Journal of Engineering Science and Technology Review 17 (2) (2024) 97 – 118

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Exploring Variants of Extreme Learning Machines for Prediction of Mutual Fund NAV

Smruti Rekha Das[1,*], Debahuti Mishra[2] and Minakhi Rout[3]

[1]*Department of Computer Science and Engineering, GITAM Deemed to be University, Visakhapatnam, Andhra Pradesh, India*
[2]*Department of Computer Science and Engineering, Siksha 'O'Anusandhan Deemed to be University, Bhubaneswar, Odisha, India*
[3]*School of Computer Engineering, KIIT Deemed to be University, Bhuabaneswar, Odisha, India*

Received 6 October 2023; Accepted 25 March 2024

Abstract

Investing money through mutual fund benefits the small investors to access equities of big companies with a small amount of capital. It experiences the fluctuation of price along with the performance of stock, which is a major part in making the fund. Here, in this paper variant of Extreme Learning Machines (ELM) are applied to forecast the end-of-year net asset value (NAV) of mutual fund. Various types of ELM such as basic ELM, evolutionary ELM, online sequential ELM and error minimized ELM are explored and applied to historical data of four mutual funds such as SBI mutual fund, UTI mutual fund, Tata Mutual Fund and Kotak Mahindra Mutual Fund for the prediction of NAV. Along with the different ELM based prediction model, this paper has explored on different types of activation functions and the number of nodes in the hidden layer used in variants of ELM. Examining the simulation result of all the models, along with different activation functions and different number of nodes, it is observed that evolutionary ELM outperforms over the other variants of ELM used in this study.

Keywords: Mutual Fund; Extreme Learning Machine; Online Sequential Extreme Learning Machine; Evolutionary ELM.

1. Introduction

Mutual fund is a beneficiary investment scheme managed by professional money manager, where a pool of money is collected from many investors and the money is invested in securities such as stocks, bonds, money market and various other assets. It is considered to be a diversified investment, where the risk can be reduced as the money is not invested in a particular securities rather diversified to various securities, so that fall of equity of one company can be adjusted by the rise of another equity. But still we can't tell that the diversification reduces the risk, as the mutual fund is associated with equity market which is very fluctuating, hence, risk remains there. Many researches have been done for the prediction of NAV of mutual fund. Researchers have explored so many application methods for the prediction of NAV. WC Chiang *et al.* introduced Artificial Neural Network (ANN) for NAV prediction and compared the [1] result with regression model, where ANN proves about its better prediction accuracy over regression model. Other than NAV prediction some researchers have made analysis over investment ability. Here, in a research work, ANN is used by DC Indroa *et al.* to predict the performance of equity [2] mutual funds following value, blend and growth investment. H Mamaysky *et al.* analyzed over the forecasting of alpha and beta using kalman filter and in his study, the author proves the better performance [3] of Kalman model over ordinary least squares timing models. Neural network has widely being used for the prediction of financial market such as stock market, forex market, mutual fund etc. H Yan *et al. e*mployed back-propagation neural network [4] for the prediction of NAV, and shows its good learning ability. The importance of neural network family for the prediction financial market is delineated by C. M. Anish *et al.,* where the author proposed feedback functional link artificial neural network (FFLANN) for the NAV prediction of Indian [5] mutual funds and compared the result with multi layer artificial neural network (MLANN) and functional link artificial neural network (FLANN). Using the performance measure of root mean square error (RMSE) and mean absolute percentage error (MAPE), FFLANN clearly proves its efficiency over other two models. After analyzing the pros and cons of various prediction model C. M. Manish *et al.* proposed an ensemble model, which is a combination of Radial Basis Function (RBF) and FLANN, where the summation of weighted outputs of both the models is [6] compared with the target output. The ensemble model proves to be a better model than the individual model through the performance measure scale RMSE and MAPE. Again author employed three adaptive models such as adaptive moving average (AMA), adaptive auto regressive moving average (AARMA) and feedback radial basis function network (FRBF) and the output of three individual models [7] is weighted optimally using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The final prediction result of NAV obtained from the above ensemble based adaptive model is better compared to the three individual models. TH Huang *et al.* analyzed over the profitable portfolio of mutual funds, which will be applicable for the investors. For this the model divided into two stages, where, in the first stage selection of mutual fund occurred base on the [8] DEA, Sharpe and Treynor indices and in the second stage linear regression model, Fruitfly Optimization Algorithm (FOA) and General Regression Neural Network (GRNN) models are applied to generate a NAV prediction model. The basic aim of this model is to predict the NAV of constituent mutual funds of the portfolio. The experimental result showed that the combination of Sharp index with

*E-mail address: sdas5@gitam.edu
ISSN: 1791-2377
doi:10.25103/jestr.172.12

GRNN optimized with FAO provided best rate of return. TH Huang *et al.* emphasized over the various economic factors, which have a great impact on NAV [9] of Indian Mutual funds. Here, author used two methods such as regression analysis and ANN, whose performance is compared using MAPE and RMSE.

To overcome the basic drawbacks of feed forward neural network such as extensively used of gradient based learning algorithm and iterative tuning of network parameters; Extreme Learning Machine (ELM) is introduced by G. B.Huang, which [10-13] randomly chooses the nodes in the hidden layer and analytically generate the output weights. The soul of ELM is that the parameters need not to be tuned each time and the generalization performance of ELM is also much better than traditional computational intelligence technique with less human interference. ELM not only tends to reach the smallest training error but also the smallest norm of weights. The applications of ELM are studied to a large extent, during the past decade. To make it more efficient various extensions have been made to the original ELM and simultaneously variants of ELM is designed to meet the requirement of specific applications. As ELM chooses random input weights and biases, so there is a chance of creating non optimal solutions. QY Zhu *et al.* proposed evolutionary ELM, where differential [14] evolutionary algorithm used to carefully choose the best input weights to obtain the optimal solution.

The research gap lies in the absence of a comprehensive exploration of Extreme Learning Machines (ELMs) specifically within mutual fund NAV prediction, despite their application in financial forecasting tasks. Our study addresses this gap by offering a detailed investigation into the effectiveness of ELMs for predicting mutual fund NAVs, enhancing understanding and providing practical insights for financial practitioners.

The motivation behind this study is to explore the prediction ability of ELM. The basic ELM is having the issue of getting non optimal solution due to choosing of random input of weights and biases and another issue in ELM is that it tends to require more number of hidden nodes than the conventional method. To overcome all these short comings of ELM, variants of ELM is proposed by some researchers. Another essential part of exploring on ELM is that, after the wide acceptance of ELM in various fields of application, the variants of ELM are developed by G. B.Huang. The employment of variants of ELM for the prediction of NAV of mutual fund is analyzed in this study.

In this paper, an empirical comparison has carried out among different variants of ELM such as basic ELM, Evolutionary ELM, OSELM (Online Sequential Extreme Learning Machine) and error minimized ELM, using NAV of different mutual funds. For experimental purpose SBI Magnum Equity, UTI Equity mutual fund, TATA Dividend Yield Fund Direct-Growth and Kotak Mahindra mutual fund has considered. The datasets are regenerated using *mean*, *standard deviation*, *kurtosis* and *skewness* as different statistical measures. In addition to this another two aspects of variants of ELM has explored such as number of nodes in the hidden layer and different activation functions used in hidden layer. Further, the MSE in training and the result of different performance measures in testing are analyzed to evaluate the efficiency of the models.

The rest of the work is organized as follows: section 2 contains the analysis of variants of ELM with algorithms, experimental result analysis is provided in section 3, performance verifications and discussions are described in section 4 and section 5 respectively, finally section 6 draws the conclusion part.

2. Analysis of variants of ELM

In this section basic ELM along with its variants such as OSELM, evolutionary ELM, and error minimized ELM are introduced. In recent years, researchers have shown their more and more interest on ELM, and to improve the performance of ELM various extension of ELM is proposed.

2.1 Basic ELM

ELM is a learning neural algorithm, introduced to develop the efficiency of Single Layer Feed Forward Neural Network (SLFN) but unlike SLFN, It is a tuning free algorithm and work much faster than traditional approach of neural network. The beauty of the ELM is that, by using some mathematical transformation, the output weights is calculated analytically, which avoid the lengthy process of training and simultaneously the training parameter need not to be adjusted iteratively. The working principle of ELM is explained briefly in this section [15-16]. Stepwise representation of basic ELM algorithm for mutual fund prediction is given below.

Algorithm 1: Basic ELM

Begin

Dataset contains (*NAV price*, *Mean*, *Standard deviation*, *Skewness*, *Kurtosis*); Size of hidden layer (*M*). Input_data (I') are the input to the basic ELM and the output is the predicted NAV.

Step 1: Divide the dataset to training (*train_input*, *train_output*) and testing (*test_input*, *test_output*) in 7:3 ratio.

Step 2: a_i and b_i are the randomly generated input weights and bias of the hidden node

Step 3: For *train_input* find

find

$$H = \begin{bmatrix} f(a_1.x_1 + b_1) & \cdots & f(a_M.x_1 + b_M) \\ \vdots & \ddots & \vdots \\ f(a_1.x_N + b_1) & \ldots & f(a_M.x_N + b_M) \end{bmatrix}_{N\times M},$$

where *N* is the size of train_input

Step 4: $\beta = inv(H' \times H) \times H' \times train\_output$

Step 5: find H_1 using the step 2 for *test_input*

Step 6: $obt\_output = H_1 \times \beta$

Step 7: obt_output is the predicted NAV price for the *test_input*

Step 8: *Error* = $MSE(obt\_output, test\_output)$

End

2.2 Evolutionary ELM

Since the input weights and biases are chosen randomly in ELM, which is [14] responsible for the creation of non optimal solution. ELM also requires more number of nodes in the hidden layer than conventional algorithm. To get the optimal solution the input weight is optimized with an evolutionary algorithm. Here, in this study Differential Evolution (DE) is considered for weight optimization. DE is having very few controlling parameters such as *crossover rate* and *mutation scale factor*, whose value has a great importance in the performance of the algorithm. It's a very simple algorithm and straight forward to implement. DE algorithm works through cycle of four stages such as initialization of the population of search variable vectors, mutation, crossover and selection [17]. The algorithm of ELM-DE for analysis of mutual fund prediction is given step wise.

Algorithm 2: Evolutionary ELM

Begin

Dataset containing (*NAV*, *Mean*, *Standard deviation*, *Skewness*, *Kurtosis*); Population size (*K*); Size of hidden layer (N_c); Crossover rate (*Cr*); Mutation scale factor (Mf). Input_data (I') are the input to the evolutionary ELM and the output is the predicted NAV price.

Step 1: Divide the dataset to train_set (*train_input*, *train_output*) and test_set (*test_input*, *test_output*) in 7:3 ratio.

Step 2: Set *K* random weight population; each of size $1 \times N_c$

$P_i = \{V_1^i, V_2^i, V_3^i, \dots\dots\dots.., V_{N_c}^i\}$

for $i = 1,2,3, \dots\dots, K$

Step 3: For each population P_i find the error value in ELM by step 5 to 9

Step 4: For each *train_input* find

Step 5: $H_l = train\_input_l \times P_i$

Step 6: $\beta_i = pseudo\_inverser(H) \times train\_output$

Step 7: $obt\_output = (test\_input \times P_i) \times \beta_i$

Step 8: $O_i = MSE(obt\_output, test\_output)$

Step 9: Find global_best P_i where $arg_{min}(O_i)$

Step 10: Until the stopping criteria is satisfied continue the loop

Step 11: For each population P_i

Step 11.1: Mutation step

A donor vector D_i is generated corresponding to the i^{th} target vector

$D_i = P_i + Mf \times (Global_{best} - P_i)$

Step 11.2: Cross over step

A trial vector is generated for the target vector

$T_i = D_i$, $if\ (rand(\,) < Cr)$ else $T_i = P_i$

Step 11.3: Selection step

Trial vector T_i is evaluated for selection considering two stages such as parent selection and survivor selection

If $O_i\,(T_i) \le O_i(P_i)\ then\ T_i$ will go to the next generation else P_i will go to the next generation.

Step 12: Repeat step 11 until stopping criteria satisfies

Step 13: global_best is considered as final weight w_{global_best} and store the corresponding β_{global_best}

Step 14: For unknown Input_data (I') find the NAV price by following equation

Predicted NAV $= (I' \times w_{global\_best}) \times \beta_{global\_best}$

End

2.3 OSELM

OSELM is the sequential modification of ELM, where the model learns one by one or chunk by chunk with a fixed or varying chunk size [18-21]. The parameter of the hidden layer nodes is selected randomly, accordingly the input weight weights and bias are randomly generated and the output weights are analytically created by using some mathematical transformation. Optimal number of hidden layer nodes should be chosen so that lowest validation error will be provided by the network. It is same as batch ELM and after training the data one by one or chunk by chunk, that particular data or chunk is discarded once the learning procedure [22] of that chunk is complete. The working principle of OSELM [23] is described stepwise in this study.

Algorithm 3: OSELM

Begin

Dataset containing (*NAV*, *Mean*, *Standard deviation*, *Skewness*, *Kurtosis*); Size of hidden layer (L); Batch size (B). Input_data (I') are the input to the OSELM and the output is the predicted NAV price.

Step 1: Divide the dataset to training (*train_input*, *train_output*) and testing (*test_input*, *test_output*) according to the batch size *B*.

Step 2: a_i and b_i are the randomly generated input weights and bias of the hidden node

Step 3: Find the output matrix H_0

$$H_0 = \begin{bmatrix} G(x_1, w_1, b_1) & \dots & G(x_L, w_L, b_L) \\ \vdots & \ddots & \vdots \\ G(x_{N_0}, w_1, b_1) & \dots & G(x_{N_0}, w_L, b_L) \end{bmatrix}$$

Step 4: Calculate the output weight β^0 by calculating P_o

$P_0 = (H_0^T H_0)^{-1}$

$\beta^0 = P_0 H_0 T_0$, Where, $T_0 = [t_1, \dots, t_{N_0}]$ is the target output and $K = N_0$

Step 5: The whole training samples $(K + 1)^{th}$ training sample is presented considering it as new samples.

$H_{K+1} = [G(x_1, w_1, b_1), \dots, G(x_L, w_L, b_L)]$

Step 6: Calculate the output weight β^{K+1} by calculating P_{K+1}

$P_{K+1} = P_K - P_K H_{K+1}^T (1 + H_{K+1} P_K H_{K+1}^T)^{-1} H_{K+1} P_K$

$\beta^{K+1} = \beta^K + P_{K+1} H_{K+1}^T (T_{K+1} - H_{K+1}\beta^K)$, where, $T_{K+1} = [t_K\ t_{K+1}]$

Step 7: K is incremented by 1 and repeat step 5 to 6 until training of all the batches is completed

Step 8: After training all the samples of OSELM, it is used to predict the unknown input vector

End

2.4 Error Minimized ELM

Error minimized ELM based on ELM is a simple and efficient learning algorithm, basically give emphasis over choosing of number of hidden nodes. Though ELM is a efficient learning approach by avoiding iterative training and descent step but the random choosing of hidden layer nodes effects a lot on the performance of the algorithm. To choose optimal number of hidden nodes by hit and trial method is a big challenge. Hence, Error minimized ELM is introduced to solve this problem, where the hidden nodes grow one by one or group by group with a fixed number or varying number of group sizes [24-26]. By the addition of new hidden nodes the network will change and according to the growth of network the output weights are updated incrementally with significantly reducing the computational complexity. The number of hidden nodes will rise one by one or group by group until optimal number of hidden nodes is obtained.

