Multi-Step Ahead Prediction Of Chaotic Time Series Using Recurrent Neural Network Models

Sanjay L. Badjate

S.B. Jain Institute of Technology, Management and Research, Nagpur (M.S),India

s_badjate@rediffmail.com

Sanjay M.Gulhane

Jawaharlal Darda Institute of Engineering and Technology, Yavatmal (M.S), India

smgulhane67@rediffmail.com

ABSTRACT

In this paper the multi step ahead prediction of typical Duffing Chaotic time series and the monthly sunspots real time series are carried out. These two time series are popularized due to their highly chaotic behavior. This paper compares the performance of two neural network configurations namely a Multilayer Perceptron (MLP) and proposed FTLRNN with gamma memory for the duffing time series for 1, 5,10,20,50 and 100-step ahead prediction and for monthly sunspot time series for 1, 6, 12, 18 & 24 month ahead prediction . The standard back propagation algorithm with momentum term has been used for both the models.

It is seen that estimated dynamic fully recurrent model clearly outperforms the MLP NN in various performance matrices such as Mean square error (MSE), Normalized mean square error (NMSE) and correlation coefficient (r) on testing as well as training data set for multi step prediction (K=1,5,10,20,50,100) for duffing time series and for the sunspot time series for 1, 6, 12, 18 &24 month ahead prediction. In addition, the output of proposed neural network model closely follows the desired output for all the step ahead prediction. It is observed that suggested recurrent models have the remarkable capability of time series prediction. The major contribution of this paper is that Various parameters like number of processing elements, step size, momentum value in hidden layer, in output layer the various transfer functions like tanh, sigmoid, linear-tan-h and linear sigmoid, different error norms L1, L2 ,Lp to L∞.

Keywords: Chaotic, Multistep-prediction

1 INTRODUCTION

Neural network have been widely applied to the prediction problem[1]. Examples range from forecasting the stock market and weather [1] to speech coding [2] and noise cancellation [3] .Predicting a chaotic time series using a neural network is of particular interest[4-6]. Not only it has an efficient method to reconstruct a dynamical system from an observed time series, but it also has many applications in engineering problems such as speech coding [7], radar detection ,modeling of electromagnetic wave propogation and scattering [8], [9]. The main motivation for analysis and research of chaotic time series is to predict the future and understand the fundamental feature and proceses in system, which are used in every sector of the human life. Recognizing chaotic dynamics is potentially important for understanding and managing real world problems.

One of the primary reasons for employing neural network was to create a machine that was able to learn from experience [10] They have the capability to learn the complex nonlinear mappings from a set of observations and predict the future.

The modeling and analysis of chaotic time series has also attracted the attention of many researchers. The static MLP network has gained an immense popularity from numerous practical application published over the past decade, there seems to be substantial evidence that Multilayer Perceptron indeed possesses an impressive ability .There have been some theoretical results that try to explain the reasons for the success [11] and [12]. However, there are some limitations of this famous static neural network configuration .It cannot cope up with rapidly changing nonlinear dynamics [10]. Therefore it is necessary a NN configuration that can learn the temporal variation or structure underlying the data in true sense. In view of this, dynamic modeling will certainly help to improve the learning and generalization ability. Hence, in this research work, a dynamic NN topology is developed so as to explicitly include time relationships in the input and output mappings.

In this paper a novel focused time lag recurrent neural network with gamma memory is proposed as an intelligent tool for predicting duffing time series for multi step (1,5,10,20,50 &100) ahead prediction and fully recurrent network model is an intelligent tool for predicting the real time sunspot monthly time series for 1,6,12,18 and 24 month ahead prediction .This paper is organized as follows. First the optimal static NN based model on MLP is attempted to model the given system. Later the best dynamic time NN model with built in gamma memory is estimated for predictions. For sunspot monthly time series the different Neural network models are attempted . Next a comparison between these models are carried out on the basis of the performance measures such as Mean square error (MSE), Normalized mean square error (NMSE) and correlation coefficient (r) on testing as well as training data set for Multi step ahead prediction (K=1,5,10,20,50,100).The various parameters like number of hidden layers, number of processing elements, step size, momentum value in hidden layer, in output layer the various transfer functions like tanh, sigmoid, linear-tan-h and linear sigmoid, different error norms L1,L2 ,Lp to L∞, Epochs variations and different combination of training and testing samples are exhaustively experimented for obtaining the proposed robust model for long term (k=20,50,100) step prediction and short term (1,5,10) prediction for duffing time series and 1,6,12 18 & 24 months ahead for monthly sunspot time series Finally, the conclusion are discussed with a recommendation to use proposed recurrent neural network configuration respectively.

2 STATIC NN BASED MODEL

Static NN s typically use MLP as a backbone. They are layered feed forward networks typically trained with static back

propagation . MLP solid based model has a solid foundation [24 25]. The main reason for this is its ability to model simple as well as complex functional relationships. This has been proven through number of practical applications [13].In [14] it is shown that all continuous functions can be approximated to any desired accuracy, in terms of the uniform norm, with a network of one hidden layer of sigmoidal or (hyperbolic tangent) hidden units and a layer of linear or tan h output unit to include in the hidden layer. .This is discussed in [13] and a significant result is derived approximation capabilities of two layer perceptron networks when the function to be approximated shows certain smoothness. The biggest advantage of using MLP NN for approximation of mapping from input to the output of the system resides in its simplicity and the fact that it is well suited for online implementation [14] .The objective of training is then to determine a mapping from a set of training data to the set of possible weights so that the network will produce predictions y(t),which in some sense are close to the true outputs y(t). The prediction error approach is based on the introduction of measure of closeness in terms of mean square error (MSE) criteria:

$$V_N(\theta, Z^N) = 1/2N \sum_{t=1}^{N} [y(t) - y\hat{}(t \mid \theta]^2$$

$$= (1/2N) \sum_{t=1}^{N} \varepsilon^2 (t, \theta) - - - - - (1)$$

The weights are then founds as: $\theta\hat{}$ = arg $\min_0 V_N(\theta, Z^N)$, by some kind of iterative minimization schem: $\theta^{(i+1)} = \theta^{(i)} + \mu^{(i)} f^{(i)}$,

Where $\theta^{(i)}$ specifies the current iterate (number "i"), $f^{(i)}$ is the search direction and $\mu^{(i)}$ the step size.

