data is Forecasting. The more accurate the forecasting, the more it could be helpful to make decisions for future. Empowering the managers in all businesses to modify current situation in order to achieve the favorable results in future is the key use of forecasting. Forecasting stock price has always been a serious issue in financial fields.Stock Market prediction is an attractive field for research due to its commercial applications and the attractive benefits it offers. It follows stochastic, non-parametric and nonlinear behavior. Stock market forecasters focus on developing approaches to successfully forecast/predict index values or stock prices, aiming at high profits using well defined trading strategies. Different techniques are utilized in the stock market for prediction tasks, such as ARCH[16], SVM[3], Bayesian network[6], Evolutionary algorithms[10], Fractal Geometry[19], Fuzzy Logic[5], Granular Computing[1], HMM[12] and so on.
As a result of high nonlinear, volatility, irregularity, and noisy environment of stock markets, it is the difficult to construct stock market behavior model and predict stock price movement direction[18]. In the past years several models and techniques had been developed to stock price prediction. Also choosing a suitable training and prediction method is still very critical problem.Kuan and Liu (1995)discussed forecasting of foreign exchange rates using ANNs. They showed that a properly designed ANN has lower out-of-sample mean squared prediction error relative to the random walk model. Sexton et al. [17] theorized that the use of momentum and start of learning at random points may solve the problems that may occur in training process in 1998. Phua et al. [15] applied neural network with genetic algorithm to the stock exchange market of Singapore and predicted the market direction with an accuracy of 81%. Kim and Han (2000) [10] used a genetic algorithm to transform continuous input values into discrete ones. The genetic algorithm was used to reduce the complexity of the feature space. Forecasting Of Indian Stock Market Index Using Artificial Neural Network. Kishikawa and Tokinaga (2000) used a wavelet transform to extract the short-term feature of stock

trends. Kim and Han (2000)used neural network modified by Genetic Algorithm. Kim and Chun (1998) used refined probabilistic NN (PNN) to predict a stock market index.

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. In this paper, BRO Algorithm as a modified version of PSO Algorithm is used to predict the stock market and then analyzed the Nifty 50 index value from1January 2008 to 11 September 2014. The illustrations showed that this algorithm was efficient in stock forecasting.

The structure of the rest of paper will be as follows: In section 2, deals with the basic model of BRO Algorithm. Section 3 predictionAlgorithm is discussed. Section 4 the results of our simulations are shown and finally the conclusion is done in section 5.

THE BASIC MODEL OF BRO ALGORITHM

Consider the global optimum of an n-dimensional function defined by

$$f(x_1, x_2, x_3, \ldots, x_n) = f(x^*) \qquad (1)$$

Where $x_i$is the search variable, which represents the set of free variables of the given function? The aim is to find a value x^*such that the function $f(x^*)$is either a maximum or a minimum in the search space.

2.1. Derivation of the BRO Equations

The kinematic equation by which a particle's final position vector can be calculated from its initial position and velocity if acceleration is constant over the time period:

$$\vec{x} = \vec{x}_o + \vec{v}_0 t + \frac{1}{2}\vec{a}t^2 \qquad (2)$$

$t$between position updates is 1 iteration; hence, and t the corresponding dimensional analysis can be dropped to simplify iterative computations. Instead, authors generally use k to denote values of the current iteration and $k + 1$ for values of the ensuing iteration. Using this notation, the basic position update equation of physics can be rewritten for iterative computation as

$$\vec{x}(k+1) = \vec{x}(k) + \vec{v}(k) + \frac{1}{2}\vec{a}(k) \qquad (3)$$

The *global best*, which is the best solution found by the swarm through the current iteration, k . Modeling cognition, each particle also iteratively accelerates toward its personal best, which is the best solution that it personally has found.

The cognitive and social acceleration constants, $c_1$and c_2 respectively, determine how aggressively particles accelerate based on the cognitive and social information available to them: these can be set identically, or a preference can be given to either acceleration type. The subtraction in Equation (3) ensures that any particle which is distant from the social best will accelerate toward it more strongly than a particle nearby. Similarly, the subtraction in Equation (4) ensures that any particle that is distant from its cognitive best will accelerate toward it more strongly than were it nearby. As a conceptual example, one could imagine children playing both indoors and out when their mother announces that dinner is ready: those farther away might be expected to run more quickly toward the dinner table.

