World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:11, 2007

Application of Extreme Learning Machine Method for Time Series Analysis

Rampal Singh, and S. Balasundaram

Abstract**—**In this paper, we study the application of Extreme Learning Machine *(ELM)* algorithm for single layered feedforward neural networks to non-linear chaotic time series problems. In this algorithm the input weights and the hidden layer bias are randomly chosen. The *ELM* formulation leads to solving a system of linear equations in terms of the unknown weights connecting the hidden layer to the output layer. The solution of this general system of linear equations will be obtained using Moore-Penrose generalized pseudo inverse. For the study of the application of the method we consider the time series generated by the Mackey Glass delay differential equation with different time delays, Santa Fe A and UCR heart beat rate ECG time series. For the choice of *sigmoid, sin* and *hardlim* activation functions the optimal values for the memory order and the number of hidden neurons which give the best prediction performance in terms of root mean square error are determined. It is observed that the results obtained are in close agreement with the exact solution of the problems considered which clearly shows that *ELM* is a very promising alternative method for time series prediction.

Keywords**—**Chaotic time series, Extreme learning machine, Generalization performance.

I. Introduction

ARTIFICIAL Neural Networks *(ANNs)* have been extensively applied for pattern classification and regression problems. The major reason for the success of *ANNs* is their ability in obtaining a non-linear approximation model function describing the association between the dependent and independent variables using the given input samples. Since *ANNs* adaptively select the model from the features presented in the input data, they are applied to a large number of classes of problems of importance like optical character recognition [7], face detection [11], gene prediction [14], credit scoring [6] and time series forecasting [12], [17].Though *ANNs* have many advantages such as better approximation capabilities and simple network structures, however, it suffers from several problems such as presence of local minima's, imprecise learning rate, selection of the number of hidden neurons and over fitting. Moreover, the gradient descent based learning algorithms such as Back Propagation *(BP)* will generally lead to slow convergence during the training of the networks.

Time series forecasting is an important and challenging problem of regression. In the regression problem by analyzing the given input samples the best fit functional model describing the relationship between the dependent and independent variables is obtained.

There are many prediction models exist in the literature [4], [9], [12] for time series. The important and widely used among them are Auto Regressive Integrated Moving Average *(ARIMA)* [1], *ANNs* [12], [17] and Support Vector Regression *(SVR)* [8], [9], [13], [15] methods. Among the above methods, *ARIMA* assumes the existence of a linear relationship in the time series values, i.e. its prediction value will be a linear function of the past observations and therefore it is not always suitable for complex real world problems [18]. Also it is proposed to combine several methods in order to obtain improved forecasting accuracy. For the study of a hybrid approach of combining *ARIMA* and *ANN* for time series forecasting we refer the reader to [18]. For time series involving seasonality, combining Seasonal time series *ARIMA (SARIMA)* and *ANN* is discussed in [16] and for the study of a combined *SARIMA* and *SVR* approach see [3].

Huang et al [5] have proposed a new learning algorithm for Single hidden Layer Feedforward Neural Network *(SLFN)* architecture called Extreme Learning Machine *(ELM)* which overcomes the problems caused by gradient descent based algorithms such as *BP* applied in *ANNs*. In this algorithm the input weights and the hidden layer bias are randomly chosen. The *ELM* formulation leads to solving a system of linear equations in terms of the unknown weights connecting the hidden layer to the output layer. The solution of this general system of linear equations is obtained using Moore-Penrose generalized pseudo inverse [10]. In this work we discuss briefly the *ELM* algorithm and study its feasibility of application for chaotic time series prediction problems.

Throughout this paper, we assume all vectors to be column vectors. For any two vectors *x, y* in the m-dimensional real space $\Re^m$, we denote the inner product of the two vectors by $x'.y$ where x' is the transpose of the vector *x* and the norm of a vector by $\|\cdot\|$. The paper is organized as follows. In Section 2, we define Moore-Penrose generalized inverse, the minimum norm least squares solution of a general linear system of equations and state the relation between them. In Section 3, we revive the *ELM* algorithm for *SLFN*. For the application of this algorithm we considered Mackey Glass delay differential equation with different time delays, Santa Fe-A and UCR heart beat rate (ECG) time series in Section 4 and the results

Rampal Singh is with Department of Computer Science, Deen Dayal Upadhyaya College, University of Delhi, New Delhi-110015, India (phone:+91-11-27570620, 09350647546, e-mail: rpsrana@ddu.du.ac.in).

S. Balasundaram is with School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi-110067, India (phone: +91-11-26704724, e-mail: balajnu@hotmail.com).

obtained using *ELM* have been compared with exact solutions. Finally, we conclude our paper in Section 5.

II. Moore - Penrose Generalized Inverse

The solution of a general linear system of equations

$$Ax = y\,,$$

where A may be a singular or rectangular matrix can be obtained by the use of the Moore-Penrose generalized pseudo inverse.

