Chaotic Characterization of Electric Load Demand Time Series & load forecasting by using GA trained Artificial Neural Network

Hitesh Kumar
Department of Electrical Engineering
DCRUST University of science & technology
Sonipat, India
hitesh00121@gmail.com

Sanju Saini
Department of Electrical Engineering
DCRUST University of science & technology
Sonipat, India
Sanjusaini.ee@dcrustm.org

Abstract—Non linear time series modeling and forecasting has fundamental importance to various practical domains and a lot of active research work is going on in this area during past several years. In this work, an artificial neural network based model is used for load forecasting. Further, its performance is improved by using a canonical genetic algorithm. The method is supported by giving the forecasting result via simulation for real non linear time series of the electric load demand of Delhi region. To evaluate forecasting accuracy as well as to compare different models, three performance measures, viz. RMSE (Root mean square Error), MAPE (Mean Absolute Percentage Error) and REP (Relative Error Percentage) have been used. In this paper, all the simulations are carried out in MATLAB 7.10.0 environment using core i5 Intel processor.

Keywords—Load forecasting; chaotic characteristics; ANN (Artificial Neural Network); GA (Genetic Algorithm).

I. INTRODUCTION

Electric power load forecasting has a major role to play in the development and efficient operation of power system by removing uncertainty in taking decisions with respect to operation of the existing resources and electrical capacity enhancement of the power system in future [20]. Accurate forecast of load demand helps in economic optimization and increases the reliability and security of electric power system [20]. However, there are a number of challenges to the forecasting of electrical power demand, e.g., seasonal variations, industrial growth, change in load demand patterns during summer and winter session, etc. [16] [1]. Main techniques for electric load forecasting are short-term (lead time span from a few minutes, hour ahead to a few days), medium-term (forecasting span ranging from one week ahead to a few days) and long-term forecasting (lead time ranging from one year to twenty year) [7].

Short-term power demand forecasting mainly helps in voltage regulation and reduction of power losses, hence is absolutely essential for proper scheduling and working of a power system. Traditional method for short-term forecasting include linear regression [11], time series models, e.g., ARMA, ARMAX, ARIMA[18][3], Kalman filter[12], etc. These methods are based on statistical techniques involving a large amount of computational time [10]. Hence, modern forecasting techniques using Artificial Neural Network (ANN) [13], neuro fuzzy techniques [21], expert system etc. have been developed. With due consideration of uncertainty in system dynamics, these techniques produce better forecasting results [20].

Recently, chaos theory is being used for short-term forecasting of a number of complex nonlinear time series [5][2]. The main characteristics of nonlinear chaotic system are aperiodicity, sensitivity to initial conditions and lack of long-term predictability. Electric load series is nonlinear in nature and short-term prediction of a time series can be done with high precision if the irregular movement characterized by the time series can be attributed as a chaotic phenomenon [8][22]. Chaos is deterministic and a chaotic time series is short-term predictable in contrast to a random time series [2].

In this paper, the chaotic nature of electric power demand time series of Delhi region has been investigated with respect to characteristics like nonlinearity, fast fourier transform and power spectral density. Method of delay coordinates has been used for phase space reconstruction of the time series after choosing the optional embedding dimension and time delay. The chaotic nature of the load demand series (hence, short time predictability) is further investigated by calculating correlation dimension and the largest lyapunov exponent of the reconstructed time series (it should be a finite positive number for chaotic system).

Finally, an Artificial Neural Network with only one hidden layer is used for short-time prediction of the load demand time series. As the conventional backpropagation learning of
artificial neural network (ANN) has the disadvantage of being trapped into local minima, so genetic algorithms have been utilized for ANN training to improve the overall performance of forecasting. The paper is organized as follows: in section II, the chaotic characteristics of a time series are explained. Forecasting of load time series using two layers ANN is given in section III. Use of GA to improve the overall performance is given in section IV. Simulation results appear in section V. Finally, the paper ends with conclusion and future directions.

II. CHAOTIC CHARACTERISTICS OF LOAD DEMAND TIME SERIES

A. Nonlinearity

Nonlinearity is the first condition for a time series to be chaotic [2].