Social acceleration:

$$c_2(\vec{g}(k) - \vec{x}_i(k)) \qquad (4)$$

Where

c_2 $i$sthesocialacceleration constant ,

(k) $i$stheposition vector of particle"i " at iteration"k ",

$\vec{g}(k)$is the global best of all particles at iteration " k "

Cognitive acceleration:

$$c_1(\vec{g}(k) - \vec{x}_i(k)) \qquad (5)$$

where:

$c_1$is the cognitive acceleration constant,

$\vec{x}_i(k)$is the position vector of particle"i " at iteration" k",
$P_i(k)$is the personal best of particle "i " at iteration "i "
Equation (3) defines the social acceleration of Global Best (Gbest). Local Best (Lbest) limits each particle's social sphere to knowledge of the best solution found by its neighbors instead of immediately granting each particle knowledge of the best solution found so far by the entire search team.
Substituting the sum of these two acceleration terms for $\vec{a}(k)$in Equation (.3), while applying the subscript adopted in (4) and (5), produces Equation (6).

$$\vec{x}_i(k+1) = \vec{x}_i(k) + \vec{v}_i(k) + \frac{1}{2}c_1\left(\vec{P}_i(k) - \vec{x}_i(k)\right) + \frac{1}{2}c_2\left(\vec{g}(k) - \vec{x}_i(k)\right) \qquad (6)$$

Having replaced physical acceleration in the position update equation of physics with social and cognitive modeling, the next step toward producing a stochastic search algorithm is the replacement of with a pseudo-random number sampled per dimension from the uniform distribution between 0 and 1, . $U(0,1)$. Note that the expected or mean value of the distribution is still. Designating the first vector of pseudo-random numbers as $\vec{r}_{1i}$ and the second as $\vec{r}_{2i}$ produces Equation (6).

$$\vec{x}_i(k+1) = \vec{x}_i(k) + \vec{v}_i(k) + c_1\vec{r}_{1i} \circ \left(\vec{P}_i(k) - \vec{x}_i(k)\right) + c_2\vec{r}_{2i} \circ \left(\vec{g}(k) - \vec{x}_i(k)\right) \qquad (7)$$

For convenience, the rather long Equation (7) is separated into a velocity update equation (8) and a position update equation (9). This primarily helps with record keeping since each value can be stored separately for post-simulation analysis. Substituting Equation (8) into (9) shows equivalency to (7).
For convenience, the rather long Equation (7) is separated into a velocity update equation (8) and a position update equation (9). This primarily helps with record keeping since each value can be stored separately for post-simulation analysis. Substituting Equation (8) into (9) shows equivalency to (7).

$$\vec{v}_i(k+1) = \vec{v}_i(k) + c_1\vec{r}_{1,i} \circ \left(\vec{P}_i(k) - \vec{x}_i(k)\right) + c_2\vec{r}_{2,i} \circ \left(\vec{g}(k) - \vec{x}_i(k)\right) \qquad (8)$$

$$\vec{x}_i(k+1) = \vec{x}_i(k) + \vec{v}_i(k+1) \qquad (9)$$

Since its conception, Equation (8) has developed two mechanisms by which to improve search behavior. The inertia weight,ω roughly simulates friction in a computationally inexpensive manner by $\omega\epsilon(-1,1)$ carrying over to the next iteration only a user-specified percentage of the current iteration's velocity. This is done by multiplying the velocity of the current iteration by 1 as shown in the first term of Equation (10) . The constriction models use a constriction coefficient instead, but the popular Type 1" parameters can be converted to Clerc's Equivalents for use in Equation (10).

$$\vec{v}_i(k+1) = \omega\vec{v}_i(k) + c_1\vec{r}_{1,i} \circ \left(\vec{P}_i(k) - \vec{x}_i(k)\right) + c_2\vec{r}_{2,i} \circ \left(\vec{g}(k) - \vec{x}_i(k)\right) \qquad (10)$$