Definition 2.1 [10]: A matrix G of size $n \times m$ is called the Moore-Penrose generalized pseudo inverse of a given matrix A of size $m \times n$, if

$$AGA = A, GAG = G, (AG)' = AG, (GA)' = GA.$$

In this case, we will denote *G* by A^+.

Definition 2.2: For the given general linear system of equations $Ax = y$ where A is a matrix of size $m \times n$ and *y* is a vector in R^m, a vector *x** in R^n is called a least squares solution if

$$|| Ax^* - y || = \min_x \; || Ax - y ||$$

Definition 2.3: The vector *x** in R^n is called a minimum norm least squares solution of the general linear system $Ax = y$, if *x** must be a least squares solution and further among all least squares solutions *x* in R^n

$$||x^*|| \;\; \leq \; ||x||$$

must be true.

Theorem 2.1 [10]: Let *G* be a matrix of size $n \times m$. Then $x^* = Gy$ is a minimum norm least squares solution of the general linear system $Ax = y$ if and only if $G = A^+$, the Moore-Penrose generalized inverse of *A*.

From the above theorem it is clear that $x^* = A^+ y$ is the unique minimum norm least squares solution.

III. Extreme Learning Machine Algorithm

Let us consider an *SLFN* having *L* number of hidden neurons. Let $G(.,.,.)$ be a real valued function so that $G(w_i, b_i, x)$ be the output of the i^{th} hidden neuron with bias $b_i \in \Re$ corresponding to the input vector $x \in \Re^m$ and the weight vector $w_i = (w_{i1}, ..., w_{im})'$ where w_{is} is the weight of the connection between the i^{th} hidden neuron and s^{th} neuron of the input layer. It is well known that for feed-forward neural networks, the output function $f(\cdot)$ will be given by

$$f(x) = \sum_{i=1}^{L} \beta_i G(w_i, b_i, x)\;,$$

where $\beta_i = (\beta_{i1}, ..., \beta_{ik})' \in \Re^k$ is the weight vector connecting the i^{th} hidden neuron with the k th neuron of the output layer. Note that for the case of additive hidden neurons, $G(.,.,.)$ will take the following form:

$$G(w_i, b_i, x) = g(w_i'.x + b_i)\,,$$

where $g : \Re \to \Re$ will be the activation function. In this work, we assume the case of additive hidden neurons.

Suppose, we are given the training data set $\{\, (x_i, y_i) \,\}_{i=1,2,...,M}$ where $x_i = (x_{i1}, ..., x_{im})' \in \Re^m$ denotes the input vector and $y_i = (y_{i1}, ..., y_{ik})' \in \Re^k$ is its corresponding output vector and M is the total number of input data patterns. Further assume that the values of the weight vectors $w_i \in \Re^m$ and the bias $b_i \in \Re$ are randomly assigned. Then, the standard *SLFN* with L number of hidden neurons approximates the input samples with zero error if and only if there exists $\beta_i \in \Re^k$ so that

$$y_j = \sum_{i=1}^{L} \beta_i G(w_i, b_i, x_j) \quad \forall j = 1,2,...,M\,. \tag{1}$$

The above set of equations can be rewritten in the following matrix form as:

$$H\beta = Y \tag{2}$$

where

$$H_{M \times L} = \begin{pmatrix} G(w_1, b_1, x_1) \ldots & G(w_L, b_L, x_1) \\ \vdots & \vdots \\ G(w_1, b_1, x_M) \ldots & G(w_L, b_L, x_M) \end{pmatrix}, \tag{3}$$

$$\beta_{L \times k} = \begin{pmatrix} \beta_1' \\ \vdots \\ \beta_L' \end{pmatrix} and \quad Y_{M \times k} = \begin{pmatrix} y_1' \\ \vdots \\ y_M' \end{pmatrix} \tag{4}$$

Note that the i^{th} column of *H* will be the output of the i^{th} hidden neuron for the inputs $x_1, x_2, ..., x_M$. Further, observe that the matrix H need not be a square matrix.

Under the assumption that the activation function $g(\cdot)$ is infinitely differentiable, it has been shown in [5] that for fixed input weight vectors w_i and biases b_i, the least squares solution $\hat{\beta}$ for the matrix equation (2) with minimum norm of output weights β can be obtained and that the smallest training error can be reached by the solution $\hat{\beta}$. Moreover, the solution $\hat{\beta}$ of the matrix equation (2) will be given by

$$\hat{\beta} = H^+ Y$$

where H^+ is the Moore-Penrose generalized pseudo inverse of the matrix *H*. Further, it has been reported in [5] that *ELM* tends to produce better generalization performance than *BP* with the main advantage being the decrease in computational time while training the network.

Training an *SLFN* is equivalent to obtaining a minimum norm least squares solution of the matrix equation $H\beta = Y$. In the course of learning, once the input weights and the hidden layer biases are randomly chosen they will not be adjusted at all. By Theorem 2.1, the smallest norm least-squares solution of the above learning machine is obtained when

$$\hat{\beta} = H^+ Y$$

Since *sin* and *sigmoid* are infinitely differentiable functions the *ELM* algorithm can be successfully applied by choosing any one of them as an activation function, and However, we studied the application of *ELM* algorithm also using *hardlim* activation function in all our experiments.