B. Power Spectrum and Power Spectral Density

Power spectral density (PSD) of a time series gives a measure of power per unit of frequency. Mathematically, it is Fourier transform of the autocorrelation sequence of the time series. PSD is also defined as squared modulus of Fourier transform of the time series, scaled by a constant [20]. Power spectrum gives information of periodic components present in a signal. For a periodic signal, the power spectrum contains discrete lines and the power is spread over a continuous range of frequencies for a stochastic signal [6]. Chaotic signals have broadband spectrum.

C. Lyapunov Exponent

Average exponential separation between initially nearby trajectories is quantified by Lyapunov exponents. Largest Lyapunov exponent should be a finite positive number for a chaotic process and infinite for a stochastic process [14]. To calculate Lyapunov exponent from an observed time series, the method of delay coordinate embedding is used for reconstructing the phase space. For this purpose, the suitable values of delay time and embedding dimension are chosen as follows:

1. Optimal Delay Time (τ)

Optimal time delay can be chosen as the first local minimum of auto-mutual information function [4] because at this time, maximum information is added by \( x_i \), to the knowledge one has from \( x_j \) [14].

2. Determination of Embedding Dimension (m)

Cao’s method [23] may be used to determine the optimal embedding dimension. For a time series of the form \( \{x_1, x_2, \ldots, x_n\} \), \( i^{th} \) reconstructed vector can be represented as Equation (1).

\[
Y_i(m) = (x_i, x_{i+\tau}, \ldots, x_{i+(m-1)\tau}) \quad \text{for } i = 1, 2, \ldots, N \tag{1}
\]

where \( N = n - (m - 1)\tau \), ‘m’ is the embedding dimension and ‘\( \tau \)’ is embedding delay.

Let \( Y_i^{NN}(m) \) is the nearest neighbor of \( Y_i(m) \), then \( A_2(i,m) \) is defined by Equation (2).

\[
A_2(i,m) = \left[ \frac{||Y_i^{m+1} - Y_i^{NN}(m+1)||}{||Y_i^m - Y_i^{NN}(m)||} \right] \tag{2}
\]

Here \( \| \| \) represents euclidean distance. \( E(m) \), i.e., the mean value of \( A_2(i,m) \) is calculated by using equation (3).

\[
E(m) = \frac{1}{n-m+1} \sum_{i=1}^{n-m} A_2(i,m) \tag{3}
\]

To determine the variation of \( E(m) \) with respect to \( m \), \( E_r(m) \) is defined by Equation (4).

\[
E_r(m) = \frac{E(m+1)}{E(m)} \tag{4}
\]

\( E_r(m) \) stops changing when \( m > m_0 \) (i.e., the minimum embedding dimension of the time series). Thus \( m_0 \) is the optimal embedding dimension.

After reconstructing the time series using optimal delay time and optimal embedding dimension, a direct method suggested by [15] may be used to determine the largest Lyapunov exponent. It involves the determination of the exponential divergence of nearby trajectories on Logarithmic scale via prediction error, \( P(k) \) (defined by Equation (5)) [19].

\[
P(k) = \frac{1}{NT} \sum_{n=1}^{N} \log_2 \left( \frac{||Y^{n+k} - Y^{nn+k}||}{||Y^n - Y^{nn}||} \right) \tag{5}
\]

where ‘\( k \)’ represents the number of time steps, ‘\( Y^{nn} \)’ being the nearest neighbor of \( Y^n \). ‘\( T \)’ is the sampling time and \( N \) is as defined in Equation (1). Slope of the curve of prediction error plotted against ‘\( k \)’ (in the intermediate values of \( k \) ) gives the largest Lyapunov exponent [15][19].

D. Correlation Dimension (d)

Correlation dimension quantifies the effect of presence of a data point on the position of the other points lying on the same attractor. Fractal value of the correlation dimension characterizes the chaos. Mathematically, it is given by the saturation value of correlation exponent, i.e., \( D_2 \) (slope of log \( C(r) \) versus log(\( r \)) plot) [14]. For an \( m \)-dimensional phase space, \( C(r) \) is defined by Equation (6).

\[
C(r) = \lim_{n \to \infty} \frac{2}{N(N-1)} \sum_{i,j=1}^{N} H(r - |Y^i - Y^j|) \tag{6}
\]

‘\( N \)’ is the number of points and ‘\( r \)’ is the radius of sphere centred on \( Y^i \) or \( Y^j \).