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \times iter}{Max\ iter}$$

where

$$\omega_{max} = initial\ weight,$$
$$\omega_{min} = final\ weight,$$
$$iter = current\ itertation\ number$$
$$Max\ iter = maximum\ \ itertation\ number$$

The equation (10) is enhanced by limiting the iteration using threshold level of global fitness value where the Global Fitness Value ranges from Global Fitness± 1.

$$\vec{v}_i(k+1) = \sum_{i=GF-1}^{i=GF+1}\left(\omega\vec{v}_i(k) + c_1\vec{r}_{1,i} \circ \left(\vec{P}_i(k) - \vec{x}_i(k)\right) + c_2\vec{r}_{2,i} \circ \left(\vec{g}(k) - \vec{x}_i(k)\right)\right) \qquad (11)$$

The equation (11) improves the efficiency of the iteration and reduces the processing. The other restriction imposed on velocity is essentially a speed limit. Rather than limiting the vector magnitude itself, the computationally simpler approach of limiting each component is implemented as shown in Equation (11), which limits the magnitude indirectly.

$$\vec{v}_{i,j}(k+1) = sign\left(\vec{v}_{i,j}(k+1)\right) \times max\left(\left|\vec{v}_{i,j}(k+1)\right|, v_j^{max}\right) \quad (12)$$

where $j \in \{1,2,\ldots.,n-1,n\}$, and n denotes the problem dimensionality

This limits the maximum step size on dimension j by clamping: (i) values greater than$v_j^{max}$to amaximum value of v_j^{max} , and (ii) values less than $-v_j^{max}$to a minimum of $-v_j^{max}$. From a physical perspective, particles with clamped velocities are analogous to birds with limited flying speeds. Considering the psychological aspects of the algorithm, clamped velocities could also be considered analogous to self-limited emotive responses. Concerning the calculation of v_j^{max} , suppose the feasible candidates for an application problem are [12, 20] on some dimension to be optimized. Clamping velocities to 50%, for example, of x_{max}would allow particles to take excessively large steps relative to the range of the search space on that dimension; in this case, the maximum step size would be 0.5*20 =10 . But stepping 10 units in any direction when the search space is only 8 units wide would be nonsensical. Since real-world applications are not necessarily centered at the origin of Euclidean space, it is preferable to clamp velocities based on the range of the search space per dimension in order to remove dependence on the frame of reference [10]; hence, subscript *j* is included in Equation (11) for sake of generality; but it can be dropped for applications with the same range of values per dimension.

BRO ALGORITHM Input:

m: The swarm size; $c1$, $c2$: positive acceleration constants; w: inertia weight

$MaxV$: Maximum velocity of particles

$MaxGen$: Maximum generation

$MaxFit$:Maximum fitness value

Output:

$Pgbest$: Global best position

Begin

Swarms $\{x_{id}, v_{id}\}$ =Generate(m); /* Initialize a population of particles with random positions and velocities on S dimensions*/

Pbest(i)=0; *i*=1,...,*m*,*d*=1,...,*S*

Gbest=0; Iter=0;

While(Iter<MaxGen and Gbest<MaxFit)

{For(every particle i)

{ Fitness(i)=Evaluate(i);

IF(Fitness(i)>Pbest(i))

{Pbest(i)=Fitness(i); $p_{id}= x_{id}$; d =1,...,*S* }

IF(Fitness(i)>Gbest)

{Gbest=Fitness(i); gbest=i;}

}For(every particle i)

{

For(every d){ $v_{id} = w*v_{id} +c_1 *rand()*(p_{id}-x_{id})+c_2 *Rand()*(p_{gd}-x_{id})$

IF($v_{id} >MaxV$) {$v_{id} = MaxV$;}

IF($v_{id} <-MaxV$) {$v_{id} =-MaxV$;}

xid= xid+ vid

}

}

Iter=Iter+1;

}/*

rand() and Rand() are two random functions in the range [0,1]*/

Detect{gbest}

While(GF-1){

vid = w*vid +c1 *rand()*(pid -xid)+c2 *Rand()*(pgd -xid)

}do GF+1

Use threshold level of global fitness value where the Global Fitness Value ranges from Global Fitness ± 1.