The *ELM* algorithm for *SLFN* can be stated [5] as follows:

Input: Training set $\{\ (x_i, y_i)\ \}_{i=1,2,...,M}$ where $x_i \in \Re^m$ and $y_i \in \Re^k$; L the number of hidden neurons and the activation function $g(\cdot)$.

1. For $i = 1,2,...,L$ randomly assign the input weight vector $w_i \in \Re^m$ and bias $b_i \in \Re$.

2. Determine the matrix *H* defined by the equation (3).

3. Calculate H^+.

4. Calculate the output weights matrix $\hat{\beta}$ by

$$\hat{\beta} = H^+ Y,$$

where Y is given by the equation (4).

Output: The Single hidden Layer Feedforward neural Network *(SLFN*) with the determined output weight vectors $\hat{\beta}_i \in \Re^k$ for the randomly chosen weight vectors $w_i \in \Re^m$ and biases $b_i \in \Re$ for $i = 1,2,...,L$.

For any input sample $x \in \Re^m$ the output value *y* can be calculated using the following formula:

$$y = \sum_{i=1}^{L} \hat{\beta}_i g(w_i'.x + b_i)$$

where w_i, b_i and the activation function *g(.)* are input and the weight vectors $\hat{\beta}_i \in \Re^k$ are the output of the *ELM* algorithm.

IV. Experiments and Results

A. Preprocessing of the Data

Time series prediction is the problem of determining a function having the underlying relationship between the previous values and the next value. Suppose N observations $\{x(i\tau)\}_{i=1,2,...,N}$ of the time series $x(t)$ are given with time delay τ. In all our experiments first the original data is normalized with zero mean and standard deviation equals to one. Then the normalized data is transformed into auto corrected data, i.e. for a given positive integer value m and $i = 1,...,(N-m)$ we define the auto corrected input vector $x_i = (x(i\tau), x((i+1)\tau),...,x((i+m-1)\tau))' \in \Re^m$ consists of the previous signal values. Here m is called the embedding dimension or memory order. The normalized auto corrected input vectors and their corresponding output values can be represented in the following matrix form

$$X = \begin{pmatrix} x(\tau) & x(2\tau) & \dots & x(m\tau) \\ x(2\tau) & x(3\tau) & \dots & x((m+1)\tau) \\ \vdots & \vdots & \vdots & \vdots \\ x((N-m)\tau) & x((N-m+1)\tau) & \dots & x((N-1)\tau) \end{pmatrix} \quad (5)$$

$$Y = \begin{pmatrix} x((m+1)\tau) \\ x((m+2)\tau) \\ \vdots \\ x(N\tau) \end{pmatrix} \quad (6)$$

respectively. Note that m determines the dimension of the input vectors of the *ELM* algorithm. The time series prediction problem may be stated as: for $i = 1,...,M$ we predict the target signal value $y_i = x((i+m)\tau) \in \Re$ corresponding to the auto corrected input vector $x_i \in \Re^m$. Observe that the number of neurons in the output layer is $k = 1$.

In order to demonstrate the effectiveness of *ELM* learning algorithm we have taken the time series generated by the Mackey Glass delay differential equation with different delays[2], Santa Fe A and UCR heart beat rate chaotic time series datasets. We have performed our experiments by choosing the *sigmoid, sin* and *hardlim* activation functions in the *ELM* learning algorithm. We use the Root Mean Square Error *(RMSE)* to evaluate the prediction performance of *ELM.* This is calculated using the following formula given by:

$$RMSE = \sqrt{\tfrac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2},$$

where n is the number of test data and y_i and $\tilde{y}_i$ are the actual and predicted values of the time series respectively. For Mackey Glass and Santa Fe A heart beat rate time series datasets, the first 70% of the total number of data values for training and the remaining data values for testing are used. However, for UCR time series dataset 60% of the total number of sample values for training and the remaining samples for testing are used. In all our experiments we applied the *ELM* source code[1] written in Matlab.

For choosing the memory order (m) and the number of hidden neurons (L) of the *ELM* network parameters, we vary m and L over a set of predefined values and determine the pair of values for *m* and *L* which gives the best performance based on the criteria of the *RMSE* on the test set. This is performed for each of the transfer functions, i.e. for *sigmoid*, *sin* and *hardlim* functions, and the best results obtained are reported.