Algorithm 4: Error Minimized ELM

Begin

Dataset containing (*NAV*, *Mean*, *Standard deviation*, *Skewness*, *Kurtosis*); Maximum number of hidden nodes (HL_{max}); Expected learning accuracy (ϵ). Input_data (I') are the input to the Error Minimized ELM and the output is the predicted NAV price.

Step 1: Divide the dataset to training (*train_input*, *train_output*) and testing (*test_input*, *test_output*) in the ratio of 7:3

Step 2: a_i and b_i are the randomly generated input weights and bias of the hidden node
Step 3: Initially j is assigned to 0, $HL_j = 1$
Step 4: Calculate hidden layer output matrix $H_0 = [g(a_1, b_1, x_1) \dots g(a_1, b_1, x_N)]^T$, where N is the size of the input
Step 5: Corresponding output_error $(H_0) = abs((H_0 \times ((inv(H_0' \times H_0)) \times H_0') \times train\_output) - train\_output)$
Step 6: while $\left((L_j < HL_{max}) \text{ and} \left(E(H_j) > \epsilon\right)\right)$
Step 7: $HL_{j+1} = HL_j + 1$ and $j = j + 1$
Step 8: The corresponding output matrix $H_{j+1} = [H_j \;\; \delta h_j]$, where δh_j is calculated as $\delta h_j = \left[g\left(a_{HL_j+1}, b_{HL_j+1}, x_1\right) \dots g\left(a_{HL_j+1}, b_{HL_j+1}, x_N\right)\right]^T$

Step 9: Update β by calculating D_j and U_j

$$D_j = \frac{\mathrm{inv}((\delta h_j' \times \delta h_j) \times \delta h_j') \times (1 - H_j \times (\mathrm{inv}(H_j' \times H_j) \times H_j'))}{\mathrm{inv}((\delta h_j' \times \delta h_j) \times \delta h_j') \times (1 - H_j \times (\mathrm{inv}(H_j' \times H_j) \times H_j') \times \delta h_j)}$$

$$U_j = \left(\mathrm{inv}\left(H_j' \times H_j\right) \times H_j'\right) - \left((\mathrm{inv}(H_j' \times H_j) \times H_j') \times \delta h_j \times D_j\right)$$

$$\beta^{j+1} = \begin{bmatrix} U_j \\ D_j \end{bmatrix} \times train\_output$$

Step 10: Update the training error $E(H_j)$
Step 11: Increment j by 1 and continue the step 7 to 10 still the stopping condition satisfies
Step 12: **End**

3. Evaluation and Result Analysis

This section covers the experimental work in the form of simulation study using different prediction models applying on different mutual fund datasets and the accessories required for such work like description of the detailed datasets, specification of the parameters of the model, extracting the statistical measures, schematic layout of the proposed model, simulation result analysis of variants of ELM through the actual versus predicted graph and MSE, exploring on different activation functions and number of nodes in the hidden layer.

3.1 Dataset description

Real life data of four different mutual funds such as SBI Magnum Equity, UTI Equity mutual fund, TATA Dividend Yield Fund Direct-Growth and Kotak Mahindra mutual fund are collected. The numbers of data for SBI Magnum Equity, UTI Equity mutual fund, TATA Dividend Yield Fund Direct-Growth and Kotak Mahindra mutual fund are collected during the period 1 march 2007 to 1 march 2017, 1 march 2007 to 1 march 2017, 2nd January 2013 to 1st December 2017 and 1st January 2008 to 1st January 2018 respectively. From the total number of available data total numbers of data patterns are generated considering the running window size of 12. Out of this total number of patterns 70% data is considered for training and 30% is for testing. The number of training and testing patterns generated from total number of patterns for each mutual fund is described in Table 1.

3.2 Specification of Parameters

ELM- It works with the principle of SLFN; hence it is having single hidden layer and the number of nodes considered for hidden layer is explored for producing a good estimation.

Evolutionary ELM- Apart from the number of hidden layer nodes, four other parameters are there for evolutionary ELM as here DE is considered as the evolutionary algorithm for the optimization of ELM. 15 numbers of nodes in the hidden layer is considered here, and the rest parameters for DE are population size, number of iterations, cross over rate and mutation scale factor. 100 are considered for population size as well as number of iterations and the cross over rate and mutation scale are 0.8 and 0.6 respectively. For scaling the difference vector, mutation scale factor is a positive control parameter.

Table 1. Details of mutual fund data available for training and testing purpose

Name of Mutual Fund	Period of Data	Total no. of available data	Total no. of data patterns generated	No. of training patterns	No. of testing patterns
SBI Magnum Equity	1 March 2007 to 1 March 2017	2867	2855	1999	856
UTI Equity mutual fund	1 March 2007 to 1 March 2017	2433	2421	1695	726
TATA Dividend Yield Fund Direct-Growth	2nd January 2013 to 1st December 2017	1203	1192	834	358
Kotak Mahindra mutual fund	1st January 2008 to 1st January 2018	3629	3618	2532	1086

OSELM- OSELM is a learning process which follows batch processing, where the training data fed into the model chunk by chunk. Here, in this study, the chunk size is 30 and the number of nodes in the hidden layer is considered as 15.

Error minimized ELM- It requires only two common controlling parameters such as maximum numbers of nodes in the hidden layer and the expected learning accuracy. This study considered 35 as maximum numbers of hidden layer nodes and the expected learning accuracy is set to 0.001.

3.3 Extraction of statistical measures

The statistical measures has generated considering 12 as the window size containing the present and previous 11 NAV price of mutual fund. The list of statistical measures along with its formula is clearly described in Table 2.

Table 2. List of selected statistical measures and their formulas

Statistical measures	Formulas
Mean	$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$, where N denotes the total number of data
Standard deviation	$\delta = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$, where $\bar{x}$ denotes the mean
Skewness	$Skew = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{x_j - \bar{x}}{\delta}\right]^3$
Kurtosis	$Kurt = \left\{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{x_j - \bar{x}}{\delta}\right]^4\right\} - 3.$

3.4 Layout of Proposed Prediction Model

The block diagram of the experimental work conducted for the prediction of NAV of mutual fund is proposed in **Figure**

1. Variants of ELM are assessed through the ability of prediction performance using four different mutual fund datasets. To compare the performance of the prediction models the simulation is carried out under identical conditions. Four mutual funds such as SBI mutual fund, UTI mutual fund, Tata mutual fund and Kotak Mahindra mutual fund are selected for the experimental purpose and the datasets are regenerated using the statistical measures such as mean, standard deviation, kurtosis and skewness. Through, the NAV of mutual fund, statistical measures are computed using a running window of 12 values, considered as the input to the prediction model. After the preprocessing phase such as normalization the total number of data patterns is divided into training and testing in the ration of 7:3. Normalization of the input data has been made to avoid the saturation problem, which may arise during the use of *sigmoid activation function*. This study has explored on different activation functions. Min-max normalization process has considered for normalization with the range lies between 0 and 1. The mathematical formula for *min-max normalization* is stated here in (1)

$$\tilde{x}^i = \frac{x^i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Where, x^i is the NAV, $\tilde{x}^i$ is the scaled price, x_{min} and x_{max} are the minimum and the maximum value of the particular attribute of the dataset. The corresponding target values to the required number of days ahead for both training and testing patterns is generated. Both training and testing patterns are under gone through the simulation process using ELM, OSELM, Evolutionary ELM and Error minimized ELM. To find out the best prediction model empirical comparisons among the above four prediction models is carried out here in this study.

3.5 Simulation result analysis of variants of ELM

The simulated result of all the mutual funds using all the models such as ELM, OSELM, Evolutionary ELM and Error minimized ELM models for the time horizon of 1 day, 3 days, 5 days, 7 days, 15 days and 30 days is analyzed in this section through the actual versus predicted graph, MSE value in training phase and the improvement of the prediction accuracy.

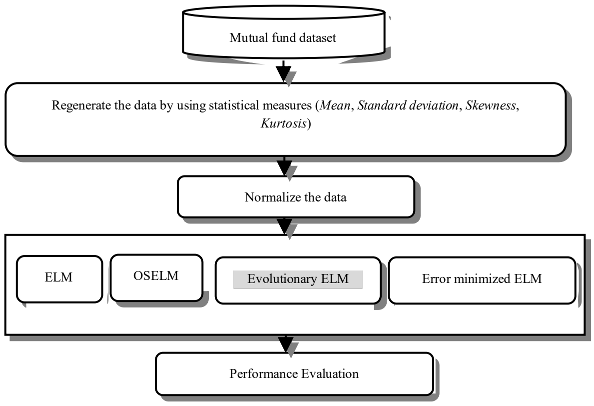
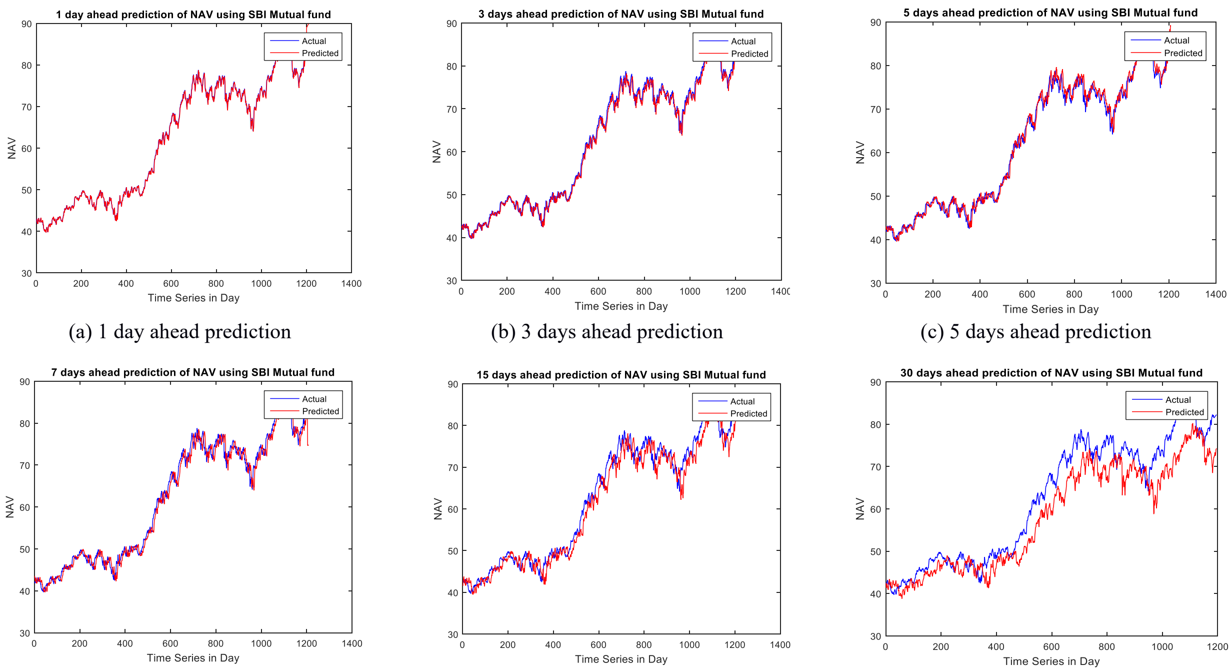
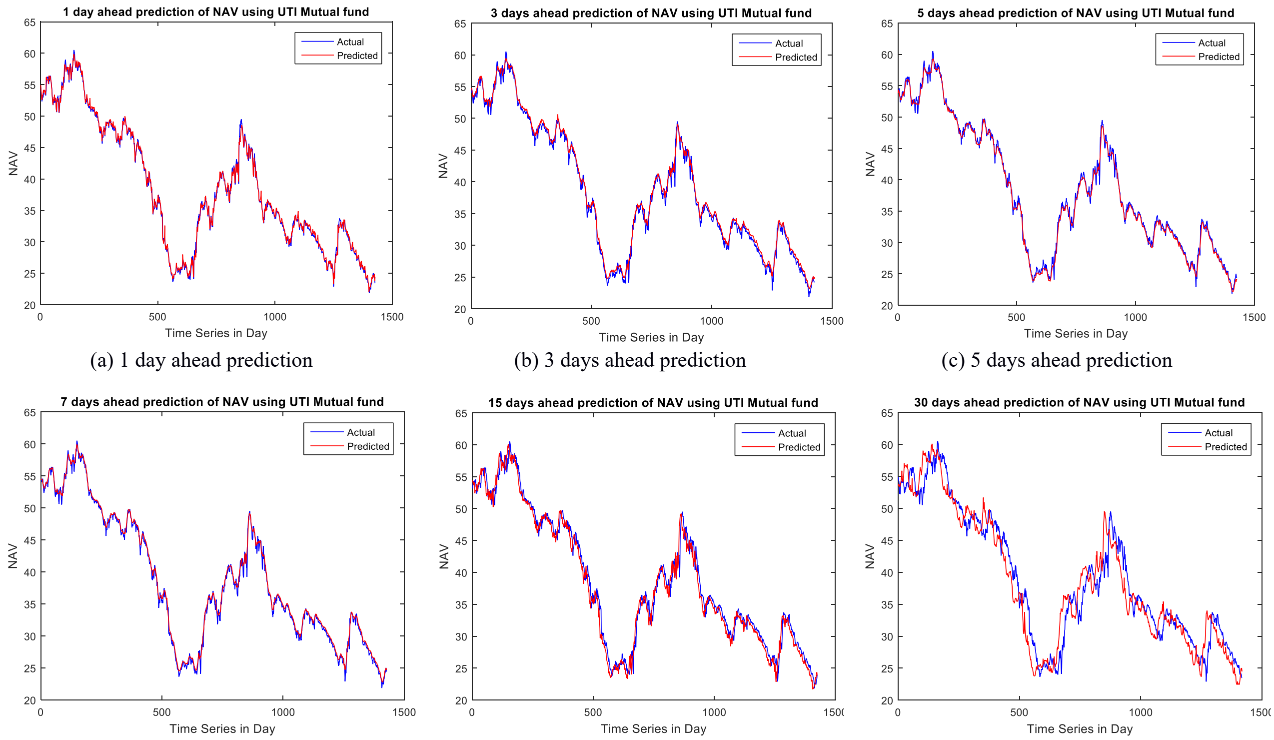


Fig. 1. Architecture of an empirical comparison among the variants of ELM



(a) 1 day ahead prediction
(b) 3 days ahead prediction
(c) 5 days ahead prediction
(d) 7 days ahead prediction
(e) 15 days ahead prediction
(f) 30 days ahead prediction