$\vec{v}_i(k+1) = \sum_{GF-1}^{GF+1}(\{\ vid\ =\ w*vid\ +c1\ *rand()*(pid\ -xid\ )+c2\ *Rand()*(pgd\ -xid\ )$Improves the efficiency of the iteration and reduce the processing

End

Psedocode:

Step1: The Input dataset S is taken from Nifty fifty Companies,
Step1: Initialize position and velocity of all the particles randomly in the N dimension space from dataset S.
Step2: Evaluate the fitness value of each particle, and update the global optimum position.
Step3: According to changing of the gathering degree and the steady degree of particle swarm, determine whether all the particles are re-initialized or not.
Step4: Determine the individual best fitness value. Compare the l_p of every individual with its current fitness value. If the current fitness value is better, assign the current fitness value to l_p.
Step5: Determine the current best fitness value in the entire population. If the current best fitness value is better than the g_p , assign the current best fitness value to g_p .
Step6: For each particle, update particle velocity,
Step7: Repeat the iteration of the particle using $gbest$ fitness value and limit the Iteration of the particle.
Step8: Update particle position.
Step9: Repeat Step2 - 7 until a stop criterion is satisfied or a predefined number of iterations are completed. While maximum iterations or minimum error criteria is not attained Particles' velocities on each dimension are clamped to a maximum velocityv_{max}. If the sum of accelerations would cause the velocity on that dimension to exceedv_{max}, which is a parameter specified by the user. Then the velocity on that dimension is limited to v_{max}.

RESULTS AND ANALYSIS

The data for the stock market prediction experiments has been collected for Nifty 50. The experimental data used consists of technical indicators and daily prices of the indices. The total number of samples for the stock indices is 3462 trading days. Each sample consists of the closing prices, opening prices, lowest prices, highest prices and total volume of stocks traded for the day. The data is divided into two sets training and testing sets. The training set consists of 365 samples and 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. BSO Parameters for estimation of coefficients Parameter value

Parameters	value
Number lower bound	-5
Number upper bound	5
Population Size	180
Number of iterations each day	200
Acceleration constant	1 2
Acceleration constant	2 2
Initial inertia weight	0.9
Final inertia weight	0.4
Minimum Error Gradient	$1 * 10^{-25}$
Epochs Before Error Gradient Termination	15
ω_{max}	0.975
ω_{min}	0.389

Table 2. indicates Sample Testing dataset of BHEL Company

Date	Open Price	High Price	Low Price	Close Price	Total Volume	No of Trades	Turnover in(Rs.in Lakh)
09/11/14	223.35	216.1	219.55	222.7	50,78,407	47,593	1,11,65,93,545.00
09/10/14	222.9	218	219	219.1	35,78,141	29,873	78,85,46,803.50
09/09/14	224.45	220.5	223.9	221.2	34,24,536	26,656	75,94,06,149.10
09/08/14	225.8	222.1	223.45	223.85	34,63,052	35,702	77,48,32,714.30
09/05/14	227.6	221.1	227.1	222.8	51,33,487	46,291	1,14,67,33,300.00
09/04/14	235.1	226.05	235	227.05	1,06,90,351	55,956	2,43,45,62,223.00
09/03/14	242	235.35	240.55	236.85	40,41,846	32,857	96,21,29,678.30
09/02/14	243.1	237	238	240.3	36,34,223	37,961	87,47,04,332.40
09/01/14	243.5	237.05	240.7	238.5	54,59,840	53,841	1,31,10,24,781.00

Table 3. The accuracy with which the stock price was predicted for BHEL Company

Parameters	High Prices		Low Prices	
Dataset	Analysis Set(Training)	Validation Set(Test	Analysis Set(Training)	Validation Set(Test)
Total Objects	143191	3146	143191	3146
Objects Covered	870	167	870	167
Min Support	117	178	104.65	172.1
Max Support	2580	577	2561	566.85
Average Support	586.827217	317.8278443	575.993815	313.0538922
Min Accuracy	0.019369048	0.012131716	0.025824442	0.013231013
Max Accuracy	0.045675963	0.026943577	0.061074954	0.029251131
Average Accuracy	0.027031321	0.02191375	0.036029315	1.084262864