When NN has been trained, the next step is to evaluate it . There is no specific rule governing the splitting of data in the literature [27].However this is done by standard method in statistics called independent validation. This method divides the available datasets into training and testing data sets. This method divides the available data sets into two sets namely training data set and testing data set .The training data set are next divide into two partitions: the first partition is used to update the weights in the network and the second partition is used to assess (or cross validate) the training performance .The testing data set are then used to assess how the network has generalized .The learning and generalization ability of the estimated NN based model is assessed on the basis of certain performance measures such as MSE, NMSE and the regression ability of the NN by visual inspection of the regression characteristics for different outputs of system under study.

Since it is very likely that one ends up in a bad local minimum, the network is trained couple of times (typically least three times) starting from different initial weights. Neuro solution

(version5) and MATLAB tool box (version7.0) are use for obtaining the results.

3 PERFORMANCE MEASURES MSE

The generalization performance of the network is valuated on the basis of following parameters [14]

The mean square error is given by

$$\frac{\sum_{j=0}^{p} \sum_{i=0}^{N} (d_{ij} - y_{ij})^2}{NP} \quad \text{MSE} = \text{-----(2)}$$

Where P = number of output PEs (processing elements) , N=number of exemplars in the data set ,y_{ij} =network output for exemplar i at PEj , d_{ij}=desired output for exemplar i at PEj.

NMSE (Normalized Mean square Error)

The normalized mean square error is defined by the following formula:

Where P=Number of output PEs,

N= Number of exemplars in data set,

MSE =Mean square error, d_{ij}= desired output for exemplar i at pej

$$NMSE = \frac{P * N * MSE}{\sum_{j=0}^{p} \left[\frac{N \sum_{i=0}^{N} dij^2 - \sum_{i=0}^{2} dij}{N} \right]} \quad \text{------- (3)}$$

[c] The size of the mean square error (MSE) can be used to determine how well the network output fits the desired output , but it doesn't necessarily reflect whether the two sets of data move in the same direction. For instance by simply scaling the network output , we can change the MSE without changing the directionality of the data. The correlation coefficient solves this problem. By definition, the correlation coefficient between a network output x and a desired output d is.

$$r = \frac{\frac{\sum_i (xi - \bar{x})(di - \bar{d})}{N}}{\sqrt{\frac{\sum_i (d - \bar{d})}{N}} \sqrt{\frac{\sum_i (xi - x)}{N}}}$$

where,

$$\bar{X} = \frac{1}{N} \sum^{N} Xi \quad \text{and} \quad \bar{d} = \frac{1}{N} \quad \text{----(4)}$$

The correlation coefficient is confined to the range [-1,1] .When r =1 ,there is perfect positive linear correlation between x & d that is they co-vary means they vary by the same amount ,when r=-1 ,there is a perfectly linear negative correlation between x and d that is they vary in opposite ways(when x increases ,d decreases that is they varies in opposite ways) and when r=0 there is no correlation between x and d i.e. the variables are called uncorrelated. Intermediate values describe partial correlations.

4 FTLRNN MODEL

Time lagged recurrent networks (TLRNs) are MLPs extended with short term memory structures. Here, a "static" NN (e.g., MLP) is augmented with dynamic properties [15 dudul liquid]. This, in turn, makes the network reactive to the temporal structure of information bearing signals. For a NN to be dynamic, it must be given memory. This memory may be classified into "short-term" and "long-term" memory. Long term memory is built into a NN through supervised learning, whereby the information content of the training data set is stored (in part or in full) in the synaptic weights of the network[16 de Varies B & J C Principe] .

However, if the task at hand has a temporal dimension, some form of "short-term" memory is needed to make the network dynamic. One simple way of building short-term memory into the structure of a NN is through the use of time delays, which can be applied at the input layer of the network (focused). A short-term memory structure transforms a sequence of Samples into a point in the reconstruction space [16]. This memory structure is incorporated inside the learning machine. This means that instead of using a window over the input data, PEs created are dedicated to storing either the history of the input signal or the PE activations.

The input PEs of an MLP are replaced with a tap delay line, which is followed by the MLP NN. This topology is called the focused time-delay NN (TDNN) . The focused topology only includes the memory kernels connected to the input layer**.** This way, only the past of the input is remembered. The delay line of the focused TDNN stores the past samples of the input. The combination of the tap delay line and the weights that connect the taps to the PEs of the first hidden layer are simply linear combiners followed by a static non-linearity.

Typically, a gamma short-term memory mechanism is combined with nonlinear PEs in restricted topologies called focused. Basically, the first layer of the focused TDNN is a filtering layer, with as many adaptive filters as PEs in the first hidden layer. The outputs of the linear combiners are passed through a non linearity (of the hidden-layer PE) and are then further processed by the subsequent layers of the MLP for system identification, where the goal is to find the weights that produce a network output that best matches the present output of the system by combining the information of the present and a predefined number of past samples (given by the size of the tap delay line) [17 Gamma modal].

Size of the memory layer depends on the number of past samples that are needed to describe the input characteristics in time. This number depends on the characteristics of the input and the task. This focused TDNN can still be trained with static back-propagation, provided that a desired signal is available at each time step. This is because the tap delay line at the input layer doesn't have any free parameters. So the only adaptive parameters are in the static feed forward path.

The memory PE receives in general many inputs, $x_1(n)$ and produces multiple outputs $y = [y_0(n), \ldots, y_D(n)]T$, which are delayed versions of $y_0(n)$ the combined input,

$$y_k(n) = g(y_{k-1}(n))\ y_0(n) = \sum_{j=1}^{P} x_j(n) \qquad \text{---(5)}$$

where, g(.) is a delay function.

These short-term memory structures can be studied by linear adaptive filter theory if g(.) is a linear operator. It is important to emphasize that the memory PE is a short-term memory mechanism, to make clear the distinction from the network weights, which represent the long-term memory of the network.