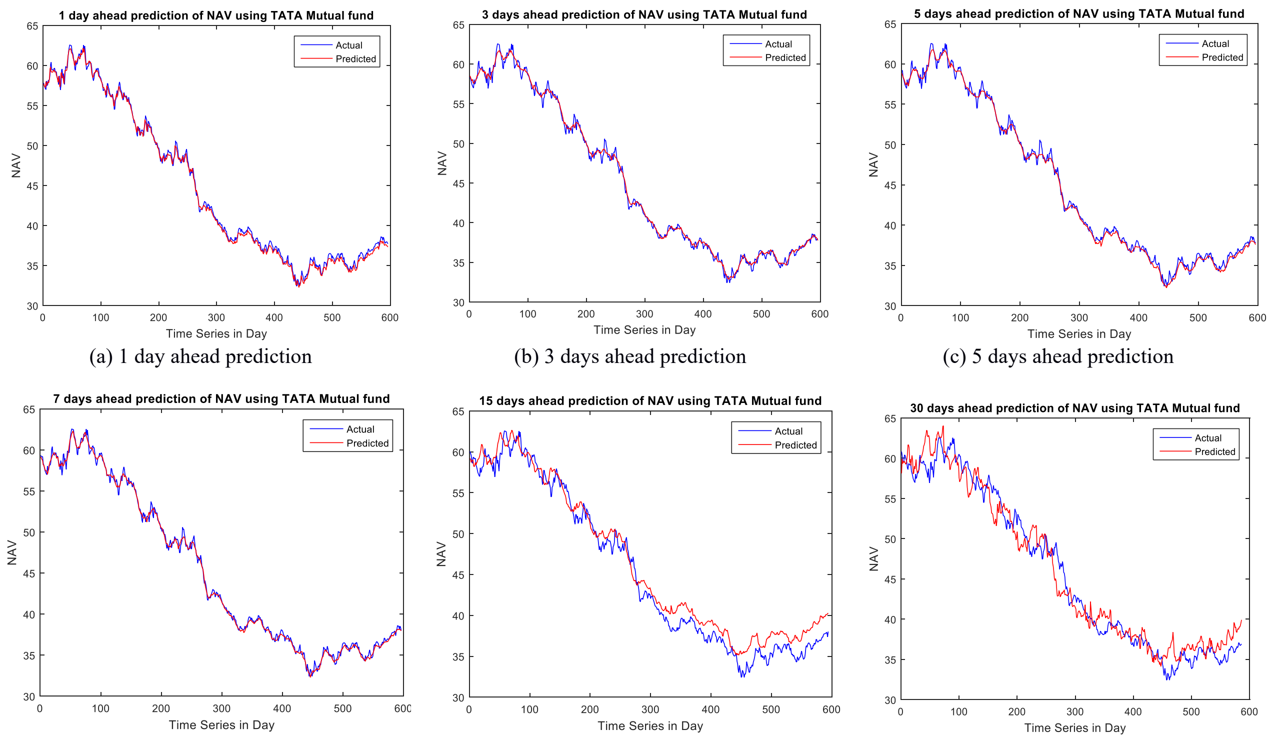
Fig. 2. Simulation graph for ELM for the days ahead prediction using SBI Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

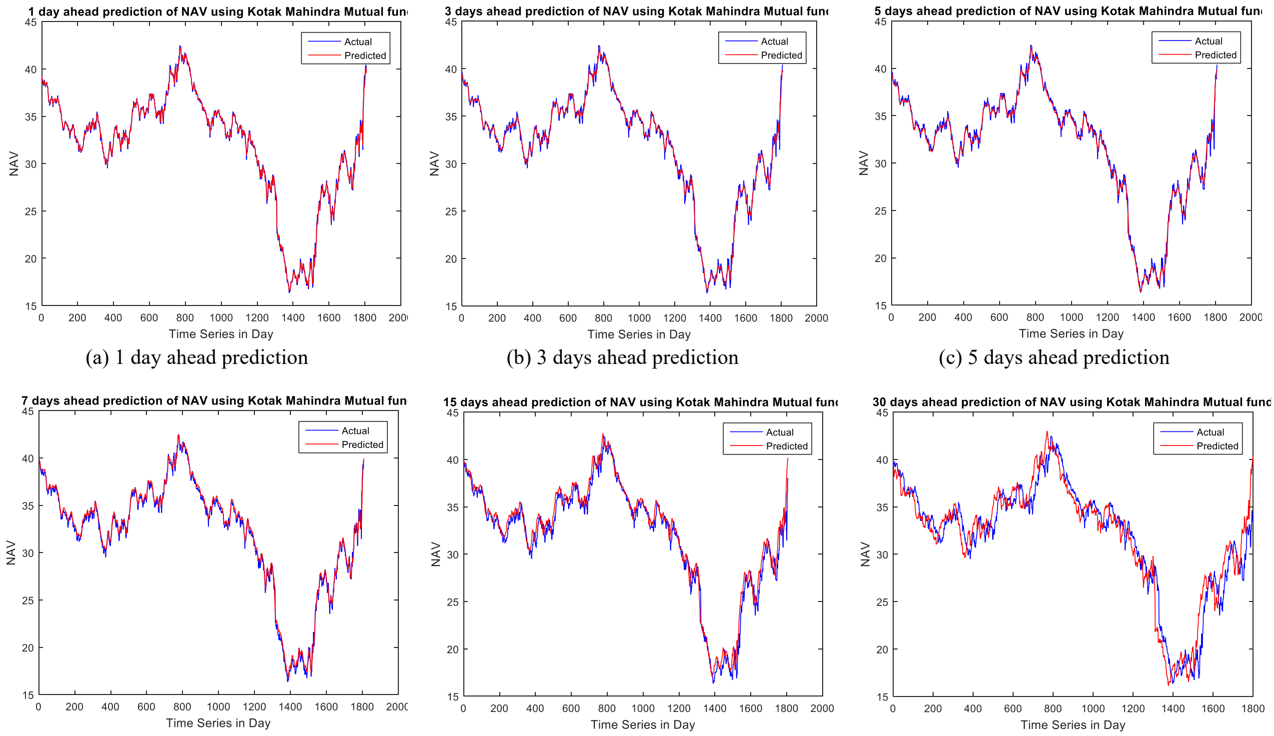
Fig. 3. Simulation graph for ELM for the days ahead prediction using UTI Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

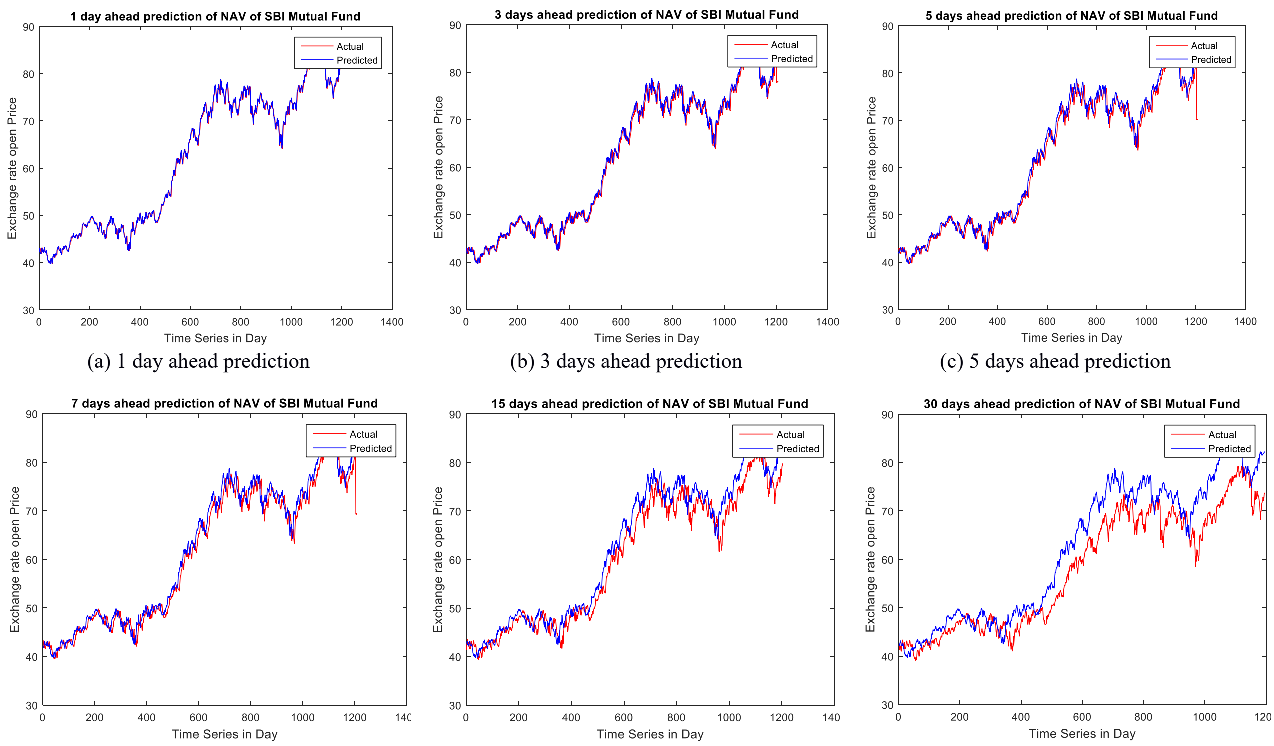
Fig. 4 Simulation graph for ELM for the days ahead prediction using TATA Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

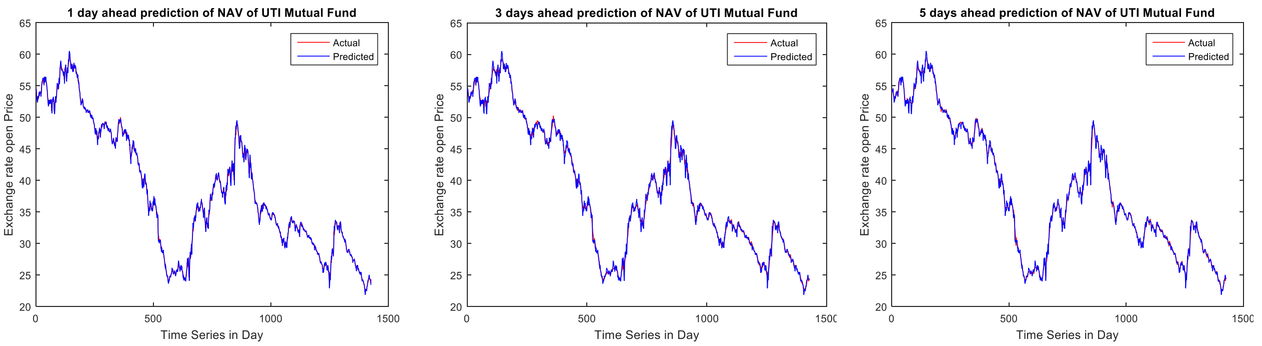
Fig. 5. Simulation graph for ELM for the days ahead prediction using Kotak Mahindra Mutual Fund



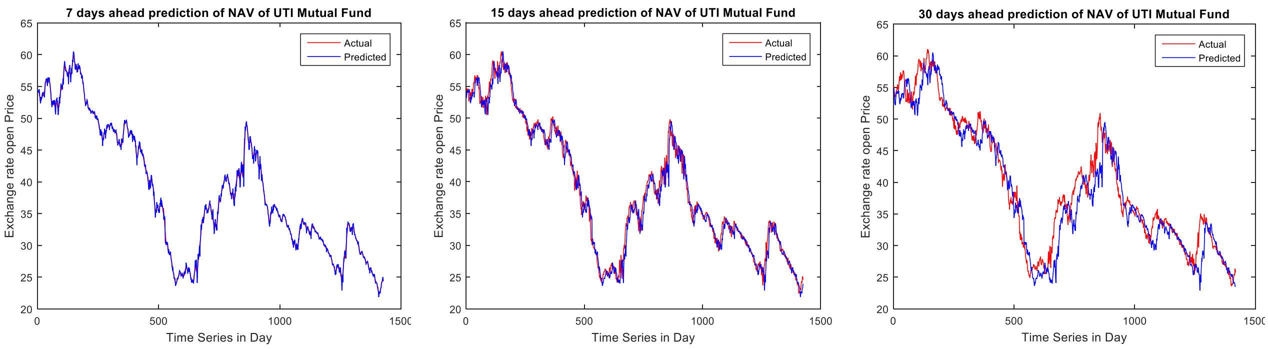
(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

Fig. 6. Simulation graph of OSELM for the days ahead prediction using SBI Mutual Fund

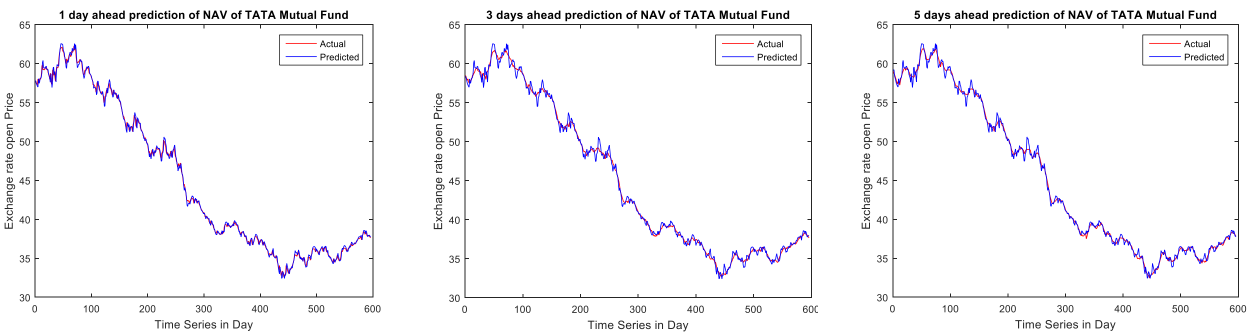


(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

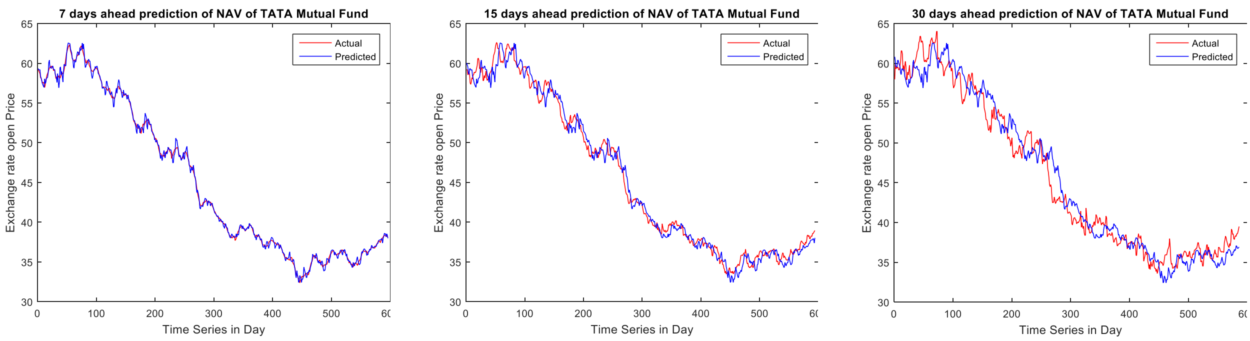


(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

Fig. 7. Simulation graph of OSELM for the days ahead prediction using UTI Mutual Fund