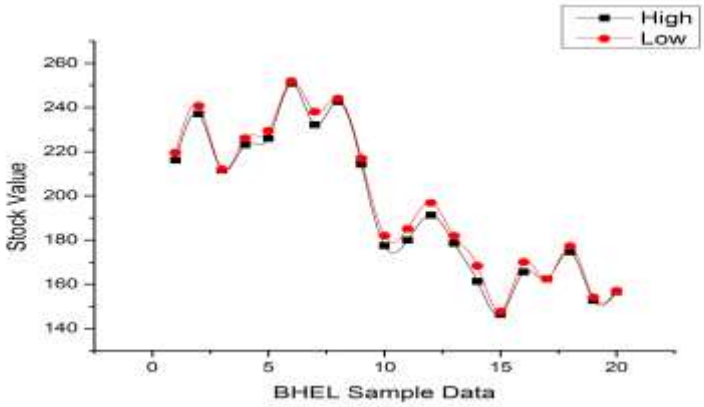

Figure 1 BRO TechniquePrediction of Low Price Value and High Price Value in BHEL Company

The Graph 1 indicates the Best Replacement Optimization Prediction of High Value and Low value in Stock Market Data from Jan 2008 to Sep 2014. The Sample Data is used to detect BRO and predict the High and Low value for BHEL Company.

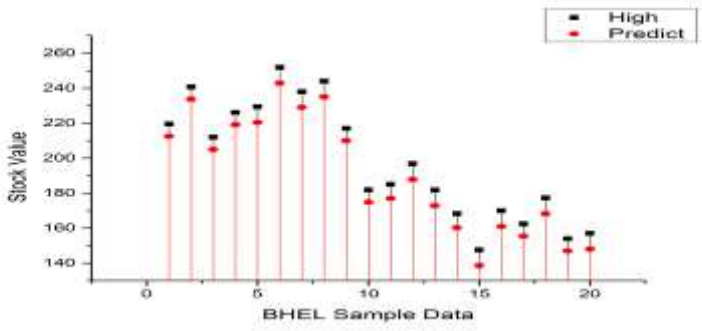

Figure 2 BRO Technique Prediction of High Price Value in BHEL Company

The Graph 2 indicates the Best Replacement Optimization Prediction of High Value in Stock Market Data from Jan 2008 to Sep 2014. The Sample Data is used to detect BRO and predict the High value for BHEL Company.

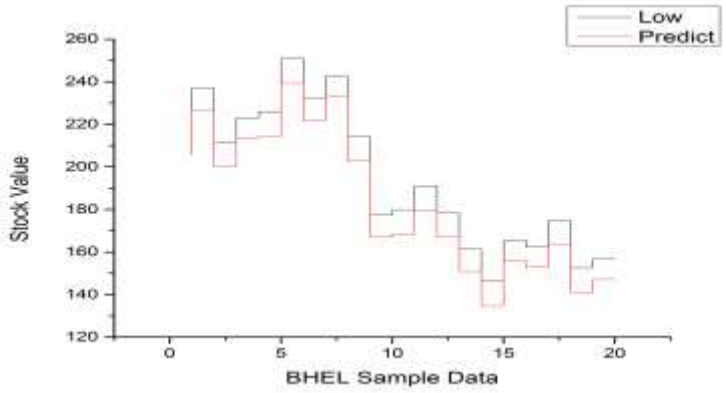

Figure 3 BRO Technique Prediction of Low Price Value BHEL Company

The Graph 3 indicates the Best Replacement Optimization Prediction of Low Value in Stock Market Data from Jan 2008 to Sep 2014. The Sample Data is used to detect BRO and predict the Low value for BHEL Company.

CONCLUSION

BRO is an evolutionary computation technique. The main idea is based in the way birds travel when trying to find sources of food, or similarly the way a fish school will behave. The way this behavior is modeled, is that the "particles" inside the "swarm" (or population) are treated as solutions to a given problem. The solution space for that problem is where the particles will be moving or traveling through, searching for the best solutions to the problem. The particles will travel following two points in the space; a leader in the swarm, which is chosen according to the global best solution found so far, 'and its memory. Every particle has a memory, which is the best solution visited by that specific particle. According to some experimental results show that BRO has greater" global search" ability, but the "local search" ability around the optimum is not very good. In order to enhance "local search" ability of BRO.

