Development and Performance Evaluation of Three Novel Prediction Models for Mutual Fund NAV Prediction

Ananya Narula*, Chandra Bhanu Jha* and Ganapati Panda**

E-mail: an14@iitbbs.ac.in; cbj10@iitbbs.ac.in; gpanda@iitbbs.ac.in

ABSTRACT

The paper compares the performance of three adaptive models based on Functional Link Artificial Neural Network (FLANN), Multi-Layered Perceptron (MLP) and Radial Basis Function (RBF) Networks employed for prediction of the net asset value (NAV) of a mutual fund scheme of a company through simulation study. The statistical features extracted from the past data are used to train the models. The prediction performance is evaluated using real life data. It is observed that the simple FLANN model predicts better for the NAV fifteen days ahead and higher. But for a short range prediction, the RBF model yields the best performance amongst the three models.

Introduction

A mutual fund is a type of professionally managed investment fund that pools money from many investors to purchase securities. It is most commonly applied only to those collective investment vehicles that are

* Indian Institute of Technology, Bhubaneswar

** Department of Electrical Engineering, Indian Institute of Technology, Bhubaneswar

ISSN 2348–0661 Print

Annual Research Journal of Symbiosis Centre for Management Studies, Pune
Vol. 3, April 2015, pp. 227–238

regulated and sold to the general public. Most mutual funds are open-ended, meaning stockholders can buy or sell shares of the fund at any time by redeeming them from the fund itself, rather than on an exchange.

A fund's net asset value or NAV equals the current market value of a fund's holdings minus the fund's liabilities, sometimes referred to as 'net assets'. It is usually expressed as a per-share amount, computed by dividing net assets by the number of fund shares outstanding. All mutual funds' buy and sell orders are processed at the NAV of the trade date. However, investors must wait until the following day to get the trade price.

Thus, in a way, the net asset value describes the company's current asset and liability position. An increasing value of NAV usually indicates that the company is growing and vice-versa. It is an important judging parameter for a potential investor or a sponsor to consider, so as to be able to make a wiser and adequately informed decision.

The day to day tracking and further analysis of the NAV of a mutual fund, hence, assumes prime importance. The present challenge is to foresee a fund's future performance with the least possible error.

Statistical methods have been widely employed previously for linear modeling of time-series data. The K-Nearest Neighbour (KNN) and Support Vector Machines (SVM) (Gao & Cherkassky, 2006) have been used for training the Linear Regression Model for real time pricing. The Auto Regressive Integrated Moving Average (ARIMA) (Priyadarshini & Chandra Babu, 2011), also known as Box-Jenkins model, has been used for forecasting of NAV of Indian mutual fund. The methods employ linear programming of the time-series data for the prediction. One major drawback of these models is their inability to capture the non-linearity in the data, which contributes to inaccuracy in the forecast. This calls for an inclination towards the usage of non-linear models. Soft and Evolutionary Computing (SEC) based techniques, with excellent non-linear model development properties, have been chosen for forecasting of currency exchange rates using the adaptive ARMA model with differential evolution (DE) based learning (Panda & Majhi, 2013), forecasting of retail sales using DE (Panda & Majhi, 2009), and active control of nonlinear noise processes using FLANN (Panda & Das, 2003). Neural networks have also been used widely in economic forecasting for market analysis and forecasting time series of political economy (Chakraborty, Mehrotra, Mohan & Ranka, 1992; Freisleben & Ripper, 1995).

The review of existing literature reveals that very little work has been reported on the development of adaptive nonlinear model for the long and short range prediction of NAV. Hence, in this paper, three nonlinear adaptive models based on the MLP, the FLANN and the RBF have been proposed. The MLP network is a standard adaptive structure but its complexity is high. On the other hand, though the RBF network finds extensive applications in many fields, fixation of required number of centres is a difficult task. The FLANN structure with trigonometric expansions reduces the number of layers to one thus leading to a simple adaptive structure. In the following section a brief overview of these three adaptive structures has been provided.

Functional Link Artificial Neural Network

As an Adaptive Predictor, the Functional Link Artificial Neural Network (FLANN) based adaptive model consists of a simple structure shown in Fig. 1. Its inputs are chosen to be the statistical features extracted from the past NAV time series. The inputs are expanded trigonometrically, multiplied with their respective weight values, and the products are then summed up to get the predicted output.

Let x_1, x_2 and x_3 be the features extracted form a given dataset. Then the trigonometric expansion consisting of 2P+1 terms for the first input x_1 would be of the form:

$\{E_{1-1}, E_{1-2}, \ldots, E_{1-2}\ P + 1\} = \{x_1, \sin(x_1), \cos(x_1), \sin(3x_1), \cos(3x_1), \ldots, \sin(Px_1), \cos(Px_1)\}$. Each input term is similarly expanded.