There are basically two types of memory mechanisms: memory by delay and memory by feedback. We seek to find the most general linear delay operator (special case of the Auto Regressive Moving Average model) where the memory traces $y_k(n)$ would be recursively computed from the previous memory trace yk-1(n). This memory PE is the generalized feed forward memory PE. It can be shown that the defining relationship for the generalized feed forward memory PE is mentioned .

$$g_k(n) = g(n) * g_{k-1}(n) \text{ k}\geq 1 \qquad \text{-(6)}$$

where * is the convolution operation, g(n) is a causal time function, and k is the tap index. Since this is a recursive equation, $g_0(n)$ should be assigned a value independently. This relationship means that the next memory trace is constructed from the previous memory trace by convolution with the same function g(n), the memory kernel yet unspecified. Different choices of g(n) will provide different choices for the projection space axes. When we apply the input x(n) to the generalized feed forward memory PE, the tap signals $y_k(n)$ become

$y_k(n)=g(n)*y_{k-1}(n) \quad k\geq 1$ ------(7)

the convolution of yk – 1(n) with the memory kernel. For k=0 we have

$y_0(n)=g_0(n)*x(n)$ ------(8)

where $g_0(n)$ may be specified separately. The projection x(n) of the input signal is obtained by linearly weighting the tap signals according to

$$x(n) = \sum_{k=0}^{D} w_k y_k(n) \qquad \text{-------(9)}$$

The most obvious choice for the basis is to use the past samples of the input signal x(n) directly, that is the kth tap signal becomes $y_k(n) = x(n – k)$. This choice corresponds to $g(n) = \delta(n – 1)$ ------(10)

In this case $g_0(n)$ is also a delta function δ(n) (delta function operator used in the tap delay line). The memory depth is strictly controlled by D, that is the memory traces store the past D samples of the input. The time delay NN uses exactly this choice of basis.

The gamma memory PE attenuates the signals at each tap because it is a cascade of leaky integrators with the same time constant gamma modal.] The gamma memory PE is a special case of the generalized feed forward memory PE where

$$g(n) = \mu(1-\mu)^n \quad \text{n} \geq 1 \qquad \text{---------(11)}$$

and $g_0(n) = \delta(n)$. The gamma memory is basically a cascade of lowpass filters with the same time constant 1 - μ. The over all impulse response of the gamma memory is

$$g_p(n) = \binom{n-1}{p-1}\mu^P(1-\mu)^{n-p}, \text{ n} \geq \text{p} \qquad \text{--------(12)}$$

where (:) is a binomial coefficient defined by

$$\binom{n}{p} = \frac{n(n-1)....(n-p+1)}{p!}$$ for integer values of n and p, the overall impulse response $g_p(n)$ for varying p represents a discrete version of the integrand of the gamma function [17], hence the name of the memory.

The gamma memory PE has a multiple pole that can be adaptively moved along the real Z-domain axis, that is the gamma memory can implement only low pass ($0 < \mu < 1$) or high pass ($1 < \mu < 2$)

transfer functions. The high pass transfer function creates an extra ability to model fast-moving signals by alternating the signs of the samples in the gamma PE (the impulse response for $1 < \mu < 2$ has alternating signs). The depth in samples parameters (D) is used to compute the number of taps (T) contained within the memory structure(s) of the network.

An exhaustive and careful experimental study has been carried out to determine the optimal parameters of the proposed FTLRNN model. It is seen that the performance of this model is optimal on the test dataset for the following Wples = 10, , No. of taps = 6, Tap Delay = 1. Trajectory Length = 50 .

4.1 Jordan/ Elman NN

The theory of neural networks with context units can be analyzed mathematically only for the case of linear PEs. In this case the context unit is nothing but a very simple lowpass filter. A lowpass filter creates an output that is a weighted (average) value of some of its more recent past inputs. In the case of the Jordan context unit, the output is obtained by summing the past values multiplied by the scalar as shown in the figure below.

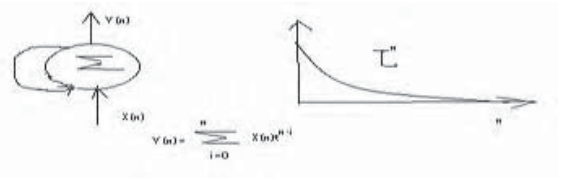

Fig Context unit response

Notice that an impulse event x(n) (i.e. x(0)=1, x(n)=0 for n>0) that appears at time n=0, will disappear at n=1. However, the output of the context unit is t1 at n=1, t2 at n=2, etc. This is the reason these context units are called memory units, because they "remember" past events. t should be less than 1, otherwise the context unit response gets progressively larger (unstable).

The Jordan network and the Elman network combine past values of the context units with the present inputs to obtain the present net output. The input to the context unit is copied from the network layer, but the outputs of the context unit are incorporated in the net through adaptive weights.. One issue in these nets is that the weighting over time is kind of inflexible since we can only control the time constant (i.e. the exponential decay). Moreover, a small change in t is reflected in a large change in the weighting (due to the exponential relationship between time constant and amplitude). In general, we do not know how large the memory depth should be, so this makes the choice of t problematic, without a mechanism to adapt it.

The Neural Wizard provides four choices for the source of the feedback to the context units (the input, the 1st hidden layer, the 2nd hidden layer, or the output). In linear systems the use of the past of the input signal creates what is called the moving average (MA) models. They represent well signals that have a spectrum with sharp valleys and broad peaks. The use of the past of the output creates what is called the autoregressive (AR) models. These models represent well signals that have broad valleys and sharp spectral peaks. In the case of nonlinear systems, such as neural nets, these two topologies become nonlinear (NMA and NAR respectively). The Jordan net is a restricted case of an NAR model, while the configuration with context units fed by the input layer are a restricted case of NMA. Elman's net does not have a counterpart in linear system theory. As you probably could gather from this simple discussion, the supported topologies have different processing power, but the question of which one performs best for a given problem is left to experimentation.

4.2 Recurrent Neural network

Fully recurrent networks feed back the hidden layer to it self . Partially recurrent networks start with a fully recurrent net and add a feed forward connection that bypasses the recurrence, effectively treating the recurrent part as a state memory. These recurrent networks can have an infinite memory depth and thus find relationships through time as well as through the instantaneous input space. Most real-world data contains information in its time structure. Recurrent networks are the state of the art in nonlinear time series prediction, system identification, and temporal pattern classification.

4.3 Self Organizing Feature Map

Kohonen's SOFM (Self Organizing Feature Map) proposed by Kohonen is a two layered network have been used in some application with great deal of success .Based on unsupervised learning ,they have been greatly used in exploratory tasks[18]. There are several approaches that adopt the basic algorithm in order to support the time sequence processing. This way of facing problem, that some authors denominate temporal SOM [19] have been used in wide variety of application such as prediction. The first layer of network layer is the input layer. Typically second competitive layer is organized as a two-dimensional grade. All the dimensions go from first layer to the second layer. The two layers are fully interconnected as each unit is connected to all of the unit in the competitive layer. When an input pattern is presented each unit in first layer takes on the values at the corresponding entry in the input pattern .The second layer unit then sums their input & compete to find a single winning unit the over all operation of SOFM (Self Organizing Feature Map) is similar to the competitive learning paradigm. Each inter connection in the Kohonen has an associated weight values for axon [20] . The initial stage at network has randomized values for the weights. Typically the weights are sot by adding a small random number at average values for the entries in the input pattern.