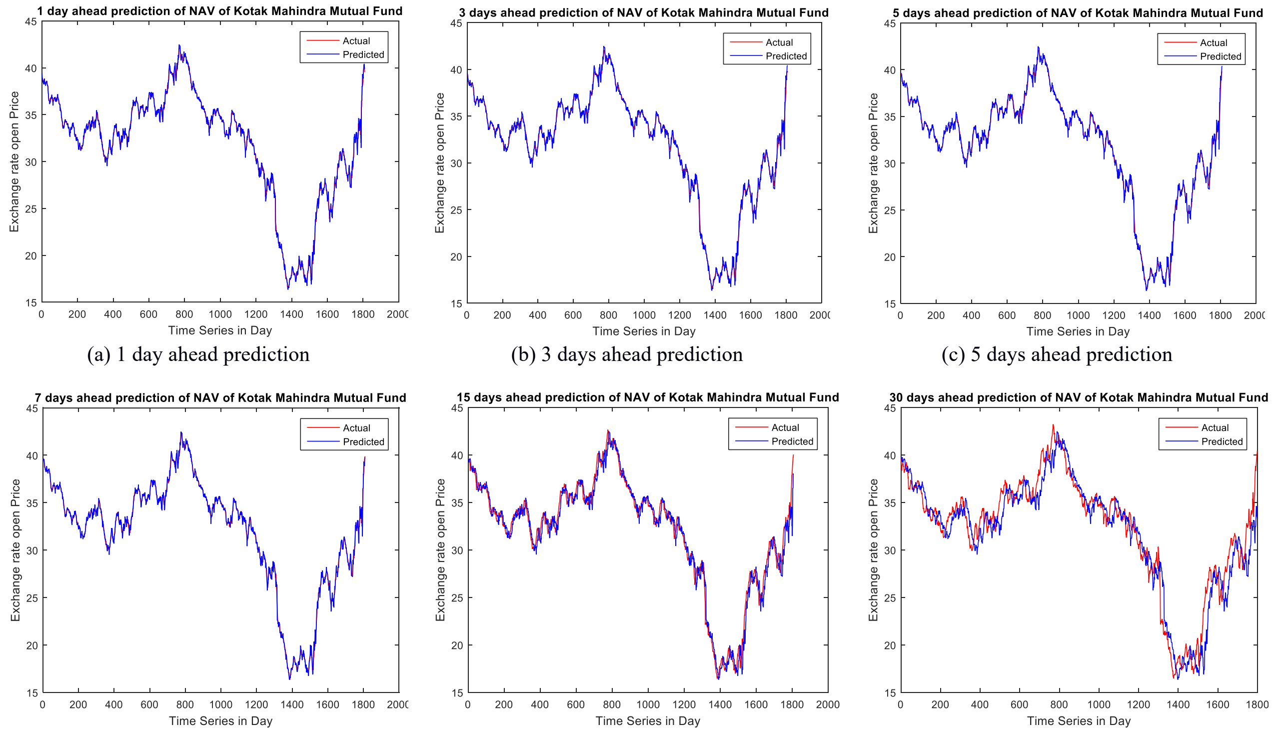


(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction



(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

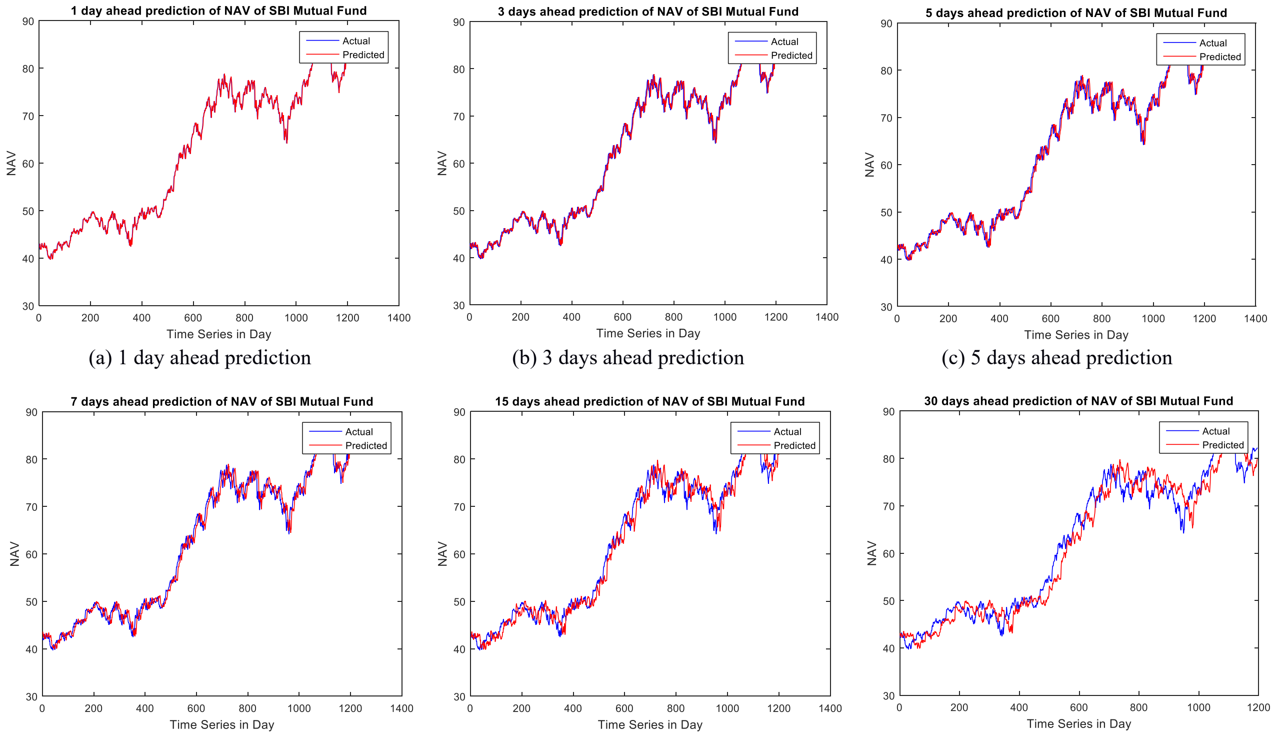
Fig. 8. Simulation graph of OSELM for the days ahead prediction using TATA Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

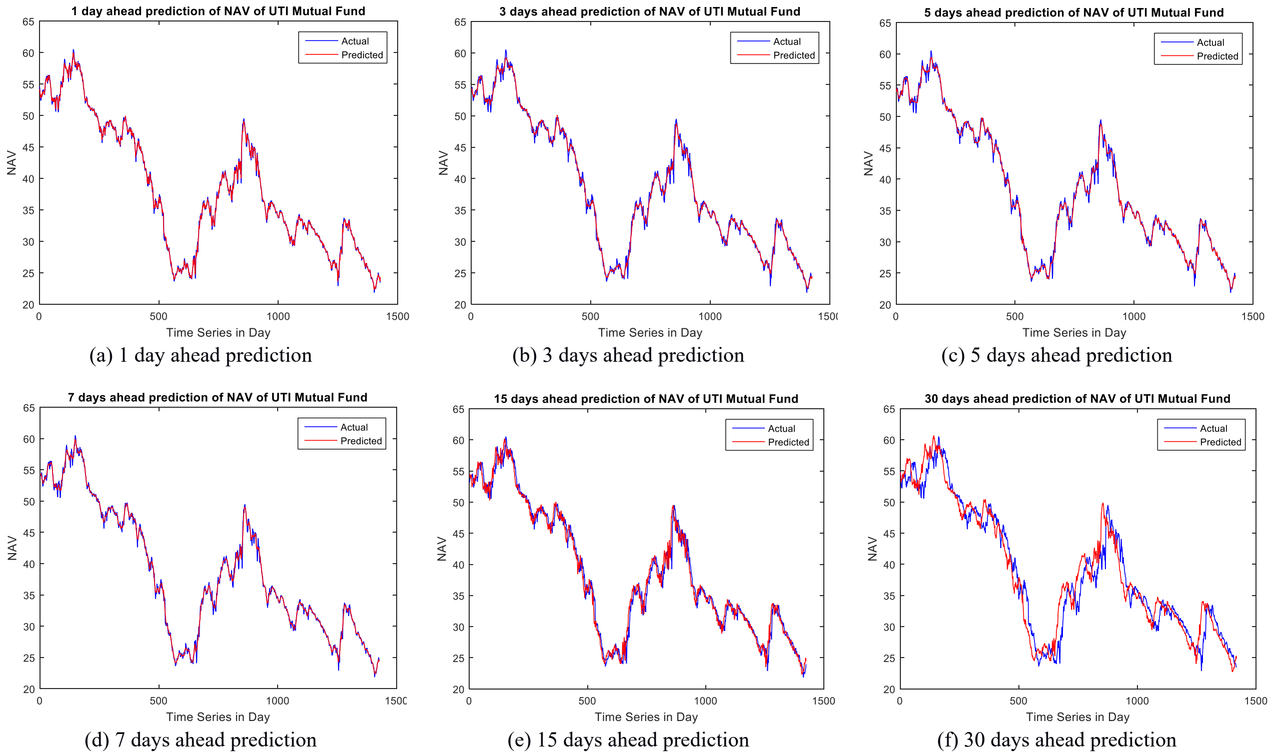
Fig. 9. Simulation graph of OSELM for the days ahead prediction using Kotak Mahindra Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

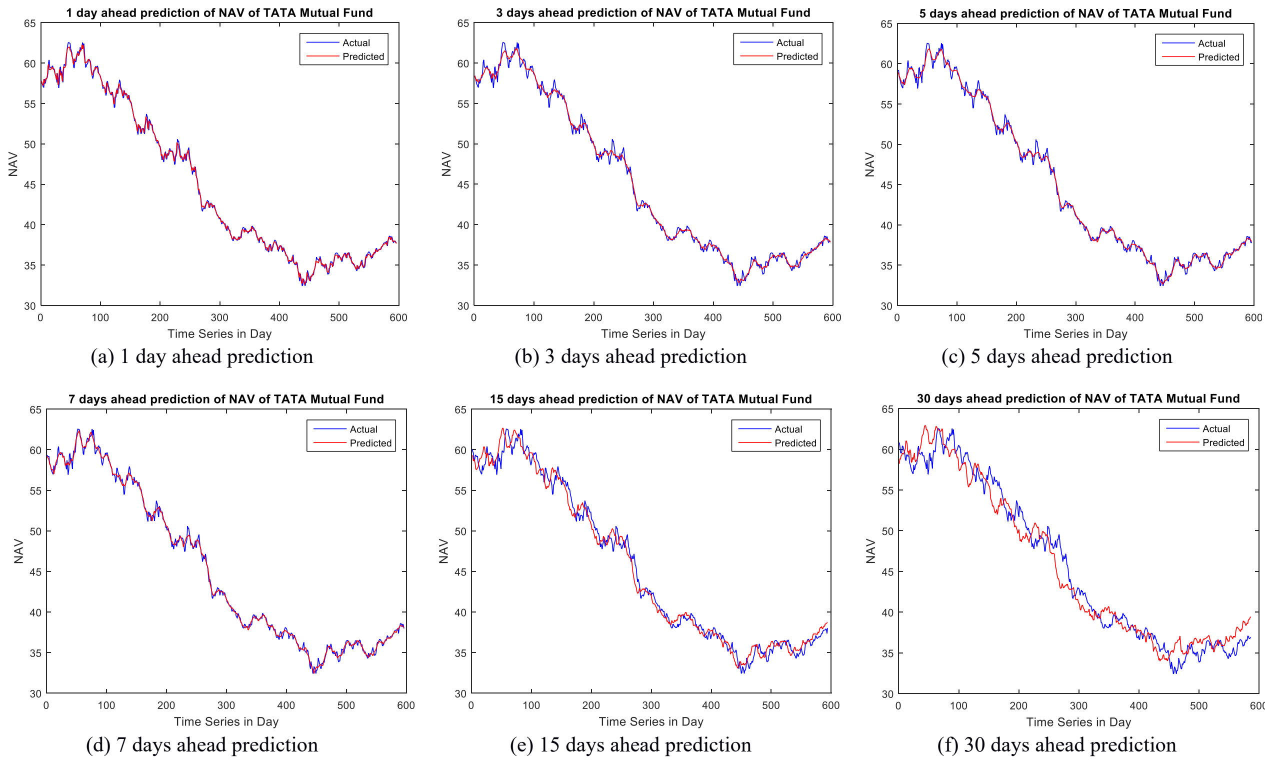
Fig. 10. Simulation graph of Evolutionary ELM for the days ahead prediction using SBI Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

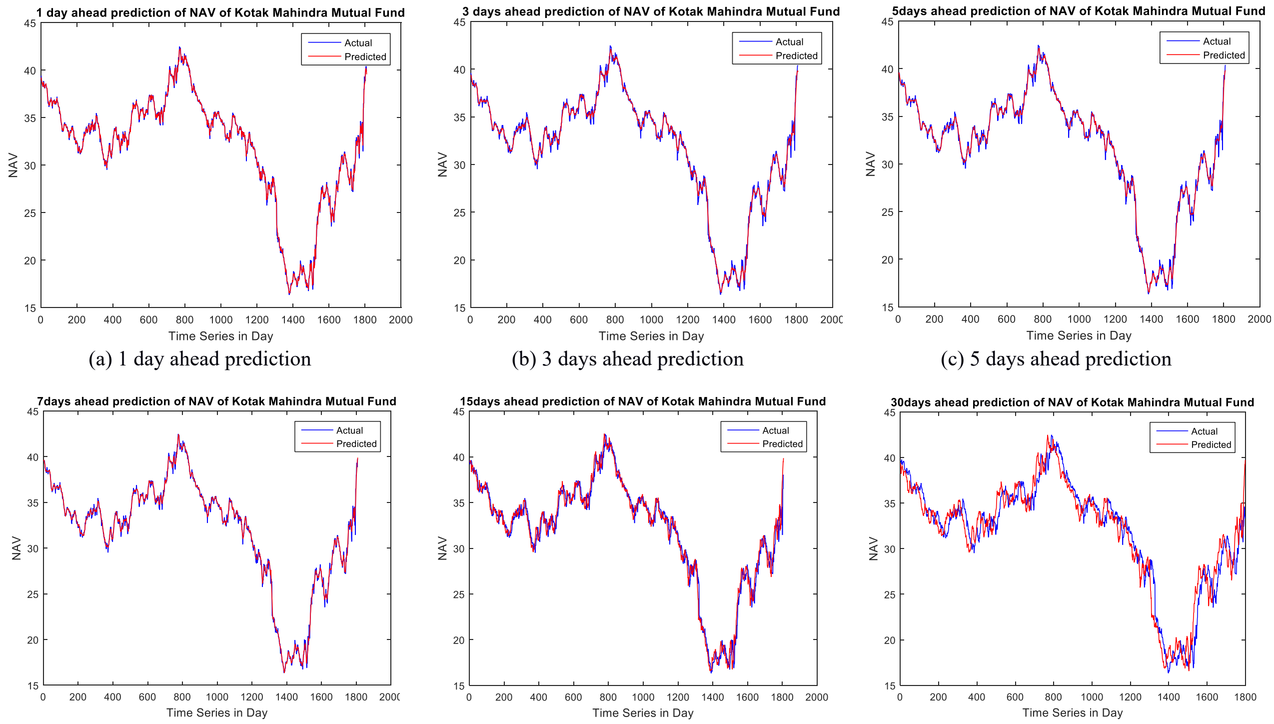
Fig. 11. Simulation graph of Evolutionary ELM for the days ahead prediction using UTI Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