The output obtained from the model becomes

$$y = \sum_{i=1}^{i=3} \sum_{j=1}^{j=2P+1} (E_{ij} \,.\, w_{ij}) \qquad \ldots (1)$$

Error, e = d – y, where d is the data succeeding the last data in the dataset.

The weights are adjusted according to the Least Mean Square (Pradhan, Routray & Basak, 2005) update equation

$$\Delta_{wij} = -\frac{de^2}{dw_{ij}} \qquad \ldots (2)$$

$$w_{ij} = w_{ij} + \mu\Delta w_{ij} \qquad \text{... (3)}$$

where, μ is a learning parameter which is adjusted suitably between 0 and 1 so that the best possible convergence is achieved.

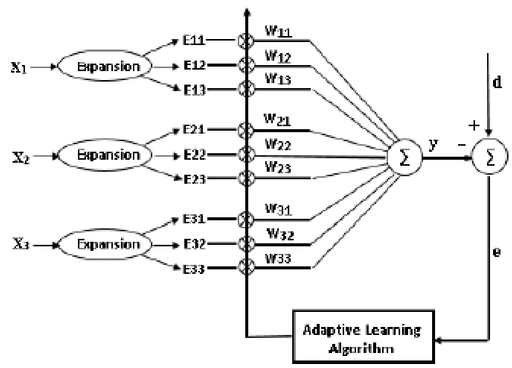

Figure 1: Functional Link Artificial Neural Network Based Prediction Model

Multi-Layered Perceptron

As an Adaptive Predictor, the Multi-Layered Perceptron (MLP) uses sigmoid functions as its base function.

The logistic function has been used as the sigmoid activation function for the present case. The choice is governed by the fact that output of the model is expected to be positive, and the normalised inputs lie well in the activation range. Two-sided sigmoid functions, whose output also lies in the negative region, have been avoided on grounds of positive-only output values.

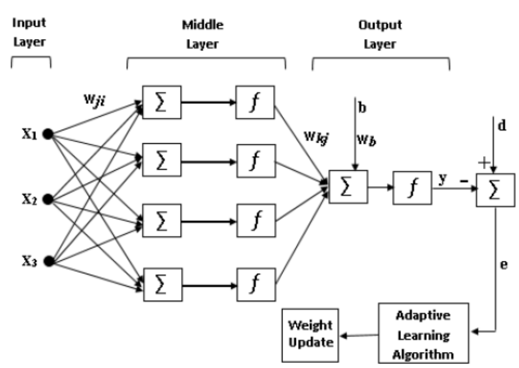

Figure 2: Multi-Layered Perceptron Based Prediction Model

The equation for the function is:

$$f(z) = \frac{1}{1+e^{-z}} \quad \text{... (4)}$$

The first layer of a three layered MLP shown in Fig. 2 consists of three inputs which correspond to the features extracted from NAV data. The middle layer produces a non-linear response given by (4) and is computed as weighted sum of first layer data as the input for the sigmoid function.

$$y_j = f\left(\sum_{i=1}^{3}\sum_{j=1}^{n} x_i . w_{ji}\right) \quad \text{... (5)}$$

The final layer, similarly, uses the weighted sum of the output from the middle layer, combined with a weighted bias function, which is 1, as an input to the sigmoid function to get a non-linear response. The final output from the model is given as

$$y = f\left(\sum_{j=1}^{n}(y_j . w_{kj}) + (w_b . b)\right) \quad \text{... (6)}$$

The weights of the MLP prediction model are updated according to (7) and (8)

$$\Delta w_{ij} = \frac{de}{dw_{ij}}, \ \Delta w_b = \frac{de}{dw_b} \quad \text{... (7)}$$

$$w_{ij} = w_{ij} + \mu\Delta w_{ij}, wb = wb + \mu_b\Delta w_b, \quad \text{... (8)}$$

where μ_b and μ are constants whose values lie between 0 and 1 and in the present case is adjusted by trial and error to give the best training performance. The error term, e = y – d, is computed and used for updating the weights.

Radial Basis Function Network

As an Adaptive Predictor, the Radial Basis Function (RBF) Network uses Gaussian functions as the base functions to achieve nonlinear relation between input and output data of the predictor. Sums of radial basis functions are typically used to approximate given functions. This approximation process can also be interpreted as a simple kind of neural network.