The structure consists of two main network & back propagation network. Every axon and the synapse network a corresponding back axon and back synapse that attach to the upper right component of the corresponding forward component. Data flows forward from input to output. The forward propagation network. The criteria compares the output with the desired response and computes the error. The error is then injected in to the back propagation network and the data flows through this network, back towards the original input.

4.4 Duffing Time Series

In science, chaos is used as a synonym for irregular behavior, whose long term prediction is essentially unpredictable chaotic differential equations exhibits not only irregular behavior but they are also unstable with respect to small perturbations of their initial condition [21 two layer perceptron dudul]. Consequently it is difficult to forecast the future of time series based on chaotic differential equations; they should be a good bench mark for a neural network design algorithm.

The duffing equation is time delay differential equation which is given as

$$\frac{dy}{dt} = y \qquad \text{----------(13)}$$

$$\frac{dy}{dt} = \left\{F^{x} Cos(\tau) - x^{3} - b^{x} y\right\} \qquad \text{-------(14)}$$

$$\frac{d\tau}{dt} = w \qquad \text{----------(15)}$$

Where driving force F=7.5, Xo=initial position 1.0, damping constant (b=0.05),Frequency w =1.0,Delay time=0.001.The chaotic time series is as shown in fig 1.

A sunspot is a region on the Sun surface (photosphere) that is marked by a lower temperature than its surroundings and has intense magnetic activity, which inhibits convection forming areas of low surface temperature. Although they are blindingly bright at temperatures of roughly 4000-4500 K, the contrast with the surrounding material at about 5800 K leaves them clearly visible as dark spots. If they were isolated from the surrounding photosphere they would be brighter than an electric arc. A minimum in the eleven-year sunspot cycle may have taken place in late 2007 and start of cycle 24 is expected in 2008.Sunspots are often related to intense magnetic activity such as coronal loops and reconnection. Most solar flares and coronal mass ejection originate in magnetically active regions around sunspot groupings [24]. Sunspot numbers rise and fall with an irregular cycle with a length of approximately 11 years. In addition to this, there are variations over longer periods. The recent trend is upward from 1900 to the 1960s, then somewhat downward. The Sun was last similarly active over 8,000 years ago. The number of sunspots has been found to correlate with the intensity of solar radiation over the period (since 1979) when satellite measurements of radiation are available. Since sunspots are dark it might be expected that more sunspots lead to less solar radiation and a decreased solar constant. However, the surrounding areas are brighter and the overall effect is that more sunspots means a brighter sun. The variation caused by the sunspot cycle to solar output is relatively small, on the order of 0.1% of solar constant. The data considered the monthly variations from January 1949 to December 2006 .The total samples are 3097 considered.

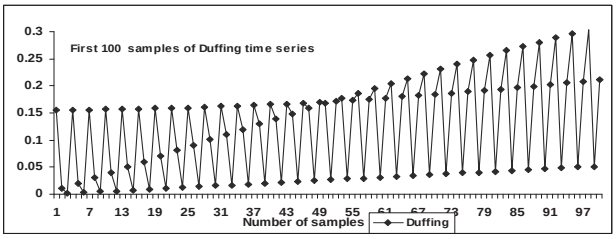

Fig. 1 Duffing time series

Sun spot Time series

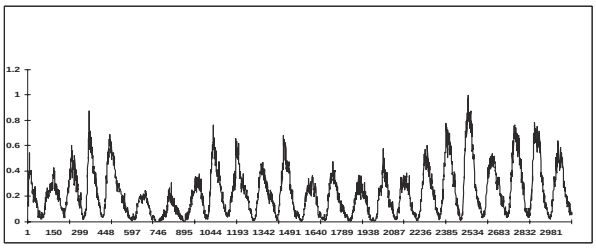

Fig. 2 Sun spot time series

5 EXPERIMENTAL RESULTS

5.1 Duffing time series

An exhaustive and careful experimentation has been carried to determine the configuration of the static MLP Model and the optimal proposed models for all the step (K=1, 5.10,20,50,100) prediction for the duffing time series and the sun spot time series . All the possible variations for the model such as number of hidden layers, number of processing elements in each hidden layer, different transfer functions like tan h, linear tanh, sigmoid, linear sigmoid in output layer, different supervised learning rules like momentum ,step, conjugant gradient and quick propagation are investigated in simulation. The step size and momentum are gradually varied from 0.1 to 1 for static back propagation rule. After meticulous examination of the performance measures like MSE, NMSE, Correlation Coefficient and the regression ability and by processing elements graph. By closely observing processing element graph of Fig.3 the optimal parameters are decided and listed in table 1 for the duffing time series .

Table 1: Parameters of FTLRNN

Sr. no.	Parameter	Hidden Layer	Output Layer
1	Processing elements	21	1
2	Transfer function	tanh	Lin tanh
3	Learning rule	Momentum	momentum
4	Step Size	1	0.1
5	Momentum	0.8	0.8

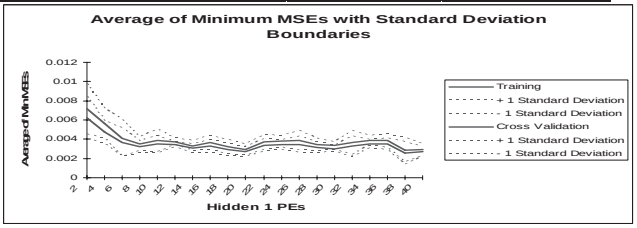

Fig. 3 Selection of processing elements

It is found that the performance of the selected model is optimal for 21 neurons in the hidden layer with regard to the MSE, NMSE, and correlation coefficient and the regression performance for the testing data sets. When we attempted to increase the number of hidden layer and the number of processing element in the hidden layer, the performance of the model is not to seen to improve significantly .On the contrary it takes too long time for training because of complexity of the model. As there is single input and single output for the given system, the number of input

and output Processing Elements is chosen as one. Now the NN Model (1:21:1) is trained three times with different weight initialization with 1000 iterations of the static back propagation algorithm with momentum term. After Comparing the performance measures like MSE, NMSE and r (regression) for 6000 samples as Training ,2500 samples as testing and 1500 as cross validation as mentioned in table 2 and 3. It is observed that the FTLRNN results out perform well as compared to MLPNN for all step (K=1,5,10,20,50,100) ahead prediction