REFERENCES

1. Ding Shifei, Li Jianying, Xu Li, Qian Jun, "Research Progress of Granular Computing", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 5, No. 1, pp. 162- 172, 2011.
2. Hatem Abdual-Kader and Mustafa Abdul Salam, (2012) "Evaluation of Differential Evolution and Particle Swarm Optimization Algorithms at Training of Neural Networks for Stock Prediction " International Arab Journal of e-Technology, vol.2, No. 4, January 2012.
3. Ilias Tsiakas, "Overnight information and stochastic volatility: A study of European and US Stock Exchanges", Journal of Banking & Finance, vol. 32, no. 2, pp. 251-268, 2008.
4. Jasic, T. and D. Wood (2004), 'The Profitability of Daily Stock Market Indices Trades Based on Neural Network Predictions: Case Study for the S&P 500, the DAX, the TOPIX and the FTSE in the Period 1965–1999', Applied Financial Economics, 14(4): 285–97.
5. Jorge Ropero, Carlos Leon, Alejandro Carrasco, Ariel Gomez and Octavio Rivera, "Fuzzy Logic Applications for Knowledge Discovery: a Survey", IJACT: International Journal of Advancements in computing Technology, vol. 3, no. 6, pp. 187-198, 2011.
6. Kenneth Wolfe R, "Turning Point Identification and Bayesian Forecasting of a Volatile Time Series", Computers and Industrial Engineering, vol. 15 pp. 378-386, 1988.
7. Kim, K, J and I Han (2000) Genetic algorithm approach to feature discretization in artificial neural network for the prediction of stock price index." Published by Elsevier science, Ltd., Experts systems with application, 19, 125-132.
8. Kim, S, K., and S.H Chun (1998) "Graded Forecasting Using An Array Of Bipolar Predictions: Application Of Probabilistic Neural Network To A Stock Market Index." International Journal of Forecasting, 14, 323-337.
9. Kuan, C-M and T. Liu (1995), 'Forecasting Exchange Rates Using Feed forward and Recurrent Neural Networks', Journal of Applied Econometrics, 10(4): 347–64.
10. Kyoung-jae Kim, Ingoo Han, "Genetic Algorithms Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index", Expert Systems with Applications, vol. 19, no. 2, pp. 125-132, 2000.
11. Majhi R, Panda G, Sahoo G and Panda A, "Prediction of S&P 500 and DJIA stock indices using Particle Swarm Optimization technique" Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on Date 1-6 June 2008.

12. Md Rafiul Hassan, Baikunth Nath, "Stock Market Forecasting Using Hidden Markov Model: a New Approach", In Proceeding(s) of the 5th International Conference on Intelligent Systems Design and Applications, pp. 192-196, 2005.
13. Mizuno, H., Kosaka , M., Yajima , H. and Komoda N. (1998), Application of Neural Network to Technical Analysis of Stock Market Prediction, Studies in Information and Control , vol.7, no.3, pp.111-120.
14. Pantazopoulos, K. N., Tsoukalas, L. H., Bourbakis, N. G., Brun, M. J., & Houstis, E. N. (1998). Financial prediction and trading strategies using neurofuzzy approaches. IEEE Transactions on Systems, Man, and Cybernetics-PartB: Cybernetics, 28, 520–530.
15. Phua, P.K.H. Ming, D., Lin, W. (2000), Neural network With Genetic Algorithms For Stocks Predictio, Fifth Conference of the Association of Asian-Pacific Operations Research Societies, 5th – 7 th july, Singapore
16. Robert F. Engle, "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation", Econometrica, vol. 50, no. 4, pp. 987-1008, 1982.
17. Sexton, R. S., R. E. Dorsey and J.D.Dohnson (1998) , Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation, Decision Support systems 22, 171- 185.
18. Wei Shen, "Theory Survey of Stock Yield Prediction Models", International Journal of Economics and Finance, vol. 1, no. 1, pp. 175-182, 2009.
19. Willem Dekker, "The Fractal Geometry of the European EEL Stock", ICES Journal of Marine Science, vol. 57, no. 1, pp. 109-121, 2000.