B. $MG_{17,} MG_{30}$ Time Series

Consider the Mackey-Glass time delay differential equation [2,8] given by

$$\frac{\partial x(t)}{\partial t} = -bx(t) + a\frac{x(t-\tau)}{1+x(t-\tau)^{10}},$$

where *a, b* are parameters and τ is the time delay. We study the application of *ELM* algorithm on two time series generated

[1] http://www.ntu.edu.sg/home/egbhuang

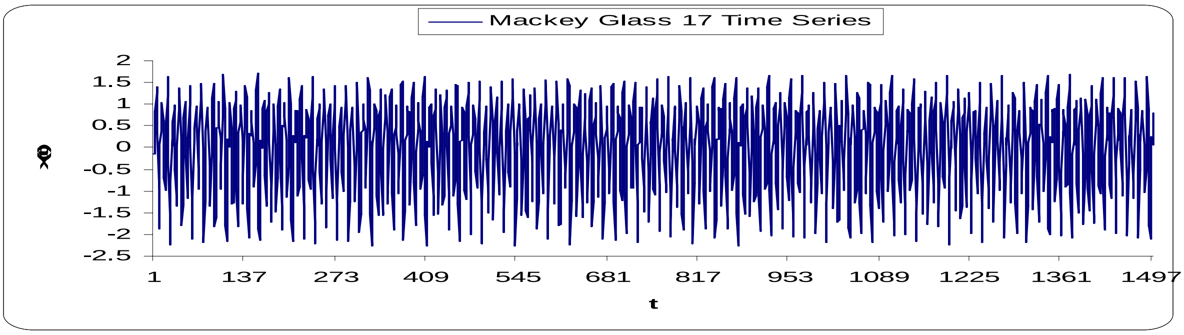

Fig. 1 The Mackey Glass Time Series with time delay $\tau = 17$

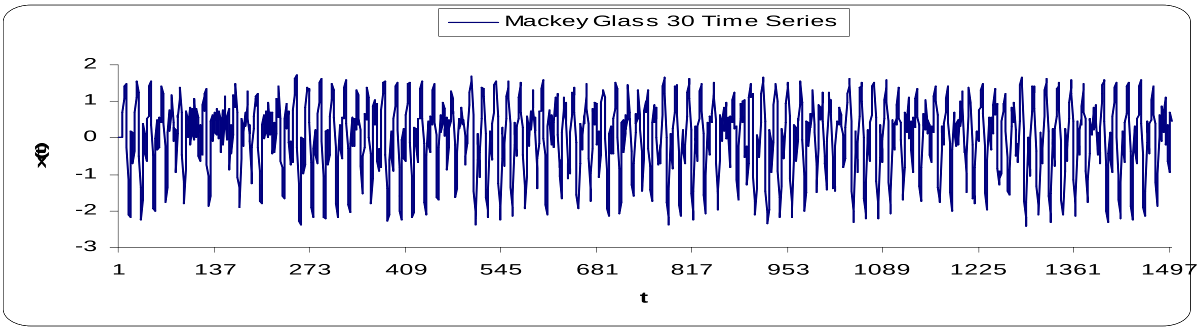

Fig. 2 The Mackey Glass Time Series with time delay $\tau = 30$

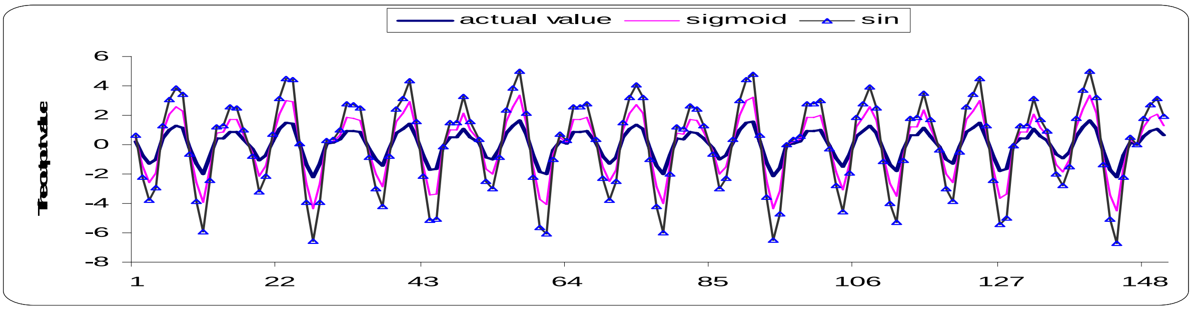

Fig. 3 Predicted result for $m = 5$ when using *sin* and *sigmoid* activation functions for MG_{17} time series corresponding to the time period from 1052 to 1201