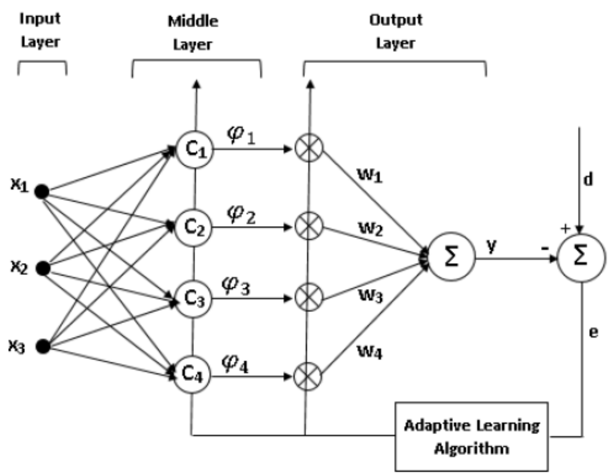

Figure 3: Radial Basis Function Network Based Prediction Model

The function used in the present case is

$$f(z) = \frac{e^{-z^2}}{2\sigma^2}, \quad \text{... (9)}$$

where σ is the spread of the function.

The input represents the features extracted from the data. It forms an N-dimensional input vector. The hidden layer consists of centres of N-dimensions each of which produces a response based on the radial distance between the centre and the input vector.

The radial distance of the input vector from the ith centre is given by

$$z_i = ||x - c_i|| = \sqrt{\sum_{j=1}^{N} (x_j - c_{ij})^2} \quad \text{... (10)}$$

N-dimensional centre, $c_i = [c_{i1}, c_{i2}, \ldots, c_{in}]$,

N-dimensional input, $x = [x_1, x_2, \ldots, x_n]$

The non-linear response from the i^{th} centre is given as

$$\varphi(i) = f(z_i) \quad \text{... (11)}$$

where $f(z_i)$ is defined in (9).

The final output is a weighted sum of the non-linear response

$$y = \sum_{i=1}^{h} (\varphi(i) \, . \, w_i) \quad \text{... (12)}$$

Error, e = y – d, where d is the data after the last data in the training set.

The centres and weights are updated based on the gradient descent method given in (13)-(15).

$$\Delta c_{ij} = -\eta_1 \frac{\partial e}{\partial c_{ij}} \qquad \text{... (13)}$$

$$\Delta_{wi} = -\eta_2 \frac{\partial e}{\partial c_i} \qquad \text{... (14)}$$

$$c_{ij} = c_{ij} + \Delta c_{ij},\ w_i = w_i + \Delta w_i, \qquad \text{... (15)}$$

where, $\eta_1$and η_2 are arbitrary constants with values ranging between 0 and 1.

Methodology

The NAV values of the HDFC Top 200 Mutual Fund was collected for 300 consecutive trading days, from 15-Oct-2012 till 2-Jan-2014, respectively. The data is normalized, with the maximum value at 0.9.

Mean and variance are the features which are subsequently extracted from the dataset to be applied as inputs to the models for training. A set of 10 consecutive data is taken at a time from the above obtained dataset, starting from the first data, corresponding to 15-Oct-2012. The extracted features, along with the last data in the set, are fed as inputs to the respective models, whose outputs are then expected to predict the data for the next trading day.

The difference the obtained outputs from the respective models and the actual value for the next day is the error in the prediction, and is used for updating the parameters of the respective models using (2), (3), (7), (8), (13), (14) and (15).

The next pattern of inputs is formed by taking the next day's data and removing the first data from the previous set. A total of 291 patterns are formed for training and testing purposes. Out of the 291 patterns of features, about 80% are used for training various prediction models and the rest are used for testing purposes.

An epoch based learning scheme is used for updating the weights of all the models. This involves application of the 233 of the 291 training patterns to the models and storing the respective weight changes for each applied set. Afterwards, the average of the 233 weight changes obtained is computed and the weight of each of the models is updated by adding this average weight change calculated.

The above steps constitute one step of the iteration process, which is subsequently repeated for several iterations until the mean square error attains the minimum value.

Weight change for a particular iteration is given as

$$\Delta w_i = \frac{\sum_{p=1}^{n} \Delta w_{ip}}{n}, \qquad \text{... (16)}$$

where, n = number of training patterns applied.

The trained models, now, are used to predict the data not used earlier for training, and their respective prediction errors are recorded. The heuristic parameters of the models are now varied and the above training process is repeated again. The parameters yielding the lowest percentage error amongst the trial values are used for the final prediction of the NAV values for a 15 day window.