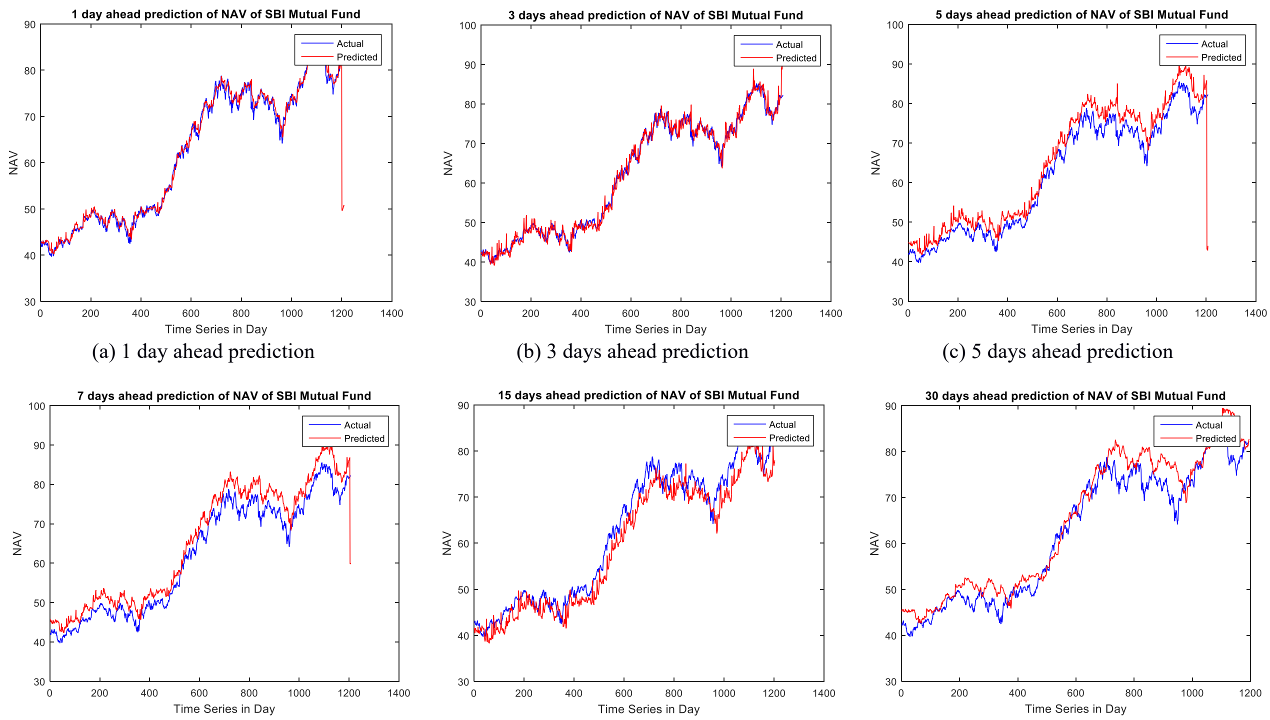
Fig. 12. Simulation graph of Evolutionary ELM for the days ahead prediction using TATA Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

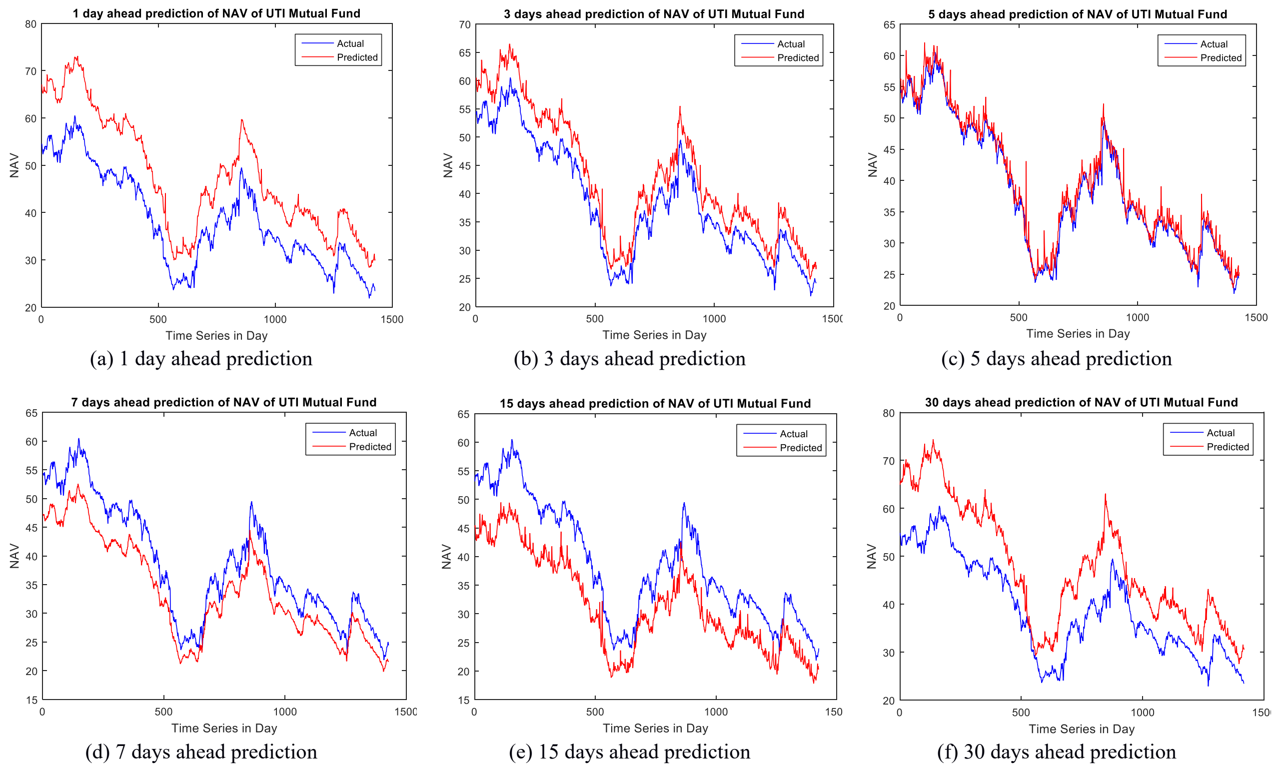
Fig. 13. Simulation graph of Evolutionary ELM for the days ahead prediction using Kotak Mahindra Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

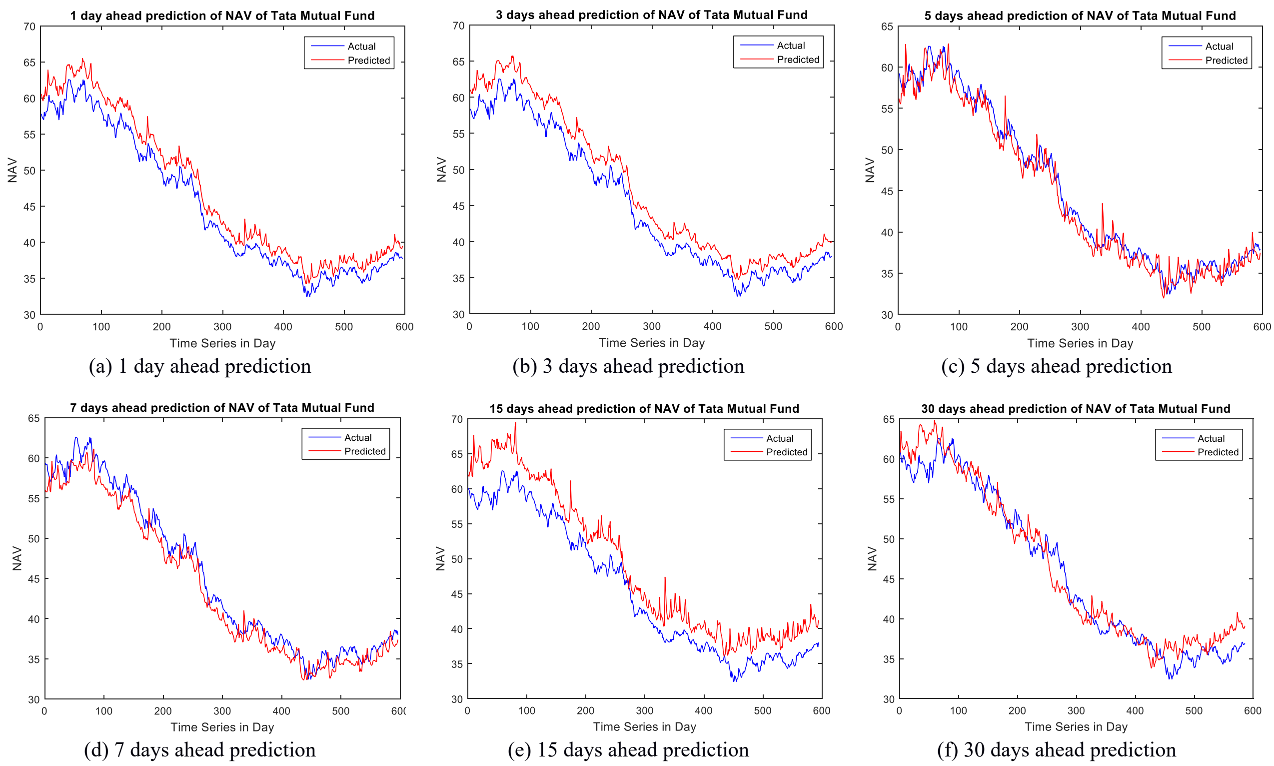
Fig. 14. Simulation graph of Error Minimized ELM for the days ahead prediction using SBI Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

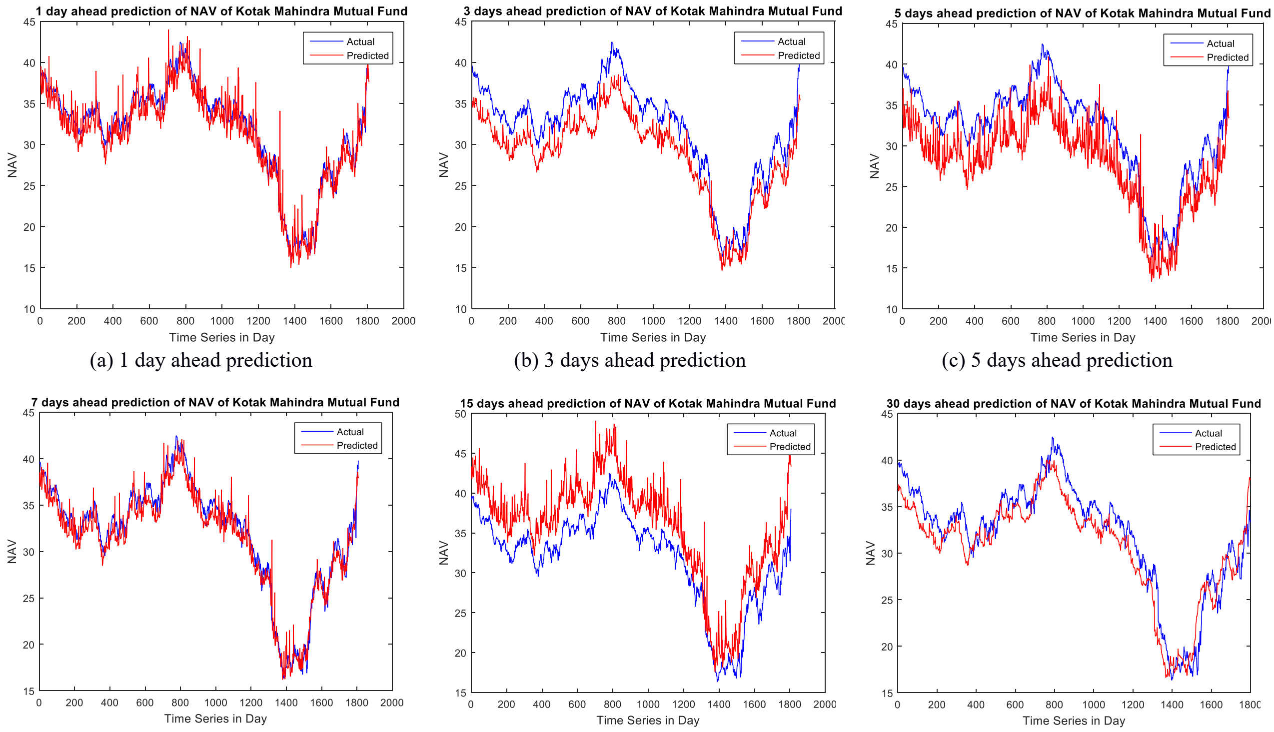
Fig. 15. Simulation graph of Error Minimized ELM for the days ahead prediction using UTI Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

Fig. 16. Simulation graph of Error Minimized ELM for the days ahead prediction using TATA Mutual Fund



(a) 1 day ahead prediction (b) 3 days ahead prediction (c) 5 days ahead prediction

(d) 7 days ahead prediction (e) 15 days ahead prediction (f) 30 days ahead prediction

Fig. 17. Simulation graph of Error Minimized ELM for the days ahead prediction using Kotak Mahindra Mutual Fund

Table 3. MSE calculation of ELM, OSELM, Evolutionary ELM and Error minimized ELM using SBI Mutual Fund.

Days ahead	ELM	Evolutionary ELM	OSELM	Error minimized ELM
1 day	1.621	0.33183	0.40278	0.94524
3 days	4.9674	0.23981	1.3335	0.25335
5 days	2.894	1.8007	1.8531	1.9499
7 days	3.3632	2.4059	3.2652	9.0337
15 days	6.2391	4.9412	9.4766	1.8972
30 days	11.0471	9.5418	30.1187	9.9589

The NAV prediction result of SBI Mutual fund using ELM, Evolutionary ELM, OSELM and Error minimized ELM learning approach are shown in Figure 2, Figure 6, Figure 10 and Figure 14 respectively. The MSE result at the time of training is shown in Table 3. The prediction is carried out for 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead for all the dataset using all prediction models such as ELM, Evolutionary ELM, OSELM and Error minimized ELM. From the above mentioned figures and the respective table for SBI mutual fund it is quite evident that ELM trained with evolutionary optimization technique performed better than OSELM, Error minimized ELM and basic ELM, for all the days' ahead prediction. Apart from the overall performance of Evolutionary ELM, the individual performance of all the models for all the days ahead is discussed. In 1 day and 5 days ahead prediction the performance sequence of the models in best to worst is described in the following way, first Evolutionary ELM, second OSELM, third Error minimized ELM and last one is ELM. Whereas, in 3 days ahead NAV prediction Error minimized ELM is performing better than OSELM and ELM. On the other hand in case of 7 days ahead prediction ELM is performing better than Error minimized ELM, and the rest performance sequence is same as 1 day ahead prediction. There is an exception in 15 days ahead prediction, where Error minimized ELM is performing better than the rest of the models and the performance sequence is Error minimized ELM, Evolutionary ELM, ELM and OSELM ordered in best to worst. In 30 days ahead prediction the performance sequence from best to worst is followed by Evolutionary ELM, Error minimized ELM, ELM then OSELM. The overall comparison of all the models in predicting SBI mutual fund specified that Evolutionary ELM is performing better than all the models.

Table 4. MSE calculation of ELM, OSELM, Evolutionary ELM and Error minimized ELM using UTI Mutual Fund.

Days ahead	ELM	Evolutionary ELM	OSELM	Error minimized ELM
1 day	0.20351	0.19767	0.19856	19.67443
3 days	0.3603	0.34164	0.3442	11.22731
5 days	0.28732	0.26871	0.26889	3.4165
7 days	0.24781	0.24073	0.24094	13.8657
15 days	3.874	1.6971	1.7334	22.4172
30 days	9.3302	7.3769	8.102	26.0895

To predict future behavior of mutual fund statistical measures performs an important role. This study chooses four statistical measures such as mean, standard deviation, kurtosis and skewness along with the NAV of mutual funds as input to the models. The value of statistical measures is calculated by using the NAV of mutual funds. The comparative results presented through actual versus predicted graph in Figure 3, Figure 7, Figure 11 and Figure 15 for ELM, OSELM, Evolutionary ELM and Error minimized ELM respectively. The experimental results of ELM, Evolutionary ELM, OSELM and Error minimized ELM for UTI Mutual fund is analyzed through MSE value during training phase presented in Table 4. For all the days such as 1day, 3 days, 5 days, 7 days, 15 days and 30 days ahead prediction the Evolutionary ELM is outperformed over the rest of the models. Apart from the evolutionary ELM, the performance of other models over UTI mutual fund is analyzed in this study. In all days ahead prediction OSELM is performing better than ELM and Error Minimized ELM and from the performance comparison between ELM and Error minimized ELM through MSE result it is clearly delineate ELM outperforms over Error minimized ELM.