Table 2: Performance of MLPNN on testing data

K (step)	MSE	NMSE	r
1	0.08750	0.5666	0.66035
5	0.07634	0.55568	0.68434
10	0.08331	0.54275	0.67991
20	0.08580	0.62975	0.63454
50	0.10079	0.74545	0.54593
100	0.11397	0.85011	0.40191

Table3: Performance of FTLRNN on testing data

K	MSE	NMSE	r
1	0.00113	0.00957	0.99545
5	0.00202	0.01315	0.99356
10	0.00335	0.02185	0.98910
20	0.00632	0.04150	0.97982
50	0.00927	0.06562	0.96663
100	0.00117	0.07870	0.96000

Then the proposed FTLRNN network model with gamma memory is trained for the best combinations resulted for training and testing exemplars. Then for the values on which the performance measures are optimum for the combination of training and testing samples ,it is experimented for 1000 to 40000 iterations for each multi step (k=1,5,10,20,50,100) ahead prediction of chaotic duffing time series . It is observed that up to the 20 step ahead prediction the result of the forecasting process is practically perfect. From the figure 4 and figure 5 i.e for the long step i e for 100 step and 50- step ahead prediction the network output is slightly deviating from the desired output but the results are better. Then in the steps of 10000 it is varied up to 40000 for out the optimum values of performance measures like MSE, NMSE and regression as shown in table 5 to 8 for 5,10.20.50 and 100-step ahead prediction. It is observed that for higher values of epochs ,the output of the FTLRNN network closely follows the actual output not only for short step (1,5,10) ahead prediction but also for long Step (20,50,100) ahead prediction.

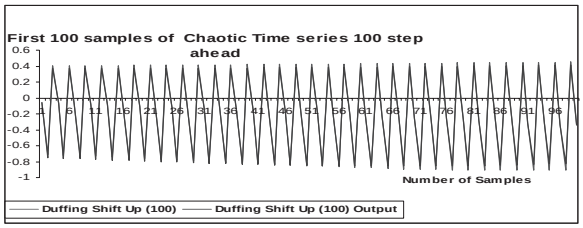

Fig. 4 Desired output & Network Output for Testing data.100 step

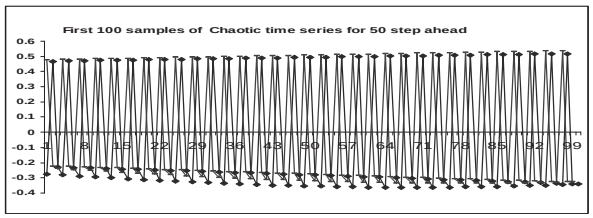

Fig. 5 Desired output & Network Output for Testing data. 50 step

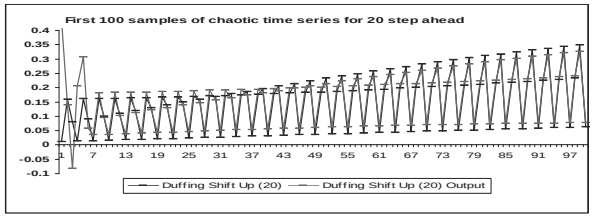

Fig. 6 Desired output & Network Output for testing data 20 step

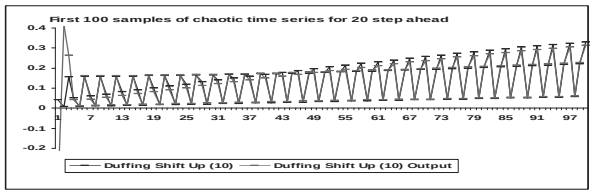

Fig. 7 Desired output & Network Output for Testing data 10 step

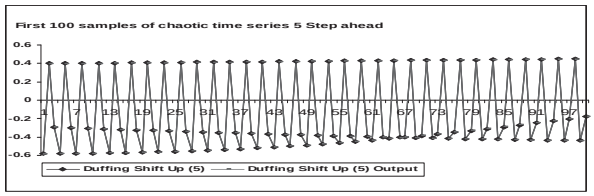

Fig. 8 Desired output & Network Output for Testing data 5 step

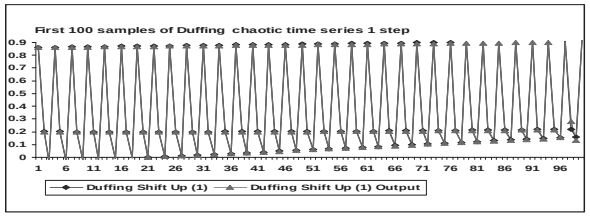

Fig. 9 Desired output & Network Output for Testing data for 1-SAP step

Table 4: For K=5, Epochs Variation above 10000

No.of Epoch	MSE (Test)	r(Test)	NMSE Test	Elapsed time min.
10000	0.00126	0.99603	0.0080	12.28
20000	0.00124	0.99606	0.0078	24.47
30000	0.00125	0.99609	0.00796	34.30
40000	0.00118	0.99624	0.00759	40.20

Table 5: For K=10, Epochs Variation above 10000

No. of Epochs	MSE (Test)	r (Test)	NMSE (Test)	Elapsed time min.
10000	0.00234	0.99265	0.01468	13.36

20000	0.00235	0.99261	0.01478	24.2
30000	0.00233	0.99267	0.01461	25.21
40000	0.00231	0.99276	0.1449	34.06

Table 6 :For K=20 Epochs Variation above 10000

No. of Epochs	MSE (Test)	r(Test)	NMSE Test	Elapsed time min.
10000	0.00427	0.98645	0.0272	16.56
20000	0.00217	0.99179	0.01643	29.44
30000	0.00424	0.98654	0.02708	36.53
40000	0.00437	0.98609	0.02791	50.38

Table 7 For K=50 ,Epochs Variation above 10000

No. of Epochs	MSE (Test)	r(Test)	NMSE Test	Elapsed time min.
10000	0.00869	0.96676	0.06767	14.2
20000	0.00797	0.96973	0.06208	18.3
30000	0.00844	0.96744	0.06573	25.2
40000	0.00893	0.96509	0.06949	30.3

Table 8: For K=100 ,Epochs Variation above 10000

No. of Epochs	MSE (Test)	r(Test)	NMSE Test	Elapsed time min.
10000	0.00627	0.97987	0.04006	15.49
20000	0.00619	0.98032	0.03959	20.33
30000	0.00613	0.98022	0.03919	26.47
40000	0.00589	0.98100	0.03767	31.54

Sun spot time series

In case of the sunspot time series , the exhaustive experimentation is carried out for 6 months ahead prediction the performance measures are compared for the Multi layer perceptron (MLP), Feed forward network(FFN), Jorden Elman, Focussed time lagged neural network model (FTLRNN), radial basis function(RBF) , fully recurrent and partially recurrent neural network models as shown in the graphs fig. 10 for M.S.E., Fig. 11 for regression r . Also the optimal parameter resulted for fully recurrent neural network with tanh axon model is listed in table 9.