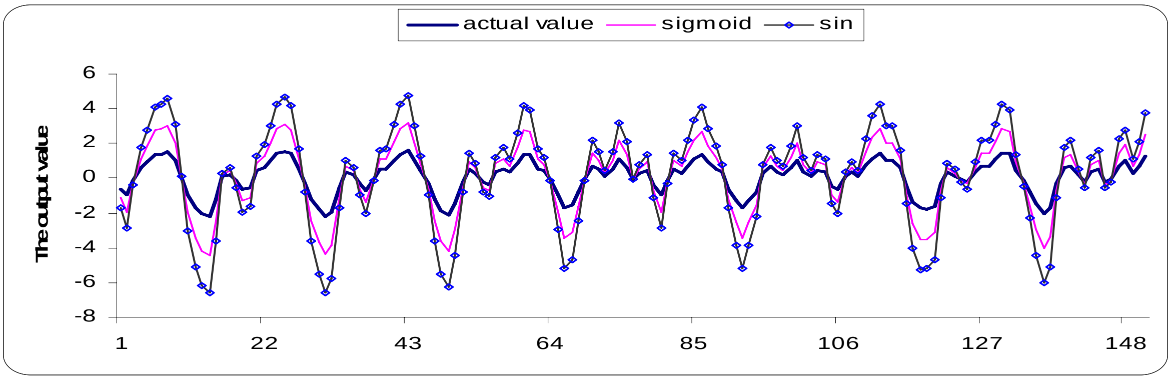

Fig. 4 Predicted result for $m = 7$ when *sin* and *sigmoid* activation functions are used for MG_{30} time series corresponding to the time period from 1052 to 1201

by the above differential equation which are widely used as benchmark data set values for analyzing the generalization ability of the method of prediction. For this, consider the

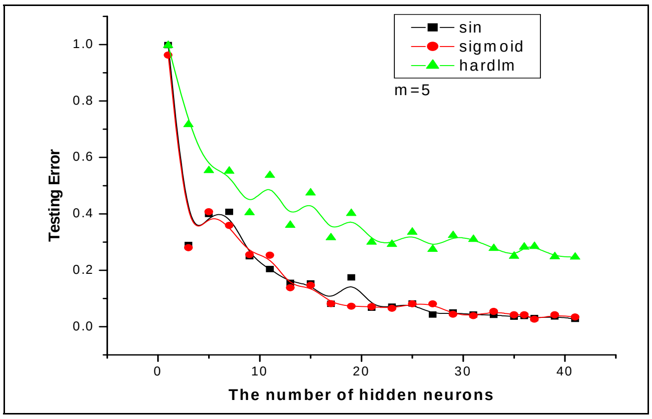

Fig. 5 Error plot for MG_{17} time series when different activation functions are used with memory order $m = 5$

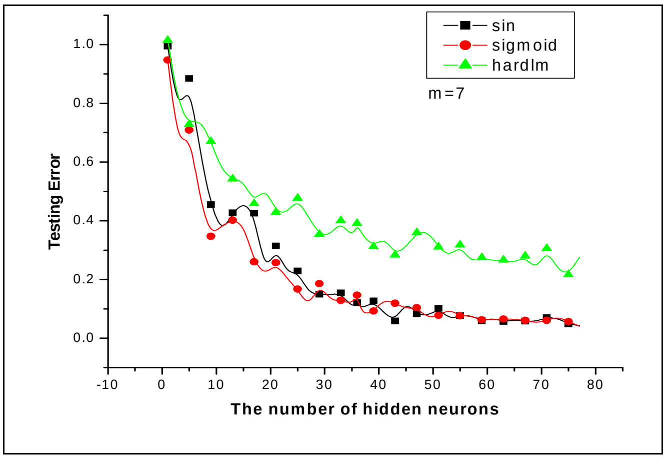

Fig. 6 Error plot for MG_{30} time series when different activation functions are used with memory order $m = 7$

haotic time series [2,8] generated using the parameter values =*0.2*, *b=0.1* and $\tau = 17,30$ where τ is the time delay. Let s call these time series corresponding to $\tau = 17,30$ as IG_{17} and MG_{30} respectively.

In order to avoid the initialization transients, the initial 500 samples are discarded [8]. We considered 1050 data oints corresponding to the sample time period from 3501 to 550 for training and the sample time period from 4551 to 000 for testing. In Fig. 1 and Fig. 2 we have shown the time eries data values from 1 to 1500 (after discarding the initial 3500 samples) for MG_{17} and MG_{30} time series respectively where the first 1050 samples are taken for training and the remaining 450 samples for testing. Experiments were performed using all the activation functions namely *sin*, *sigmoid* and *hardlim* functions. As it was discussed earlier, the best prediction performance for MG_{17} time series was obtained by varying the memory order $m = \{5,7,9\}$ and the number of hidden neurons $L = \{1,3,...,41\}$ for the choice of each one of the above activation functions. It was found that the best prediction performance was obtained for the *sigmoid* activation function having its corresponding values of *m* and *L* being *m = 5* and *L = 37* respectively. In Fig. 3 we have shown the actual and the predicted time series of MG_{17} for the time period from 1052 to 1201. The *RMSE* curve is plotted for all the activation functions with the memory order *m=5* on the test data set in Fig. 5.

Similarly for MG_{30} times series, by varying the memory order $m = \{5,7,9\}$ and the number of hidden neurons $L = \{1,3,...,77\}$ for the choice of each one of the activation functions, the best prediction performance was obtained again for the *sigmoid* activation function but the corresponding values of the memory order and the number of hidden neurons being $m = 7$ and $L = 77$ respectively. In Fig. 4 we have shown the actual and the predicted time series of MG_{30} for the time period from 1052 to 1201. Also we have plotted the *RMSE* curve obtained for each of the activation functions with the memory order *m=7* on the test data set in Fig. 6.