Table 5. MSE calculation of ELM, OSELM, Evolutionary ELM and Error minimized ELM using Tata Mutual Fund.

Days ahead	ELM	Evolutionary ELM	OSELM	Error minimized ELM
1 day	0.25824	0.13593	0.13775	5.1797
3 days	0.24467	0.23484	0.24483	7.0477
5 days	0.40648	0.18562	0.19232	2.4172
7 days	0.40343	0.15777	0.15881	3.05
15 days	1.3334	1.1609	1.1947	15.8434
30 days	3.6491	3.3977	3.5707	5.9491

From the meticulous simulation results for Tata mutual fund shown in Figure 4, Figure 8, Figure 12 and Figure 16 in the form of actual versus predicted graph and the comparative MSE results shown in Table 5 in the training phase exhibit that Evolutionary ELM outperforms over all other models such as ELM, OSELM and Error minimized ELM irrespective of the days ahead to be predicted. In evolutionary ELM, DE is considered for weight optimization. In order to get better simulation result DE's control variables are not very difficult to choose. The crosses over rate and mutation scale are showing better result at 0.8 and 0.6 respectively. The optimal weight is fed to the testing phase for prediction result. ELM optimized with DE provides better prediction performance for four different NAV values for different days' ahead prediction ranging from one day to 30 days ahead prediction. In terms of prediction accuracy Evolutionary ELM is best, which is evident from the MSE result given in Table 5. Apart from the Evolutionary ELM, the comparative performance of other variants of ELM model such as OSELM and Error minimized ELM along with basic ELM for Tata mutual fund is marked out here in this study. Except 3 days ahead prediction, in all different days ahead to be predicted considered here, for simulation, it is shown that OSELM is performing better than basic ELM and Error minimized ELM and comparing the result between ELM and Error minimized the conclusion is obtained concerning the better performance of ELM. In 3 days ahead prediction the performance of ELM is better compared to OSELM and Error minimized ELM

Table 6. MSE calculation of ELM, OSELM, Evolutionary ELM and Error minimized ELM using Kotak Mahindra Mutual Fund.

Days ahead	ELM	Evolutionary ELM	OSELM	Error minimized ELM
1 day	0.093972	0.090726	0.092574	2.3934
3 days	0.1522	0.14589	0.15186	10.2867
5 days	0.16576	0.11677	0.12188	19.2647
7 days	0.1742	0.093565	0.093924	1.7391
15 days	0.98306	0.77458	0.86364	18.3605
30 days	3.3755	2.8068	3.3538	5.1545

To evaluate the prediction performance of all the models such as ELM, Evolutionary ELM, OSELM and Error minimized ELM; four mutual fund dataset is fed into the model. These four mutual funds such as SBI mutual fund, UTI mutual fund, Tata mutual fund and kotak Mahindra mutual fund, generate the data patterns for both training and testing. The trained parameters are used directly for the testing without requiring any training again. The actual and predicted graph of Kotak Mahindra mutul fund for ELM, OSELM, Evolutionary ELM and Error minimized ELM is shown in the Figure 5, Figure 9, Figure 13 and Figure 17 respectively. For each input pattern the MSE is calculated for each model during the training phase. From the MSE result given in Table 6, it can be clearly figured that evolutionary ELM outperforms over the rest of the model for different day's ahead prediction. Unlike other variants of ELM such as basic ELM, OSELM and Error minimized ELM, in evolutionary ELM after getting the error from the difference between actual NAV and predicted NAV, the error is used to update the input weights by using DE, to get optimal solution for getting better prediction efficiency. Apart from the proficient accuracy of Evolutionary ELM, the performance accuracy of the rest of the model is observed. The performance sequence of ELM, OSELM, Error minimized error is shown in descending order, at first OSELM is performing better over all the dataset, then comes basic ELM and at last Error minimized ELM. The performance is applicable for all the days ahead prediction in Kotak Mahindra mutual fund.

Table 7. Comparison of overall prediction accuracy of Evolutionary ELM with OSELM, ELM and Error minimized ELM network for NAV base on the MSE for SBI Mutual fund

	Comparison level (%)			**Comparison level (%)**			**Comparison level (%)**		
Days ahead	Evolutionary ELM	OS-ELM	Improvement in (%)	Evolutionary ELM	ELM	Improvement in (%)	Evolutionary ELM	Error minimized ELM	Improvement in (%)
1 day	0.33183	0.40278	17.61	0.33183	1.621	79.52	0.33183	0.94524	64.89
3 days	0.23981	1.3335	82.01	0.23981	4.9674	95.17	0.23981	0.25335	5.34
5 days	1.8007	2.9531	39.02	1.8007	2.894	37.77	1.8007	1.9499	7.65
7 days	2.4059	3.8652	37.75	2.4059	3.3632	28.46	2.4059	9.0337	73.36
15days	4.9412	9.4766	47.85	4.9412	6.2391	20.80	4.9412	1.8972	No improvement
30 days	9.5418	30.1187	68.31	9.5418	11.0471	13.62	9.5418	6.4589	No improvement

Table 8. Comparison of overall prediction accuracy of Evolutionary ELM with OSELM, ELM and Error minimized ELM network for NAV base on the MSE for UTI Mutual fund

	Comparison level (%)			**Comparison level (%)**			**Comparison level (%)**		
Days ahead	Evolutionary ELM	OS-ELM	Improvement in (%)	Evolutionary ELM	ELM	Improvement in (%)	Evolutionary ELM	Error minimized ELM	Improvement in (%)
1 day	0.19767	0.19856	0.44	0.19767	0.20351	2.86	0.19767	19.67443	98.99
3 days	0.34164	0.3442	0.74	0.34164	0.3603	5.17	0.34164	11.22731	96.95
5 days	0.26871	0.26889	0.06	0.26871	0.28732	6.47	0.26871	3.4165	92.13
7 days	0.24073	0.24094	0.08	0.24073	0.24781	2.85	0.24073	13.8657	98.26
15days	1.6971	1.7334	2.09	1.6971	3.874	56.19	1.6971	22.4172	92.42
30 days	7.3769	8.102	8.94	7.3769	9.3302	20.93	7.3769	26.0895	71.72

Table 9. Comparison of overall prediction accuracy of Evolutionary ELM with OSELM, ELM and Error minimized ELM network for NAV base on the MSE for Tata Mutual fund

	Comparison level (%)			**Comparison level (%)**			**Comparison level (%)**		
Days ahead	Evolutionary ELM	OS-ELM	Improvement in (%)	Evolutionary ELM	ELM	Improvement in (%)	Evolutionary ELM	Error minimized ELM	Improvement in (%)
1 day	0.13593	0.13775	1.32	0.13593	0.25824	47.36	0.13593	5.1797	97.37
3 days	0.23484	0.24483	4.08	0.23484	0.24467	4.01	0.23484	7.0477	96.66
5 days	0.18562	0.19232	3.48	0.18562	0.40648	54.33	0.18562	2.4172	92.32
7 days	0.15777	0.15881	0.65	0.15777	0.40343	60.89	0.15777	3.05	94.82
15days	1.1609	1.1947	2.82	1.1609	1.3334	No improvement	1.1609	15.8434	92.76
30 days	3.3977	3.5707	4.84	3.3977	3.6491	6.88	3.3977	5.9491	42.88

Table 10. Comparison of overall prediction accuracy of Evolutionary ELM with OSELM, ELM and Error minimized ELM network for NAV base on the MSE for Kotak Mahindra Mutual fund

	Comparison level (%)			**Comparison level (%)**			**Comparison level (%)**		
Days ahead	Evolutionary ELM	OS-ELM	Improvement in (%)	Evolutionary ELM	ELM	Improvement in (%)	Evolutionary ELM	Error minimized ELM	Improvement in (%)
1 day	0.090526	0.092574	2.21	0.090526	0.093972	3.66	0.090526	2.3934	96.21
3 days	0.14589	0.15186	3.93	0.14589	0.1522	4.14	0.14589	10.2867	98.58
5 days	0.11677	0.12188	4.19	0.11677	0.16576	29.55	0.11677	19.2647	99.39
7 days	0.093565	0.093924	0.38	0.093565	0.1742	46.28	0.093565	1.7391	94.41
15days	0.77458	0.86364	10.31	0.77458	0.98306	21.2	0.77458	18.3605	95.78
30 days	2.8068	3.3538	16.3	2.8068	3.3755	16.84	2.8068	5.1545	45.54

In reference to Table 7 the improved result of Evolutionary ELM for SBI mutual fund in predicting the follow up days by,

(a) More than 17.61%, 82.01%, 39.02%, 37.75%, 47.85%, 68.31% (1 day, 3days, 5 days, 7 days, 15 days and 30 days respectively) compared to OSELM,

(b) More than 79.52%, 95.17%, 37.77%, 28.46%, 20.80%, 13.62% (1 day, 3days, 5 days, 7 days, 15 days and 30 days respectively) compared to ELM,

(c) More than 64.89%, 5.34%, 7.65%, 73.36% (1 day, 3days, 5 days, 7 days respectively) and for 15 days and 30 days ahead prediction of NAV there is no improvement compared to Error minimized ELM.

In reference to Table 8 the improved result of Evolutionary ELM for UTI mutual fund in predicting the follow up days by,

(a) More than 0.44%, 0.74%, 0.06%, 0.08%, 2.09%, 8.94% (1 day, 3days, 5 days, 7 days, 15 days and 30 days respectively) compared to OSELM,

(b) More than 2.86%, 5.17%, 6.47%, 2.85%, 56.19%, 20.93% (1 day, 3days, 5 days, 7 days, 15 days and 30 days respectively) compared to ELM,

(c) More than 98.99%, 96.95%, 92.13%, 98.26%, 92.42%, 71.72% (1 day, 3days, 5 days, 7 days, 15 days, 30 days respectively) compared to Error minimized ELM.

In reference to Table 9 the improved result of Evolutionary ELM for Tata mutual fund in predicting the follow up days by,

(a) More than 1.32%, 4.08%, 3.48%, 0.65%, 2.82%, 4.84% (1 day, 3days, 5 days, 7 days, 15 days and 30 days respectively) compared to OSELM,

(b) More than 47.36%, 4.01%, 54.33%, 60.89%, 6.88% (1 day, 3days, 5 days, 7 days and 30 days respectively) compared to ELM and there is no improvement of Evolutionary ELM over ELM at 15 days ahead prediction.

(c) More than 97.37%, 96.66%, 92.32%, 94.82%, 92.76%, 42.88% (1 day, 3days, 5 days, 7 days, 15 days, 30 days respectively) compared to Error minimized ELM.

In reference to Table 10 the improved result of Evolutionary ELM for Kotak Mahindra Mutual Fund n predicting the follow up days by,

(a) More than 2.21%, 3.93%, 4.19%, 0.38%, 10.31%, 16.3% (1 day, 3 days, 5 days, 7 days, 15 days and 30 days respectively) compared to OSELM.

(b) More than 3.66%, 4.14%, 29.55%, 46.28%, 21.2%, 16.84% (1 day, 3days, 5 days, 7 days, 15 days and 30 days respectively) compared to ELM,

(c) More than 96.21%, 98.58%, 99.39%, 94.41%, 95.78%, 45.54% (1 day, 3days, 5 days, 7 days, 15 days, 30 days respectively) compared to Error minimized ELM.

Table 11. MSE calculation for various numbers of nodes in ELM, OSELM and Evolutionary ELM for SBI Mutual Fund.

No of Nodes	ELM	Evolutionary ELM	OSELM
3	1.4652	0.33242	0.26515
5	0.61479	0.13324	0.16523
7	0.40278	0.37835	0.11875
9	0.34677	0.22177	0.15621
15	0.30213	0.12688	0.10981
20	1.38812	2.41686	2.89624
25	1.17478	2.72406	2.99143
30	1.24482	1.99406	1.67253
35	2.48901	8.05609	1.62543

Table 12. MSE calculation for various numbers of nodes in ELM, OSELM and Evolutionary ELM for UTI Mutual Fund.

No of Nodes	ELM	Evolutionary ELM	OSELM
3	1.92313	1.32143	1.02871
5	0.92353	0.93182	0.96723
7	0.54123	0.73198	0.62198
9	0.78234	0.85318	0.51984
15	0.32125	0.489122	0.29319
20	2.23171	3.981264	3.891276
25	1.872395	3.812654	4.189324
30	1.923872	1.981742	1.83287
35	5.451265	7.032892	1.98236

Table 13. MSE calculation for various numbers of nodes in ELM, OSELM and Evolutionary ELM for TATAMutual Fund.

No of Nodes	ELM	Evolutionary ELM	OSELM
3	1.56123	0.98321	0.34521
5	0.92317	0.61243	0.189324
7	0.51943	0.50321	0.387123
9	0.61234	0.57625	0.31543
15	0.41431	0.46254	0.21762
20	3.91652	5.91627	5.91624
25	2.91726	3.91726	4.92712
30	2.91726	2.91726	1.26514
35	6.27835	7.91754	2.19285

Table 14. MSE calculation for various numbers of nodes in ELM, OSELM and Evolutionary ELM for Kotak Mahindra Mutual Fund.

No of Nodes	ELM	Evolutionary ELM	OSELM
3	1.72855	0.312876	0.615231
5	0.91264	0.816253	0.182673
7	0.91287	0.627413	0.328172
9	0.712652	0.582756	0.312541
15	0.618251	0.426325	0.172642
20	3.911726	5.273524	9.282631
25	2.861423	3.826354	3.298523
30	2.178241	2.937463	2.192851
35	3.167243	6.761725	1.413292

Selecting the number of nodes in the hidden layer itself plays an important role in neural network. Many researchers have given their effort to analyze the [27] the solution of the problem, that in order to get best result with minimum training time what will be the number nodes kept in the hidden layer. Unfortunately no one is succeed in finding the optimal formula in order to get the number of nodes in the hidden layer with reducing the training time with maximum performance accuracy. The performance of nodes in the hidden layer also depends upon the types of data taken in the input layer used for training. If the number of nodes increases then it's a chance to get better performance accuracy. But in this way the complexities will be maximum, which is not an optimal solution. Here, this study has been explored over the number of nodes. The MSE result has obtained for 3, 5, 7, 9, 15, 20, 25, 30, 35 different number of nodes for ELM, Evolutionary ELM and OSELM through simulation using four mutual fund dataset. In Error minimized ELM, the number of nodes in the hidden layer increases according to the algorithm specific control, hence for this experiment over different number of nodes Error minimized ELM is not included. From the result of MSE during training for three different models such as ELM, Evolutionary ELM and OSELM for four datasets SBI Mutual fund, UTI Mutual fund, Tata mutual fund and Kotak Mahindra mutual fund shown in Table 11, Table 12, Table 13, and Table 14, it can be observed that from 3 up to 15 number, as the number of nodes increases the MSE is decreasing and it is giving better result when the number of nodes is 15. After 15 the MSE is increasing, which is observed in all the models. It proves in this study, that considering 15 numbers of nodes in the hidden layer contribute maximum to the input layer for getting better prediction accuracy.