Table 9 Parameters of fully recurrent network.

Sr. no.	Parameter	Hidden Layer	Output Layer
1	Processing elements	15	1
2	Transfer function	tanh	tanh
3	Learning rule	Momentum	momentum
4	Step Size	1	0.1
5	Momentum	0.8	0.8

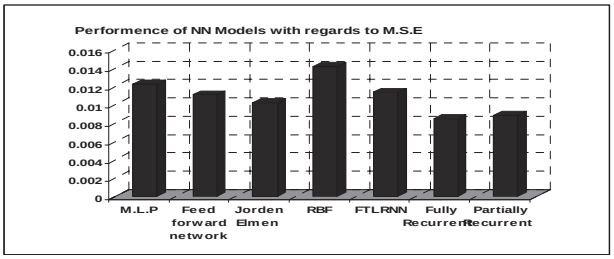

Fig. 10 Performance of various networks with regards to M.S.E for 6 month ahead

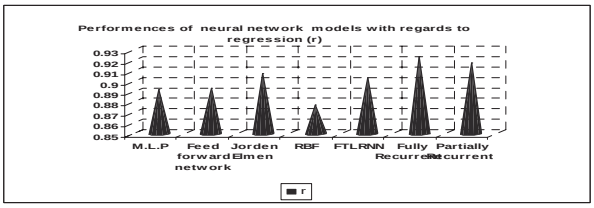

Fig. 11 Performance of various networks with regards to r for 6 month ahead

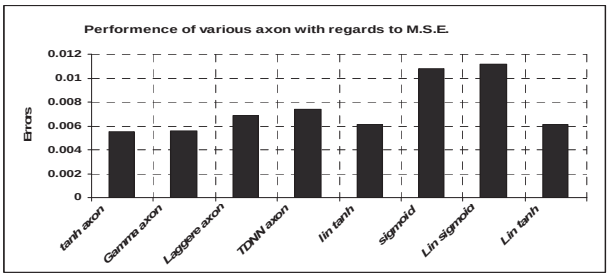

Fig. 12 Performance of various Axons with regards to M.S.E for 6 month ahead prediction

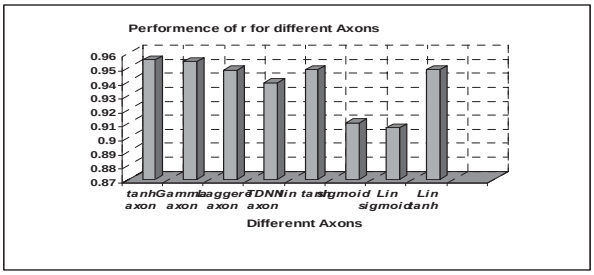

Fig. 13 Performance of various Axons with regards to r for 6 month ahead prediction

Then the proposed Fully recurrent neural network model with tanh axon is trained for the best combinations resulted for different tan h, Gamma axon ,Laggere axon , TDNN axon, sigmoid and linear sigmoid axon of recurrent model and the various performance measures are compared as shown in graphs Fig. 12 for M.S.E. and Fig.13 for r for 6 -month ahead prediction. From the plots it is resulted that for tan h axon the performance measures value are significant as compared to other axons. Then for this configuration of the model for tanh axon different combinations of training and testing samples is varied

keeping cross validation constant to find out the optimal value of performance measures for the combination of training & testing samples. The results are as shown in a graph of Fig. 14 and Fig. 15 with regards to M.S.E. and regression r for 1, 6,12,18,24 month ahead prediction. For 1 month and 6 month ahead the results are optimum for training samples 60% and testing 25%, for 12 month ahead the results are optimum for training samples 70% and testing 15%, for 18 and 24 month ahead the results are optimum for training samples 80% and testing 5%. Then for these combinations the number of Epochs is varied from 2000 to 20000 to get for which number of epochs the performance values are optimum. The results are plotted as shown in figure 17 with regards to r for all months, fig. 18 for 1 and 6 month ahead fig. 19 for 12,18, and 24 month ahead with regards to N.M.S.E , fig 27 for 1 and 6 month ahead and fig 26 for 12, 18 & 24 month ahead with regards to M.S.E. Also the graph between the desired output and actual output are plotted for First 300 samples on testing data set as shown in figure 19 for 1 month ahead, figure 20 for 6 month ahead, figure 21 for 12- month ahead, and figure 22 for 24 month ahead. It is observed that the output of proposed network closely follows the actual output,

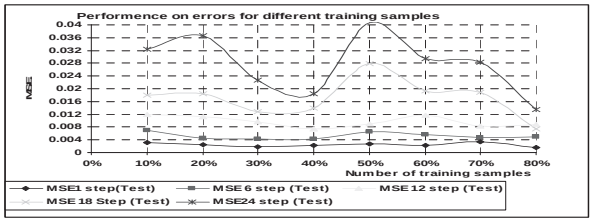

Fig. 14 Performance of M.S.E with different training samples for 1,6,12,18 &24 month ahead

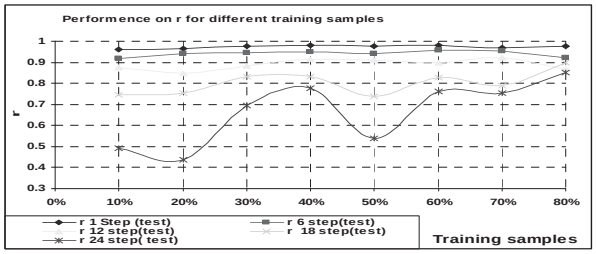

Fig. 15 Performance of r with different training samples for 1,6,12,18 & 24 month ahead prediction