C. Santa Fe-A Time Series

This is a laser time series data set shown in Fig. 7 recorded from a Far-Infrared-Laser in a chaotic state, which is approximately described by three coupled non-linear ordinary

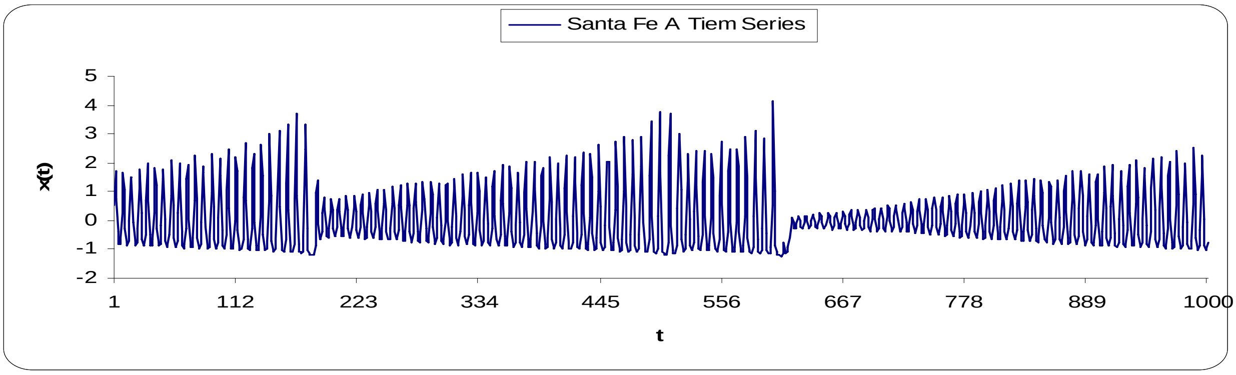

Fig. 7 The Santa Fe A Laser Time Series

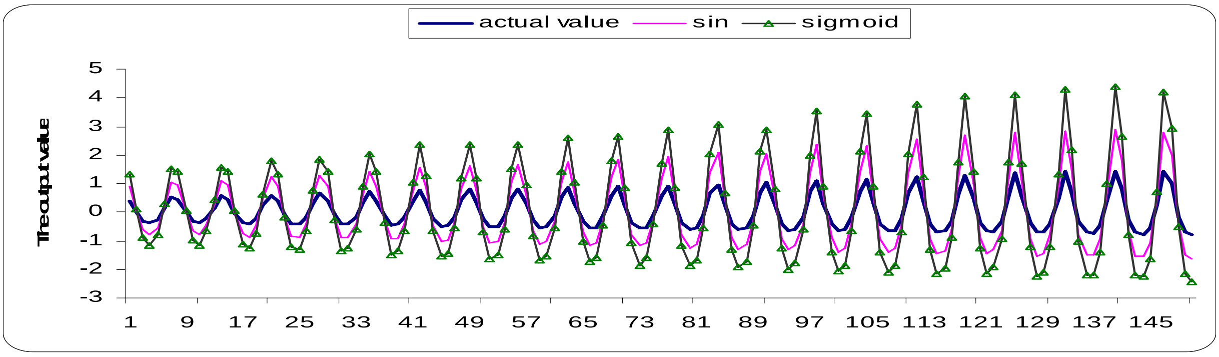

Fig. 8 Predicted result for $m = 5$ when using sin and *sigmoid* activation functions on Santa Fe-A time series corresponding to the time period from 702 to 851

differential equations[2]. The data set contains 1000 data points (See Fig. 7). Among them the first 700 data points will be used for training and the remaining 300 points for testing. By varying m = {3,5,7} and L = {1,3,…,81} it was determined that the best prediction performance was obtained for the *sin* activation function for which the corresponding values of the memory order and the number of hidden neurons being m=5 and L= 49 respectively. In Fig. 8, we have plotted the actual and the predicted values for *sin* and *sigmoid* activation functions corresponding to the time period from 702 to 851. Finally, for m = 5 we have shown the *RMSE* values for all the activation functions in Fig. 9.

D. UCR Time Series Datasets

We repeat our experiments on time series datasets from a diverse set of domains available from UCR Time Series Data Mining Archive[3]. We consider two important time series datasets both of human heart beat.