Results

Simulation studies are made to find out the best set of parameters for each model. In case of FLANN model, nine trigonometric expansions provides the best possible prediction performance. The same is achieved by considering five middle layers in case of MLP (Fig. 4) and five

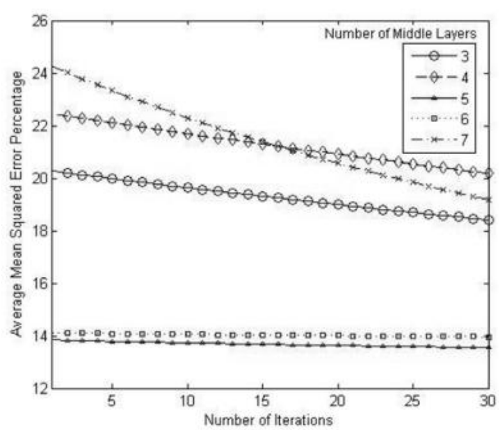

Figure 4: Comparison of Convergence Characteristics of MLP Using Different Number of Hidden Layers

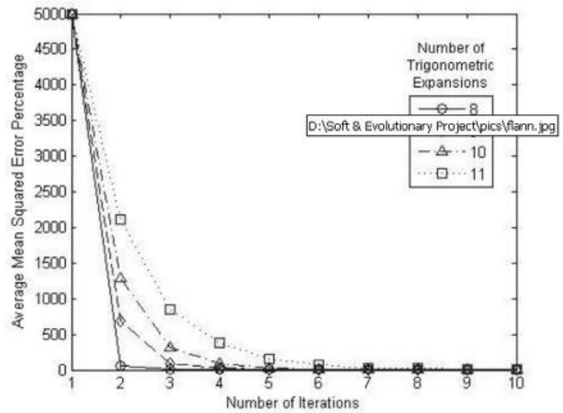

Figure 5: Comparison of Convergence Characteristics for FLANN Predictor for Different Number of Trigonometric Expansions

centres in case of RBF (Fig. 6) by using Gaussian basis function. The relative performance of a particular model for different testing patterns is unaffected by using other heuristic parameters in the model.

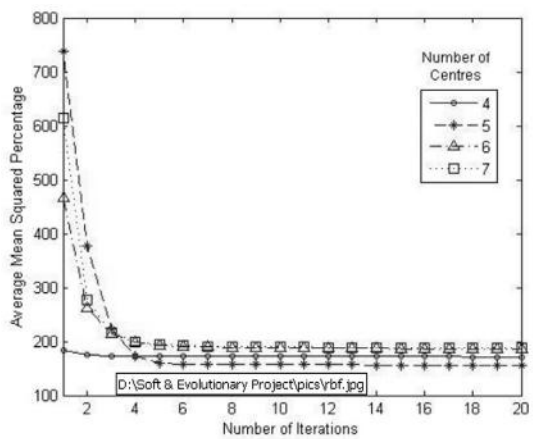

Figure 6: Comparison of Convergence Characteristics for RBF Predictor for Different Number of Centres

The trained models are used to predict 15 days' ahead NAV values. Table 2 shows the performance of the models for fifteen working days' ahead prediction using optimized parameters and 3000 iterations each. The set of parameters used in the simulation study for the models are shown in Table 1.

Table 3 shows the performance of the models for seven as well as fifteen days' ahead prediction.

The NAV predicted by the models have been plotted alongside the actual values (Fig. 7) after denormalisation.

The FLANN model closely predicts the true NAV values. However, the RBF model provides lesser errors for short term prediction, with accuracy decreasing with prediction time. But it does not replicate the shape of the original NAV series. The prediction error is higher in case of the MLP model.

Table 1: Set of Heuristic Constants Used for the Prediction Models

Functional Link Artificial Neural Network	*Multi-Layered Perceptron*	*Radial Basis Function Network*
$\mu = 0.05$	$\mu = \mu b = 0.25$	$\sigma = 0.9,\ \eta_1 = \eta_2 = 0.45$

Table 2: NAV Prediction Error Using the Three Different Models

Days Ahead	*Actual Data*	*FLANN*		*MLP*		*RBFN*	
		Predicted Data	*Error %*	*Predicted Data*	*Error %*	*Predicted Data*	*Error %*
1	231.489	228.781	1.192	245.621	6.105	234.011	1.089
2	231.541	228.789	0.951	245.153	5.879	233.404	0.804
3	230.988	228.485	0.211	242.655	5.051	230.920	-0.029
4	228.969	227.291	0.769	240.623	5.090	229.096	0.055
5	229.053	227.401	0.131	239.424	4.528	228.172	-0.384
6	227.701	226.599	1.879	241.071	5.872	230.346	1.161
7	230.94	228.687	0.673	243.833	5.583	233.618	1.159
8	230.238	228.291	1.837	245.272	6.530	235.396	2.240
9	232.564	229.715	1.083	247.436	6.395	237.928	2.306
10	232.232	229.542	0.142	244.853	5.435	235.436	1.379
11	229.869	228.118	1.585	244.173	6.223	234.750	2.123
12	231.793	229.291	1.728	247.638	6.836	238.414	2.856
13	233.324	230.172	1.804	250.267	7.262	241.065	3.317
14	234.403	230.745	1.468	251.216	7.173	241.742	3.130
15	234.185	230.580	-0.255	247.015	5.479	237.003	1.203