3.6 Descriptions of activation functions

In ANN the weighted sum of input is passed to the activation function to transform the activation level in to [28] output. Activation function is used in the network to introduce the non linearity, so that more complex pattern can be learnt by the network. Descriptions of the linear and non linear activation functions used here in this study are discussed below:

(a) Pure Linear is a neural transfer function, which calculates layer outputs from its net input. As the function is linear, hence the output result is not restricted to any range. The name of the function is purelin( ). The mathematical function of linear activation function is given in (2)

$$\text{Purelin}(x) = x, \text{ the range is within } (-\infty \text{ to } +\infty) \tag{2}$$

(b) Positive linear transfer function is a linear transfer function calculating layer outputs from its net input, which returns x if it is greater than or equal to zero and returns zero if x is less than or equal to zero. The name of the function is poslin(). Mathematical equation of this function is stated in (3)

$$\text{Poslin(x)} = \begin{cases} x, \text{if } x \geq 0; \\ 0, \text{if } x <= 0; \text{ the range is within } (0 \text{ to } +\infty) \end{cases} \tag{3}$$

(c) Rectified linear unit (ReLU) is a simpler non liner activation functions, which is most popular in neural network. It returns zero if the input is less than zero otherwise returns the raw output. ReLU does not go to the negative region rather saturate at exactly zero, but in positive region or in upper range, it does not saturate and converges faster. The mathematical formulation of this function is given in (4)

$$\text{F}(x) = \begin{cases} 0, \text{if } x < 0 \\ x, \text{otherwise} \end{cases} \tag{4}$$

(d) Leaky ReLU is having all the characteristics of ReLU such as computationally efficient and does not saturate at positive region but except one thing, instead of returning zero, when the x<0, it returns a small positive slope 0.01. The mathematical formula of Leaky ReLU is given in (5)

$$\text{F(x)} = \begin{cases} \mathbf{0.01, if\ x < 0} \\ \mathbf{x, otherwise} \end{cases} \tag{5}$$

(e) Sigmoid function is a nonlinear activation function, widely used in neural network. Unlike linear function, its range is within 0 to 1. Here, the large negative number is converted in to 0 and the large positive number is converted to 1. The mathematical formula of the function is described in (6)

$$\text{F}(x) = \frac{1}{1+e^{-x}} \tag{6}$$

(f) Hyperbolic tangent function is a continuous function, which produces output in the range between -1 and +1. If a strongly input value is fed to the network, it gives the output value very near to zero. This function is described mathematically in (7)

$$\text{F(x)} = \frac{1-e^{-x}}{1+e^{-x}} \tag{7}$$

(g) Sinc function is a sinusoidal activation function, whose output closes to zero when the value of x is either large positive or large negative. When the value of x is equal to 0, then there is an exception that instead of undefined value, the sinc(0) is defined to 1. Mathematically it is formulated in (8)

$$\text{F(x)} = \begin{cases} \frac{Sin(x)}{x}, x \neq 0 \\ 1, x = 0 \end{cases} \tag{8}$$

Table 15. MSE calculation using various activation functions applying on ELM, OSELM, Evolutionary ELM and Error minimized ELM for SBI Mutual Fund.

Activation Functions	ELM	Evolutionary ELM	OSELM	Error minimized ELM
Pure linear	0.71596	0.33411	0.43521	0.69864
Leaky ReLU	**0.46848**	0.33387	0.43254	**0.39234**
Positive linear	0.48487	0.33238	0.54355	0.39834
ReLU	0.49602	**0.33312**	**0.37652**	0.41786
Sigmoid	$2.363e^{+3}$	30.0695	37.3245	45.9675
Hyperbolic tangent	$1.0265e^{+4}$	35.2183	39.9812	77.9876
Sinc	33.09	0.33416	0.4365	22.8978

Table 16. MSE calculation using various activation functions applying on ELM, OSELM, Evolutionary ELM and Error minimized ELM for UTI Mutual Fund.

Activation Functions	ELM	Evolutionary ELM	OSELM	Error minimized ELM
Pure linear	0.19876	0.19775	0.19987	0.34323
Leaky ReLU	**0.18824**	**0.19767**	0.2033	0.21793
Positive linear	0.19861	0.19794	0.19976	0.19852
ReLU	0.18951	0.19774	**0.19973**	**0.19971**
Sigmoid	$2.7852e^{+3}$	21.8213	38.7673	84.9866
Hyperbolic tangent	1.9845^{+4}	16.8725	23.7634	47.9812
Sinc	13.0756	0.19829	0.24322	2.28978

Table 17. MSE calculation using various activation functions applying on ELM, OSELM, Evolutionary ELM and Error minimized ELM for TATA Mutual Fund.

Activation Functions	ELM	Evolutionary ELM	OSELM	Error minimized ELM
Pure linear	0.016421	0.13661	0.15412	0.34512
Leaky ReLU	**0.12134**	0.2011	0.2111	0.21154
Positive linear	0.41929	0.13647	0.2165	0.41632
ReLU	0.13736	**0.13625**	**0.13689**	**0.13699**
Sigmoid	97.2637	35.2993	44.893	67.9384
Hyperbolic tangent	41.023974	19.444157	21.4522	44.9273
Sinc	63.54121	0.13616	0.83411	73.9919

Table 18. MSE calculation using various activation functions applying on ELM, OSELM, Evolutionary ELM and Error minimized ELM for Kotak Mahindra Mutual Fund.

Activation Functions	ELM	Evolutionary ELM	OSELM	Error minimized ELM
Pure linear	0.37329	0.10978	0.10944	0.196652
Leaky ReLU	**0.13412**	**0.08342**	**0.15412**	0.30112
Positive linear	0.14087	0.107907	0.17003	0.29101
ReLU	0.28209	0.090742	0.281712	**0.08195**
Sigmoid	$1.6453e^{+3}$	38.5481	56.9812	$5.3653e^{+3}$
Hyperbolic tangent	6534.99889687	99.2707	104.6712	251.3826
Sinc	112.9221	0.090762	0.2165	23.4221

In most general way it can be said activation function, also known as transfer function limits the output result to a finite value. The activation function can be linear and also it can be non linear, and in most the cases, it has been observed that the maximum activation functions used in neural network are nonlinear. In neural network activation function is needed to introduce the non linearity. The interesting properties of units can be captured using nonlinear mapping. Activation functions for each neuron specify the output for the input given to that neuron. Here in this study, both linear as well as nonlinear activation functions have considered. Comparing with other activation functions ReLU is considering more mileage. It is a non linear activation function with the range between 0 to infinity, which can blow up the activation. Discussing over the sparsity of activation function, it is observed that sigmoid and hyperbolic sigmoid activate all neurons in an analogue way, it means to describe the output all most all activations has to be processed. To make the activation sparse and efficient, some neurons in the network can be avoided for activation, as the activation in dense is very costly in terms of complexities. In that scenario ReLU gives benefit, as the network yields 0 for negative value of x. It means a fewer neurons are firing (sparse activation). ReLU is giving horizontal line for negative value, the result of which, the gradient can move towards zero; hence the weight cannot be adjusted during descent. The neurons in that stage stop responding to error variations. To overcome this problem Leaky ReLU is introduced which slightly inclined the line making the horizontal line into non-horizontal line. The main idea is to recover eventually during training and the gradient to be non zero. Compared to sigmoid and hyperbolic tangent, ReLU and Leaky ReLU both are very simple and less computationally expensive as it needs a very simpler mathematical expression. Pure linear and positive linear are two linear transfer functions used in this study. The range of pure liner is within negative infinitive to positive infinitive, which means it does not restrict to any range and simultaneously giving the output as the raw value. Furthermore discussing about positive linear activation function, the negative side of zero returns zero and the positive side of zero returns the value. In addition to this, another activation function Sinc is considered for the experimental study, which converges to zero for the large positive value or large negative value. Analyzing the simulation result given in Table 15, Table 16, Table 17 and Table 18 for different activation functions applying over ELM, OSELM, Evolutionary ELM and Error minimized ELM for SBI, UTI, Tata Mutual fund and Kotak Mahindra Mutual fund, it is observed that in most of the cases ReLU and Likey ReLU is giving better result in the form of minimum MSE in training phase. For SBI mutual fund Leaky ReLU is performing better for ELM and Error minimized ELM, whereas ReLU outperforms over all the activation functions for OSELM and Evolutionary ELM. Similarly the result of activation functions applying on UTI mutual fund is verified, where, it is found that in case of ELM and Evolutionary ELM Leaky ReLU is performing better and in the rest of the model performance of ReLU is better. Exploring over Tata mutual fund, it shows Leaky ReLU is giving better result in ELM but in rest of the model ReLU is performing better. Moreover for Kotak Mahindra mutual fund Leaky ReLU outperforms over all the activation functions applying on ELM, Evolutionary ELM and OSELM except in Error minimized ELM where ReLU is performing better.

The convergence graph of Evolutionary ELM for all the mutual fund datasets is exhibited in **Figure 18**. To make the comparison standardize the experiment has done for 100 numbers of iterations. From the error convergence graph, it can be clearly figured that, as the number of days increases the performance in terms of MSE and the converging speed is decreases.

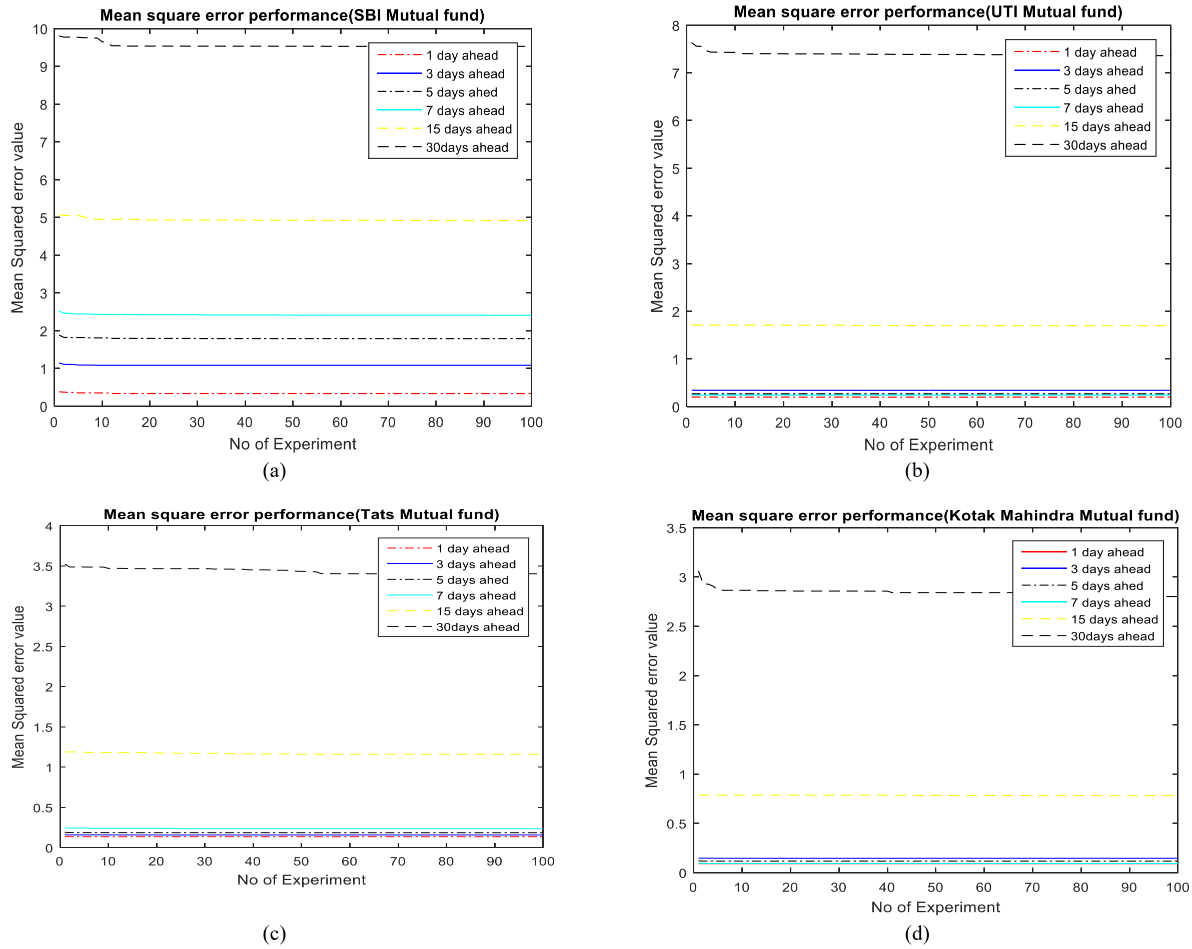


Fig. 18. Error convergence graph of Evolutionary ELM for (a) SBI, (b) UTI, (c) TATA and (d) Kotak Mahindra Mutual fund

4. Performance verifications

The training performance of the model has stopped after getting a stable value and at that state the optimal weight is carried out to testing phase. Here in this study 70% data considered for training and 30% of data is considered for testing [29]. For evaluating the performance of the model RMSE (Root Mean Square), MAPE (Mean Absolute Percentage Error), MAE (Mean Absolute Error), Theil'U and ARV are used in testing phase [30-33]. The result of the above mentioned performance measure over all the datasets for all the models used in this study is shown in Table 19.

Table 19. Performance evaluation measure for all the models over all the dataset

Dataset	Method	RMSE	MAPE	MAE	Theil'U	ARV
SBI Magnum Mutual Fund	ELM	0.86852	0.72706	0.49445	0.0052758	0.0018986
	OSELM	0.63465	0.72686	0.45349	0.0012916	0.0019917
	Evolutionary ELM	**0.57705**	**0.68524**	**0.4211**	**5.6758e^{-06}**	**0.0016713**
	Error Minimized ELM	0.67925	0.73542	0.49112	0.003871	0.0018265
UTI Equity Mutual Fund	ELM	0.49367	1.1573	0.60451	0.0003137	0.0023213
	OSELM	0.4456	0.86368	0.32017	0.00016719	0.0019402
	Evolutionary ELM	**0.44491**	**0.85817**	**0.31908**	**5.2113e^{-07}**	**0.0019398**
	Error Minimized ELM	0.44956	0.92763	0.59127	0.0006234	0.003452
Tata Mutual Fund	ELM	0.5275	0.63471	0.27791	0.0020729	0.001735
	OSELM	0.37115	0.62206	0.28026	0.00038173	0.0015322
	Evolutionary ELM	**0.36935**	**0.61261**	**0.27635**	**4.168e^{-06}**	**0.001517**
	Error Minimized ELM	0.41653	0.63271	0.27924	0.001284	0.001519
Kotak Mahindra Mutual Fund	ELM	0.39211	0.74842	0.25162	0.00078292	0.003306
	OSELM	0.30426	0.72039	0.21312	3.273e^{-05}	0.0025603
	Evolutionary ELM	**0.30125**	**0.69458**	**0.20472**	**3.2497e^{-07}**	**0.0025223**
	Error Minimized ELM	0.38154	0.74131	0.24127	0.0005163	0.002698

From the performance of four different variants of ELM model using NAV values of four different mutual funds to predict 1 day ahead NAV value, it is evident that Evolutionary ELM offers less error value for all the performance measure in testing phase. Hence, Evolutionary ELM is the best in terms of prediction efficiency, as it is showing better performance not only in training by calculating MSE but also in testing phase applying over the above mentioned performance measure. MSE value in training cannot be the only parameter to evaluate the performance of the model; hence through the different performance measures Evolutionary ELM proves its efficiency among four variants of ELM such as basic ELM, Evolutionary ELM, OSELM and Error minimized ELM compared in this study.