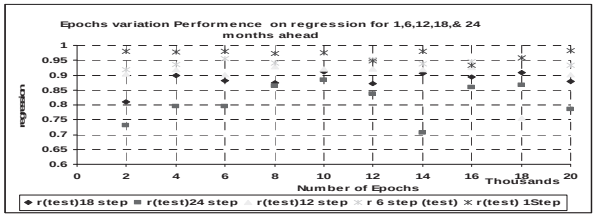

Fig. 16 Epochs variation performance on r for all months ahead prediction

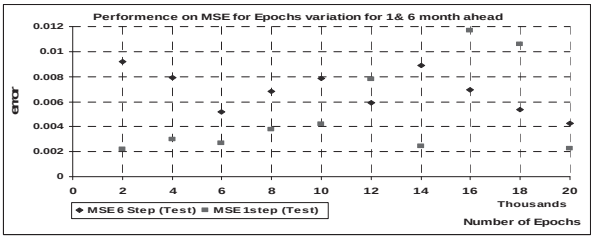

Fig. 17 Epochs variation performance on M.S.E for 1& 6- month ahead prediction

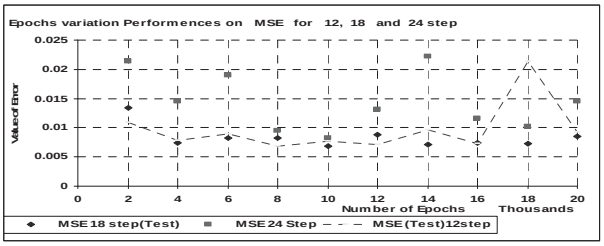

Fig.18 Epochs variation performance on M.S.E. for 12, 18 & 24 month ahead prediction

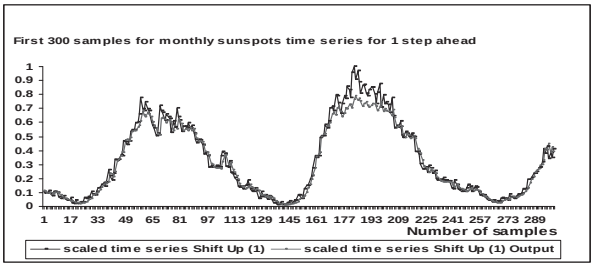

Fig. 19 First 300 samples of Desired output Vs Actual output for 1- month ahead for optimized network

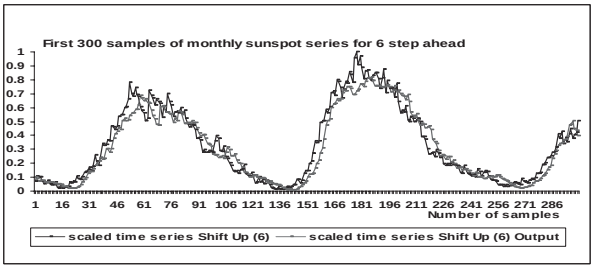

Fig. 20 First 300 samples of Desired output Vs Actual output for 6- month ahead for optimized network

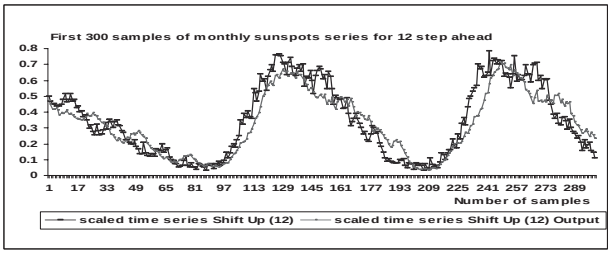

Fig. 21 First 300 samples of Desired output Vs Actual output for 12- month ahead for optimized network

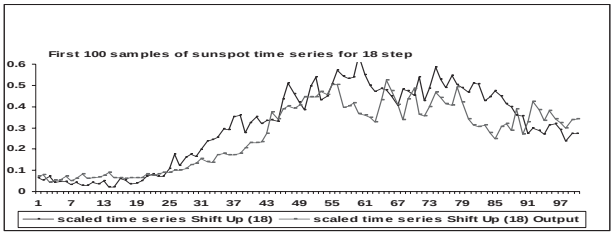

Fig. 22 First 300 samples of Desired output Vs Actual output for 18-month ahead for optimized network

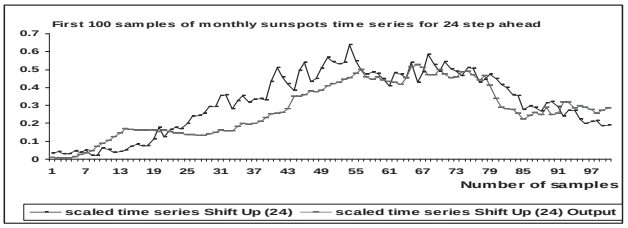

Fig.23 First 300 samples of Desired output Vs Actual output for 24- month ahead for optimized network

6 CONCLUSION

It is seen that focused time lagged recurrent network with gamma memory is able to predict the chaotic duffing time series so for as long distant predictions are quite well in comparison with the Multilayer preceptron (MLP) . Static NN configuration such as MLP NN based model is failed to cope up with the underlying nonlinear dynamics. It is seen that MSE, NMSE of the proposed dynamic model for testing data set as well as for training data set are significant than those of static MLP NN model . In addition it is also observed that the correlation coefficient of this model for testing and training exemplars are much higher and closed to unity for the multi step (k=1,5,10,20,50,100) ahead prediction. For the static MLP structure for 50 step ahead prediction the value of regression is 0.54593 and MSE is 0.10079 for the proposed model it is 0.96663 and MSE it is 0.00927. Similarly for static MLP for 100- step ahead the value of regression is 0.40191 and MSE it is 0.11397 and for the proposed FTLRNN the value of regression is 0.96000 and MSE it is 0.00117. Next for which training and testing samples the performance measures are optimum for that the number of epochs are varied from 2000 onwards to achieve more optimum results as shown in the figure 10 to 14. Then above 20000 epochs are varied in the steps of 10000as shown in the tables 10 to 14 for all steps. It is resulted from the experiments that for varying the higher values of iteration (i.e 10000 to 40000) the proposed model performs significantly better and the performance measures values like MSE, NMSE are reduced and the Correlation coefficient (r) values are approaching closed to the unity for all the steps as mentioned in the tables from table10 to 15. For the 100 step ahead prediction the value correlation coefficient is 0.96000 for 1000 iterations and when the number of iteration is increased up to the value 40000 then correlation coefficient is increases to 0.98100 which is closed to the unity. Hence the focused time lagged recurrent neural network with gamma memory filter has out performed the static MLP based neural network significantly for short step K=1,5,10 and for long step K=20,50,100 ahead prediction as well as it is found that for increasing value of the number of found epochs steps (k=1,5,10,20,50,100) the performance measures for any optimum. and the proposed network output closely follows the actual output for all the steps.