Table 3: Average Performamce of the Three Models for 7-Day and 15-Day Span

Mean Absolute Error %	*FLANN*	*MLP*	*RBFN*
7 Days	0.831%	5.444%	0.668%
15 Days	1.0132%	5.962%	1.549%

Conclusion

The FLANN model predicts very closely the actual NAV values. However, the RBF model provides the best prediction results for short term prediction of one week. It has lesser error but does not replicate the shape of the original NAV curve better than the FLANN model. It provides the best performance among the three. While for long term prediction of 15 days and above, the FLANN model predicts better on an average, while the RBF model tends to lose accuracy with time. Overall, the FLANN simulation results based on real life data demonstrates better prediction consistency among the three models.

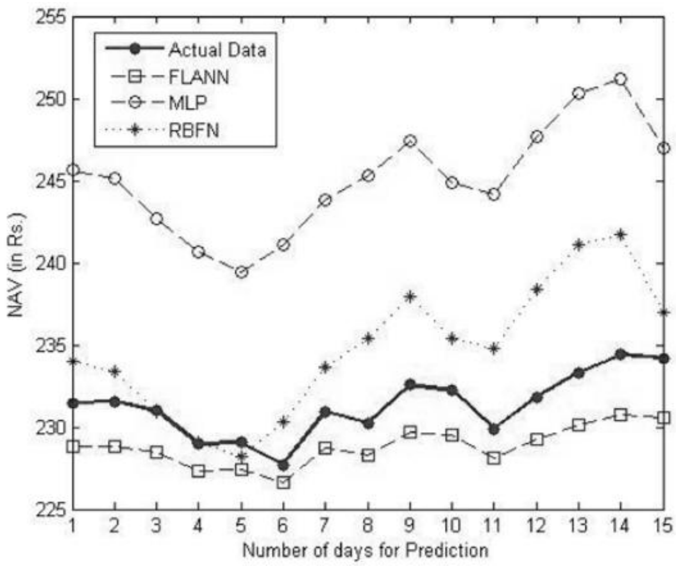

Figure 7: Comparison of Actual NAV with that Predicted by the Three Different Prediction Models

References

Chakraborty, K., Mehrotra, K., Mohan, C. K., Ranka, S., "Forecasting the Behaviour of Multivariate Time Series Using Neural Networks", 1992. DOI: 10.1016/S0893-6080(05)80092-9.

Freisleben, B., Ripper, K., "Economic Forecasting Using Neural Networks", IEEE, *International Conference on Neural Networks*, 1995.

Gao, Hui, and Cherkassky Vladimir, "Real-Time Pricing of Mutual Funds", *IEEE, International Joint Conference on Neural Networks*, 2006. DOI: 10.1109/IJCNN.2006.247065.

Panda, Ganapati and Das, Debi Prasad, "Functional Link Artificial Neural Network for Active Control of Nonlinear Noise Processes", *International Workshop on Acoustic Echo and Noise Control*, 2003.

Panda, Ganapati, and Majhi, Babita, "Forecasting of Currency Exchange Rates using an adaptive ARMA model with Differential Evolution based technique", *Journal of King Saud University, Computer and Information Sciences*, 2013. DOI: 10.1016/j.jksuci.2013.01.002.

Panda, Ganapati, and Majhi, Babita, "Forecasting of Retail Sales Data Using Differential Evolution", *IEEE, World Congress on Nature and Biologically Inspired Computing*, 2009. DOI: 10.1109/NABIC.2009.5393740.

Pradhan, A.K., Routray, A and Basak, A, "Power system frequency estimation using Least Mean Square Technique", *Power Delivery, IEEE Transactions On (Volume 20, Issue: 3*), July 2005. DOI: 10.1109/TPWRD.2004.843453.

Priyadarshini E., Chandra Babu A., "Prediction of the Net Asset Values of Indian Mutual Funds using Auto-Regressive Integrated Moving Average", *Journal of Computer Applications Research and Development*, 2011.