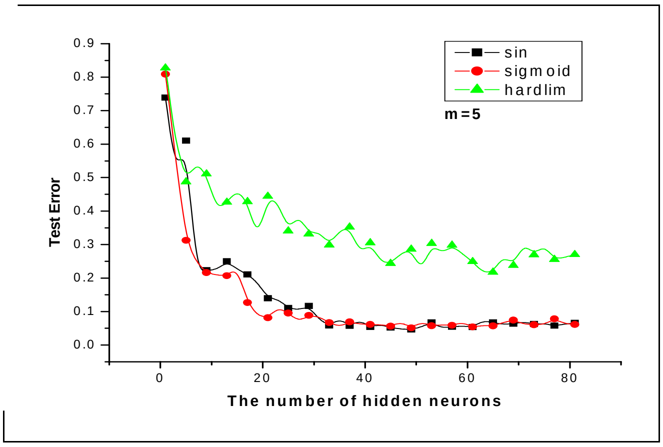

Fig. 9 Error plot for Santa Fe-A time series for each of the activation function is used with memory order $m = 5$

i. The Time Series of Human Heart Beat

First let us consider the ECG time series of human heart beat dataset shown in Fig. 10. ECGs are time series of the electrical potential between two points on the surface of the body caused by a beating heart. This time series data set consists of 3751 sample values. In our experiment data points from 1 to 2251 are taken as the training set and the remaining points from 2252 to 3751 as the test set. As it was explained earlier, by varying the memory order $m = \{3,4,5,6,7,8\}$ and the number of hidden neurons $L = \{1,...,41\}$, the best prediction performance was obtained for the choice of sigmoid activation function for which the corresponding values of the memory order and the number of hidden neurons being $m = 5$ and $L = 41$ respectively. From Fig. 11 we observe that the predicted values are in close agreement with the actual values.

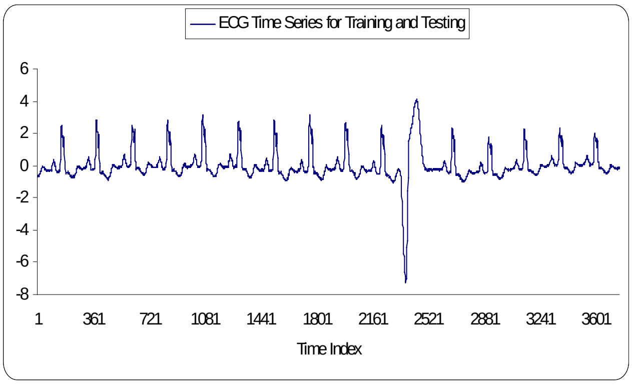

Fig. 10 UCR time series of human heart beat electro cardio- gram

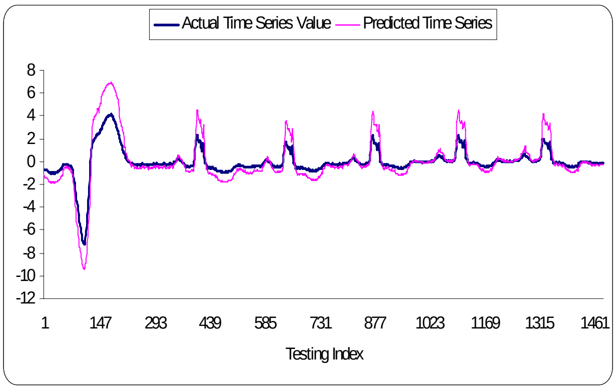

Fig. 11 Predicted and actual values of human heart beat ECG time series for the choice of *sigmoid* activation function where $m = 5$ and L = 41

[2] This Time Series is available on: httt://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html
[3] http://www.cs.ucr.edu/~eamonn/time_series_data.

ii. The Time Series of Human Heart Beat (Second dataset)

This is the second ECG time series used in our experiment and is shown in Fig. 12. It consists of 3750 data values. In this example, data values from 1 to 2251 are considered for training and the remaining data values from 2252 to 3750 as the test set. By varying the memory order $m = \{3,4,5,6,7,8\}$ and the number of hidden neurons $L = \{1,...,81\}$,the best prediction performance was obtained for the choice of sigmoid activation function having its corresponding values of the memory order and the number of hidden neurons being $m = 6$ and $L = 61$ respectively.

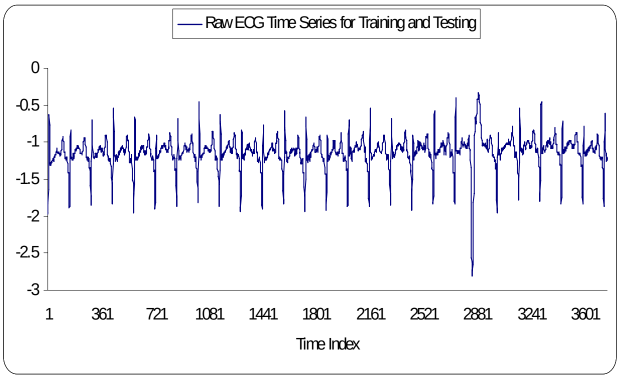

Fig. 12 UCR time series (second dataset) of human heart beat electro- cardiogram

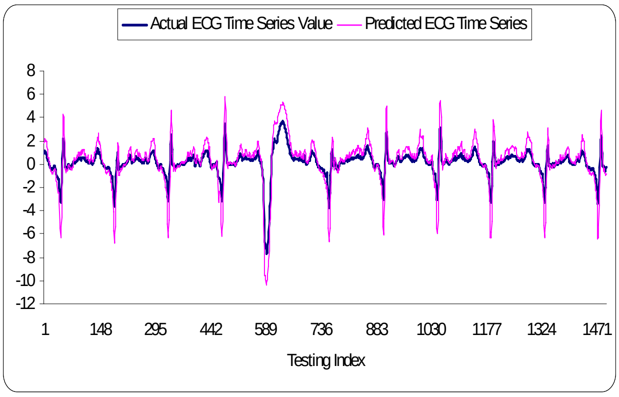

Fig. 13 Predicted and actual values of human heart beat (second ataset) ECG time series for the choice of *sigmoid* activation function where $m = 6$ and L = 61