For the monthly sun spot time series ,It is observed that fully recurrent neural network model with tanh axon learns the dynamics of the system for 1, 6 ,12, 18 and 24- month ahead prediction quite well as compared with the Static MLP, Feed forward , FTLRNN, Jorden Elman and partially neural network model. It is seen that MSE, NMSE of the proposed dynamic model for testing data set as well as for training data set are significant. . In addition it is also observed that the correlation coefficient of this model for testing and training exemplars are much higher and closed to unity for the multi step (k=1,6,12,18,24) ahead prediction. Then for which combination training and testing samples the results are optimum for that combination the proposed model is trained for 2000 to 20000 epochs for all the month ahead prediction and the results are plotted with regards to regression r. It is observed that for 12 month ahead prediction for 2000 epochs the value of M.S.E is 0.01090 and correlation coefficient r is 0.90350 and it is improved for M.S.E to 0.00685 and correlation coefficient r is 0.92798 for 8000 epochs , for 18 month ahead prediction for 2000 epochs the value of M.S.E is 0.01355 and correlation coefficient is 0.81103 and it is improved for M.S.E to 0.00684 and correlation coefficient is 0.91093 for 10000 epochs and for 24-months ahead prediction for 2000 epochs the value of M.S.E is 0.02133 and correlation coefficient r is 0.73131 and it is improved for M.S.E to 0.00824 and correlation coefficient to 0.88460 for 10000 epochs which is closed to unity. Also the graph between the desired output and actual output are plotted for1, 6, 12, 18 and 24-month ahead prediction. It is visually inspected that that the output of proposed neural network closely follows the actual output,

7 REFERENCES

[1] A.S. Weigend and N.A. Gershenfeld, “Time series prediction : Forecasting the future and Understanding the past” . Reading , M.A.:Adision – Wesley1994.

[2] B.Townshend , “ Nonlinear prediction of speech signals, in nonlinear Modelling and fore casting , M. Casdagli and S. Eubank ”, Eds . Reading MA :Addison –wesley ,1992,pp 433-453.

[3] H. Leung and T Lo , “ Chaotic radar signal processing over the sea” IEEE J. Oceanic Eng ., Vol 18 pp287-295,1993.

[4] M. Casdagli, “Nonlinear prediction of chaotic time series ” ,Phys. D , Vol 35,pp.335-356,1989.

[5] J.B. Elsener, “ predicting time series using a neural network as a method of distiuinguishing chaos fro noise” , J phys, A:Math .Gen vol. 25 ,pp 843 -850, 1992 .

[6] X. He and A. Lapedes , “Nonlinear modeling and prediction by successive approximation using radial basis function” . Phy . D , Vol. 70 ,pp289-301,1993.

[7] H. Leung , “ Applying chaos to radar detection in an ocean environment An experimental study , IEEE.J. Oceanic Engg ., Vol 20 pp 56”-64,1995.

[8] K.M. Short, “Steps towards unmasking secure communications” , International Journal of Bifurcation chaos , Vol 4 ,pp.959-977,1994.

[9] P.G. Cooper, M.N. Hayes ,and J.E.Whalen , “ Neural networks for propagation modeling” , Electromagnetic Environmental Test facility , Fort Huachuca ,AZ ,Atlantic Res .corp. Rep .92-12-002,1992.

[10] Sanjay V Dudul Identification of a Liquid of a Liquid Saturated Steam Heat Exchanger Using Focused Time Lagged Recurrent Neural Network Model ,IETE Journal of Research Vol53 No1,January-February2007,pp69-83.

[11] Barron, A R, Universal approximation bounds for superposition's of a sigmoid function, IEEE Transactions on information ***Theory,39,pp930-945,1993***.

[12] Juditsky , A Hjalmarson, H, Benveniste, A, Delyon , B,Ljung , L, Sjoberg, J & Zang, Q,Nonlinear black-box models in system identification: Mathematical foundation, *Automatica,vol 31 , no 12,pp 1725,1995*.

[13] Waibel ,A, T Hanazana, G Hinton K Shikano & K J Lang, phoneme recognition using time delay neural Transactions on Acoustics , speech , and signal processing *vol ASSP-37,pp328,1989.*

[14] Jose C Principe , Neil R Euliano & W Curt Lefebvre, Neural and Adaptive Systems –Fundamentals Through Simulation, *John-Wiley & Sons,Inc,2000*.

[15] Sanjay V Dudul Identification of a Liquied of a Liquid Saturated Steam Heat Exchanger Using Focused Time Lagged Recurrent Neural Network Model ,IETE Journal of Reasearch Vol53 No1,January-February2007,pp69-83.

[16] Varies B & J C Principe, The gamma model-A new neural model for temporal processing, Neural Network , *vol5,pp565,1992.*

[17] Press, NY, K.Funahashi : On the Approxima tion Realization of Continuous Mapping by Neural Networks, Neural *Networks.vol.2, No.3, 1989*

[18] K.Doya, S.Yoshizawa : Memorizing oscillatory patterns in the analog neural network, proceedings of the *3rd IJCNN I, 27-32, 1989*

[19] M.Casdagli :- Non-linear predictions in chaotic time series physica D,35 335_359,1989.

[20] Narendra K S & Parathasarathy K,Gradient Methods for Optmization ,IEEE transactions on Neural Networks , vol 2 no2, pp 252-62,1991.

[21] DeboeckG., Kohonen T.,Visual Explorationz in *Finance with self organising* Maps,Springer,1998.

[22] Guimaraes G., Moura Pires F.,An *Essay in Classifying Self – organising Maps for Temproal sequence processing* ,In Allison .,N.H.,Allison L., Slack J,(eds), Advances in self organizing Maps ,pp259-266, Springer,2001.

[23] Kohonen Teuvo, *Self organizing Maps* , New York :Springer-Verlag,2001.

[24] Sanjay V. Dudul Prediction of Lorenz chaotic attractor using two layer Perceptron , Journal of Applied soft computing ,Elsevier Journal pp 2005.

[25] Mini Qi and G Peter Zhang. " Trend Time Series Modeling and Forecasting with Neural Networks" ,IEEE Transactions on Neural Networks , Vol. 19 NO 5 ,May 2008.