5. Overall work analysis

A mutual fund is a professionally-managed investment scheme, usually run by an asset management company that brings together a group of people and invests their money in stocks, bonds and other securities. All the mutual funds are registered with SEBI. They function within the provisions of strict regulation created to protect the interests of the investor. The present investigation in this study, regarding mutual fund prediction reveals some interesting observations. Four mutual fund data such as SBI Magnum Equity, UTI Equity mutual fund, TATA Dividend Yield Fund Direct- Growth and Kotak Mahindra mutual fund has considered for this study. SBI Magnum Equity is positioned as large cap fund with robust investment process with analyzing the broader economic outlook and about UTI Equity mutual fund, it invests maximum of its funds in equity with a risk of medium to high and very few funds is invested in debt and money market with low and medium risk. Both these mutual fund gives a maximum returns despite of high risk profile. TATA Dividend Yield Fund Direct-Growth is another mutual fund, which invests 70% of its fund in shares with high dividend yields; it means the Tata dividend yield is greater than dividend yield of BSE Sensex and the most profiting mutual fund such as Kotak Mahindra mutual providing wide range of schemes and promise to offer greater benefits of investment. These four mutual funds are very popular in their respective investment scheme with high profit.

The above four mutual fund is applied over the four prediction models such as ELM, OSELM, Evolutionary ELM and Error minimized ELM. ELM provides better result with faster [34] learning speed and least human involvement. Unlike neural network the hidden layer of ELM need not to be tuned at each iteration. Considering the generalization performance of ELM, some improved methods of ELM has introduced by the researcher. This study has analyzed the variants of ELM especially OSELM, Evolutionary ELM and Error minimized ELM including the basic ELM. Since Evolutionary algorithms [14] are used widely in the form of global searching optimization method which is very promising for the training of network. In addition to this, based on batch processing, OSELM is developed where the data learn one by one or chunk by chunk with a fixed number of chunks or varying number of chunks. Basically in ELM all the training data should be available before training process, which is not suitable when learning is an ongoing process as the complete set of data in not available. In that case the training of past data will repeat at the [35] arrival of new data, which needs a lot of time to complete the execution. This shortcoming of ELM can be short out through OSELM by learning the data one by one or block by block. Error minimized ELM is another variant of ELM, which is introduced to solve the issue of choosing optimal number of hidden [25] nodes. Instead of choosing the number of hidden nodes in hit and trial method, the number of nodes grows one by one or group by group till the optimal solution obtained. When new nodes is added the output weights updates incrementally, with significantly reducing the computational complexity. All these variants of ELM are efficient in their relative approach. Comparing the simulation result of all these variants of ELM, considered here in this study as prediction models, it is found that Evolutionary ELM outperforms over all the models.

The relative improvements of Evolutionary ELM over all the variants of ELM is analyzed in this study and it is found that the prediction accuracy of Evolutionary ELM is more than 90% compared to Error minimized ELM for UTI, Tata and Kotak Mahindra mutual fund in the time horizon of 1day, 3 days, 5 days, 7 days and 15 days but for 30 days ahead prediction the improvement is above 40%. There is an exception in SBI mutual fund, where the improvement is less compared to other mutual fund used here in this study. Moreover the Evolutionary ELM improves over OSELM in the range of 17% to 80% in different time horizon for SBI mutual fund but for UTI, Tata and Kotak Mutual fund the improvement range is within 0.06% to 16.3% for different days ahead prediction. Finally the improvement of Evolutionary ELM over basic ELM is computed and found that, the overall improvement for all the mutual fund is within 2.85% to approximately 60% for all time horizons. One exception is that the improvement is above 79% for 1 day and 3 days ahead prediction in case of SBI mutual fund and no improvement for 15 days ahead prediction in Tata mutual fund.

In addition to the comparison of training error among the above prediction model, another two interesting parameters such as number of hidden nodes and activation functions has been explored. Random selection of input nodes in the hidden layer take part an important role in neural network. In this study, 15 numbers of nodes contribute better efficiency to the prediction model compared to other number of nodes considered for experimental work. Error minimized ELM has not under gone through this comparison as this algorithm itself designed its network for the hidden layer nodes. Simultaneously various activation functions has fed in to the prediction model and after analyzing the result it is found that in maximum cases ReLU and Likey ReLU has given better prediction accuracy with minimum error in training phase compared to other activation functions. Apart from the training performance of the prediction model, the testing performance is also evaluated through different performance measures as MSE in training cannot be the only selection procedure to compare the efficiency among the models. From different performance measures, it is found that Evolutionary ELM is performing better in all the performance measures for all the mutual fund datasets consider here in this study. Exploring over different prediction models with different number of nodes and activation functions, this study proves Evolutionary ELM outperforms over the rest prediction models, with 15 numbers of nodes and ReLU and Likey ReLU as an activation functions.

6. Highlights

(1) Four mutual fund data such as SBI Magnum Equity, UTI Equity mutual fund, TATA Dividend Yield Fund Direct-

Growth and Kotak Mahindra mutual fund with high profit investment scheme have been considered for this study.
(2) The prediction analysis of variants of ELM such as; standard ELM, OSELM, Evolutionary ELM and Error minimized ELM are analyzed by testing over the above mutual funds.
(3) Experiments show that Evolutionary ELM is performing better in comparison with standard ELM, OSELM and Error minimized ELM considering MSE and performances with respect to measures for training and testing process respectively.
(4) In addition to this the relative improvements of the proposed Evolutionary ELM over standard ELM, OSELM and Error minimized ELM is calculated, which proves about its efficiency.
(5) Furthermore, this study has explored over another two aspects of prediction model such as number of hidden nodes and activation functions.

7. Conclusions and Future work

In this brief, the prediction performance of variants of ELM is analyzed by applying over different mutual funds. This study considered basic ELM, Evolutionary ELM, OSELM and Error minimized ELM as the prediction model to predict the NAV of SBI Magnum Equity, UTI Equity mutual fund, TATA Dividend Yield Fund Direct-Growth and Kotak Mahindra mutual fund. Compared with the MSE, in training of different prediction model, Evolutionary ELM is found to be better with maximum accuracy. Simultaneously testing phase is compared with different performance measures. In both training and testing phase Evolutionary ELM proves to be better than other prediction models. Exploring the other two aspects of network such as number of hidden layer nodes and activation functions, the study found, the model obtaining minimum MSE at 15 numbers of nodes. Further analyzing on activation functions, it is realized that in most of the cases ReLU and Likey ReLU achieve better generalization performance. Incorporating ELM variants can enhance portfolio management strategies for mutual fund managers, enabling better identification of profitable opportunities, optimized asset allocation, and risk mitigation. In future, apart from this four, other variants of ELM can be explored for different area of financial market prediction.

References

[1] W. C. Chiang, Timothy L. Urban, and Gable W. Baldridge. "A neural network approach to mutual fund net asset value forecasting." *Omega*, vol. 24, no. 2, pp. 205-215, Apr. 1996, doi: https://doi.org/10.1016/0305-0483(95)00059-3.
[2] D. C. Indro, C. X. Jiang, B. E. Patuwo, and G. P. Zhang. "Predicting mutual fund performance using artificial neural networks." *Omega,* vol. 27, no. 3, pp. 373-380, Jun. 1999, doi: https://doi.org/10.1016/S0305-0483(98)00048-6.
[3] H. Mamaysky, M. Spiegel, and H. Zhang, "Estimating the dynamics of mutual fund alphas and betas." *The Rev. Financ. Stud.,* vol. 21, no. 1, pp. 233-264, Jan. 2008, doi: https://doi.org/10.1093/rfs/hhm049.
[4] Y. Hongcheng, L. Wenyi, L. Xiuying, K. Huihua, and L. Chufu. "Predicting net asset value of investment fund based on BP neural network." in *2010 Internat. Conf. Comp. Applic. Sys. Model. (ICCASM),* vol. 10, IEEE, Oct 2010, pp. V10-635, https://doi.org/10.1109/ICCASM.2010.5622625.
[5] C. M. Anish, and Babita Majhi. "Prediction of mutual fund net asset value using low complexity feedback neural network." i*n IEEE Int. Conf. Curr. Tren. Adv. Comp*. (*ICCTAC),* IEEE, Mar. 2016, pp. 1-5, doi: https://doi.org/10.1109/ICCTAC.2016.7567345.
[6] C. M. Anish, and Babita Majhi. "An ensemble model for net asset value prediction." 1n *2015 IEEE Pow, Commun. Inf. Techn. Conf. (PCITC)*, IEEE, Oct. 2015, pp. 392-396, doi: https://doi.org/10.1109/PCITC.2015.7438197.
[7] C. M. Anish, Babita Majhi, and Ritanjali Majhi. "Development and evaluation of novel forecasting adaptive ensemble model." *The J. Financ. Dat. Sci.,* vol. 2, no 3, pp. 188-201, Sep. 2016, doi: https://doi.org/10.1016/j.jfds.2016.12.002.
[8] T. H. Huang, and Y. H. Leu. "A mutual fund investment method using fruit fly optimization algorithm and neural network." *Appl. Mechan. Mater.*, vol. 571, pp. 318-325, Aug. 2014, doi: https://doi.org/10.4028/www.scientific.net/AMM.571-572.318.
[9] E. Priyadarshini, and A. Chandra Babu. "A Comparative Analysis for Forecasting the NAV's of Indian Mutual Fund using Multiple Regression Analysis and Artificial Neural Networks." *Int. J. Trade, Econ. Financ.,* vol 3, no. 5, pp. 347, Oct. 2012.
[10] G. B. Huang, Q. Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications." *Neurocomputing* vol. 70, no. 1-3, pp. 489-501, Dec 2006, doi: https://doi.org/10.1016/j.neucom.2005.12.126.
[11] G.B. Huang, D. H. Wang, and Y. L.. "Extreme learning machines: a survey." *Int. J. Mach. Learn. Cybern.,* vol. 2, pp. 107-122, Jun. 2011.
[12] G. B. Huang, Q. Y. Zhu, and C. K. Siew. "Extreme learning machine: a new learning scheme of feedforward neural networks." in *IEEE Int. Joint Conf. Neural Netw. (IEEE Cat. No. 04CH37541)*, vol. 2, IEEE, Jul. 2004, pp. 985-990, doi: https://doi.org/10.1109/IJCNN.2004.1380068.
[13] G. Huang, G. B. Huang, S. Song, and K. You. "Trends in extreme learning machines: A review." *Neur. Networ.,* vol. 61, pp. 32-48, Jan. 2015, doi: https://doi.org/10.1016/j.neunet.2014.10.001.
[14] Q. Y. Zhu, A. K. Qin, P. N. Suganthan, and G. B.Huang. "Evolutionary extreme learning machine." *Patt. Recogn.,* vol. 38, no. 10, pp.1759-1763, Oct. 2015, doi: https://doi.org/10.1016/j.patcog.2005.03.028.
[15] S. Ding, X. Xu, and R. Nie. "Extreme learning machine and its applications." *Neur. Comput. Applic.s,* vol. 25, pp. 549-556, Sep. 2014.
[16] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie. "Extreme learning machine: algorithm, theory and applications." *Artif. Intell. Rev.,* vol. 44, pp.103-115, Jun. 2015.
[17] S. Das, and P. N. Suganthan. "Differential evolution: A survey of the state-of-the-art." *IEEE Trans. Evolut. Computat.,* vol. 15, no. 1, pp. 4-31, Oct. 2010, doi: https://doi.org/10.1109/TEVC.2010.2059031.
[18] Y. Lan, Y. C. Soh, and G. B.Huang. "Ensemble of online sequential extreme learning machine." *Neurocomputing,* vol. 72, no. 13-15, pp. 3391-3395, Aug. 2009, doi: https://doi.org/10.1016/j.neucom.2009.02.013.
[19] S. Yongjiao, Y. Yuan, and G. Wang. "An OS-ELM based distributed ensemble classification framework in P2P networks." *Neurocomputing* vol. 74, no. 16, pp. 2438-2443, Sep. 2011, doi: https://doi.org/10.1016/j.neucom.2010.12.040.
[20] J. Zhao, Z. Wang, and D. S. Park. "Online sequential extreme learning machine with forgetting mechanism." *Neurocomputing,* vol. 87, pp. 79-89, Jun. 2012, doi: https://doi.org/10.1016/j.neucom.2012.02.003.
[21] J. S. Lim, L. Seokjin, and H. S. Pang. "Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations." *Neur. Comput. Applic.,,* vol. 22, pp. 569-576, Mar. 2013.
[22] G. B.Huang, N. Y. Liang, H. J. Rong, P. Saratchandran, and N. Sundararajan. "On-Line Sequential Extreme Learning Machine." *Comput. Intell.,* 2005, pp. 232-237, Jul. 2005.
[23] K. S. Yap, and H. J. Yap. "Daily maximum load forecasting of consecutive national holidays using OSELM-based multi-agents system with weighted average strategy." *Neurocomputing,* vol. 81,

pp. 108-112, Apr. 2012, dpi: https://doi.org/10.1016/j.neucom.2011.12.002.

[24] Y. Lan, Y. C. Soh, and G. B.Huang. "Random search enhancement of error minimized extreme learning machine." In *ESANN,* Apr 2010.

[25] G. Feng, G. B.Huang, Q. Lin, and R. Gay. "Error minimized extreme learning machine with growth of hidden nodes and incremental learning." *IEEE Trans. Neural Netw.,* vol. 20, no 8, pp. 1352-1357, Jul. 2009, doi: 10.1109/TNN.2009.2024147.

[26] S. Balasundaram. "Application of error minimized extreme learning machine for simultaneous learning of a function and its derivatives." *Neurocomputing,* vol. 74, no.16, pp. 2511-2519, Sep. 2011, doi: https://doi.org/10.1016/j.neucom.2010.12.033.

[27] S. Karsoliya. "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture." *Int. J. Eng. Trends Techn.,* vol. 3, no. 6, pp. 714-717, Dec. 2012.

[28] P. Sibi, S. Allwyn Jones, and P. Siddarth. "Analysis of different activation functions using back propagation neural networks." *J. Theor. Appl. Inform. Techn.,* vol. 47, no. 3, pp. 1264-1268, Jan. 2013.

[29] C. M. Anish, Babita Majhi, and Ritanjali Majhi. "Development and evaluation of novel forecasting adaptive ensemble model." *The J. Financ. Dat. Sci.,* vol. 2, no. 3, pp. 188-201, Sep. 2016, doi: https://doi.org/10.1016/j.jfds.2016.12.002.

[30] B. Majhi, M. Rout, and V. Baghel. "On the development and performance evaluation of a multiobjective GA-based RBF adaptive model for the prediction of stock indices." *J. King Saud University-Comput. Inform. Sci.,* vol. 26, no. 3, pp. 319-331, Sep. 2014, doi: https://doi.org/10.1016/j.jksuci.2013.12.005.

[31] S. R. Das, D. Mishra, and M. Rout. "A hybridized ELM-Jaya forecasting model for currency exchange prediction." *J. King Saud University-Comput. Inform. Sci.,* vol. 32, no. 3, pp. 345-366, Mar. 2020, doi: https://doi.org/10.1016/j.jksuci.2017.09.006.

[32] T. AE Ferreira, G. C. Vasconcelos, and P. JL Adeodato. "A new intelligent system methodology for time series forecasting with artificial neural networks." *Neural Process. Let.,* vol. 28, pp. 113-129, Oct. 2008.

[33] R. Kalyan Nayak, D. Mishra, and A. K. Rath. "An optimized SVM-k-NN currency exchange forecasting model for Indian currency market." *Neur. Comput. Applic,* vol. 31, no. 7, pp. 2995-3021, Jul. 2019.

[34] S. Ding, X. Xu, and R. Nie. "Extreme learning machine and its applications." *Neur. Comput. Applic.,* vol. 25, pp. 549-556, Sep. 2014.

[35] J. Zhao, Z. Wang, and D. S. Park. "Online sequential extreme learning machine with forgetting mechanism." *Neurocomputing,* vol. 87, pp. 79-89, Jun. 2012, doi: https://doi.org/10.1016/j.neucom.2012.02.003.