Fig. 13 illustrates the predicted and the actual values for the ›st set where the predicted values obtained using *ELM* and the actual values are shown in thin and thick solid lines respectively. The results show that the predicted values are in close agreement with the actual values.

V. Conclusion

In this paper, we studied the application of Extreme Learning Machine algorithm for chaotic time series generated by the Mackey Glass delay differential equation with different time delays, Santa Fe A and UCR heart beat rate ECG time series. We performed our experiments using *sigmoid*, *sin* and *hardlim* activation functions and demonstrated that the *ELM* algorithm using *sin* and *sigmoid* activation functions can achieve high prediction accuracy. Also from our study we conclude that *ELM* is a promising method for time series prediction problems.

References

[1] P. J. Brockwell and R. A. Davis, "*Introduction to Time Series Forecasting*", 2nd ed., Springer, Berlin, 2002.

[2] M.Casdagli, "Nonlinear Prediction of Chaotic Time Series", Physica D, 35, (1989), pp. 335-356.

[3] K. Y. Chen and C. H. Wang, "A Hybrid SARIMA and Support Vector Machines for Forecasting the Production Values of the Machinery Industry in Taiwan", Expert Systems with Applications, (2006).

[4] Y. Chen, B.Yang and J.Dong, "Time Series Prediction using a Local Linear Wavelet Neural Network", Neurocomputing, 69 (2006), pp.449-465.

[5] G. B. Huang, Q. Y. Zhu and C. K. Siew, "Extreme Learning Machine: Theory and Applications", Neurocomputing, 70, (2006), pp.489-501.

[6] R.Malhotra and D.K.Malhotra, "Evaluating Consumer Loans Using Neural Networks", Omega, 31, (2003), pp.83-96.

[7] N.Mani and P.Voumard, "An Optical Character Recognition Using Artificial Neural Network", IEEE Int. Conf. on Systems, Man, and Cybernetics, Vol. 3, (1996), pp.2244-2247.

[8] S.Mukherjee, E.Osuna and F.Girosi, "Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines", *in* Neural Networks for Signal Processing VII, Proceed. of the IEEE Signal Processing Society Workshop, FL, (1997), pp.511-520.

[9] K.R.Muller, A.J.Smola, G.Ratsch, B.Schlkopf and J.Kohlmorgen, "Using Support Vector Machines for Time Series Prediction", in B.Schlkopf, C.J.C. Burges and A.J.Smola (Eds), Advances in Kernel Methods- Support Vector Learning, MIT Press, Cambridge, MA, (1999), pp.243-254.

[10] C.R.Rao and S.K.Mitra, *Generalized Inverse of Matrices and its Applications*, Wiley, New York, (1971).

[11] H.A.Rowley, S.Baluja and T.Kanade, "Neural Network based Face Detection", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.20, No. 1, (1998), pp.23-38.

[12] Z.Tang, P.A.Fishwick, "Feedforward Neural Nets as Models for Time Series Forecasting", ORSA J. Comput. 5(1993), pp.374-385.

[13] F.E.H.Tay and L.Cao, "Application of Support Vector Machines in Financial Time Series Forecasting", Omega 29 (2001), pp.309-317.

[14] Q.Tong, H.Zheng and X.Wang, "Gene Prediction Algorithm Based on the Statistical Combination and the Classification in terms of Gene Characteristics", Int. Conf. on Neural Networks and Brain, Vol.2, (2005), pp.673 – 677.

[15] T.B.Trafalis, H.Ince, "Support Vector Machine for Regression and Applications to Financial Forecasting", Proceedings of the IEEE INNSENNS Int. Joint Conf., Vol.16, IEEE (2000), pp. 348-353.

[16] F.M.Tseng, H.C.Yu and G.H.Tzeng, "Combining Neural Network Model with Seasonal Time Series ARIMA Model", Technological Forecasting and Social Change, 69 (2002), pp.71-87.

[17] G.P.Zhang, E.B.Patuwo and M.Y.Hu, "A Simulation Study of Artificial Neural Networks for Nonlinear Time Series Forecasting", Comput.Oper.Res. 28,(2001), pp.381-396.

[18] G.P.Zhang, "Time Series Forecasting using a Hybrid ARIMA and Neural Network Model", Neurocomputing, 50, (2003),pp. 159-175.