\( H \) is the Heaviside step function [14] defined by Equation (7).

\[
H(u) = \begin{cases} 
1 & \text{for } u \geq 0 \\
0 & \text{for } u < 0 
\end{cases} \tag{7}
\]

where \( u = r - |Y^i - Y^j| \)
III. PREDICTION OF CHAOTIC TIME SERIES USING ANN

For a time series of the form \( \{x_1, x_2, \ldots, x_n\} \), the predicted value at \((t+\Delta t)\)th instant is a function of the values of the time series data at \(t\), \((t-\tau)\), \(t-(m-1)\tau\) instants as represented by equation (8).

\[
x(t+\Delta t) = f(x(t), x(t-\tau), \ldots, x(t-(m-1)\tau))
\]

Here, \(m\) and \(\tau\) are the optimal values of embedding dimension and time delay respectively. The time series is divided into two parts. First part is used to provide training data and rest of the data in time series is used as test sample. Using equation (8), the training data (i.e., input & output matrices) for ANN is generated, as shown in Equation (9).

\[
\text{Input} = \begin{bmatrix}
x_1 & x_{1+\tau} & \ldots & x_{1+(m-1)\tau} \\
x_2 & x_{2+\tau} & \ldots & x_{2+(m-1)\tau} \\
\vdots & \vdots & \ddots & \vdots \\
x_N & x_{N+\tau} & \ldots & x_{N+(m-1)\tau}
\end{bmatrix}
\]

\[
\text{Output} = \begin{bmatrix}
x_{1+(m-1)\tau+\Delta t} \\
x_{2+(m-1)\tau+\Delta t} \\
\vdots \\
x_{N+(m-1)\tau+\Delta t}
\end{bmatrix}
\]

An ANN consisting of six input nodes and two layers, i.e., one hidden layer (with five neurons) and one output layer (with one neuron) has been used for prediction of load demand (one hour ahead). Each neuron in the hidden layer has a tan-sigmoid type activation function define by eqn. (10) and neuron in the output layer has a linear activation function. Network is trained by using back-propagation training algorithm.

\[
tansig(x) = \left(\frac{2}{1+\exp(-2x)}\right) - 1
\]

IV. USE OF GA TO TRAIN ANN

An artificial neural network trained by using backpropagation algorithm may be stuck in local minima. To deal with this limitation, and to improve the prediction accuracy, the ANN is trained by using an optimization technique, i.e., genetic algorithm.

In genetic algorithm, number of iterations, parameters, population size and initial weights play an important role. Here the task of genetic algorithm is to evolve the values of 36 parameters with a goal to minimize the objective function i.e. MAPE. The population size is chosen to be equal to 450 (ten times the number of total variables to be optimized) which is a reasonably large number to maintain a fair amount of randomness in the algorithm; of GA. The population is evaluated at each generation and at the time of each successive epoch; a proportion of the existing population is selected after completion of a fitness-based process and is combined with some randomness to breed a new generation. Roulette Wheel selection is used as the selection operator in which each of the individual is assigned a slice of ‘Roulette wheel’, sized proportionate to individual’s fitness. The wheel is spun ‘population size’ times. Further, for each new solution to be produced, a pair of ‘parent’ solutions is selected and are allowed to undergo crossover (typical value of crossover fraction = 0.8). Two point crossover and uniform mutation operators have been used. Mutation operator helps the genetic algorithm in searching a broader search space by providing genetic diversity (by emulating little alteration in genetic properties during reproduction in natural species). In uniform mutation, the selected gene (typical value of the probability of mutation of each gene = 0.01) is replaced by a uniform random number selected within specified bounds of that gene. Elitism is added to the algorithm by surviving the two best fit individuals (elite children) in the current generation to the next generation. As a result, the best value of objective function can only decrease or stay the same from one generation to the next generation.

The average change of objective function value over the immediately previous five generations within a tolerance band of ±0.000001 is used as a stopping criterion.
V. RESULTS AND DISCUSSION

As a case study, 1272 samples of data representing daily hourly load demand of Delhi for 53 days (from 01/02/2016 to 24/03/2016) have been downloaded from the official site of State Load Dispatch Center (SLDC), Delhi (www.delhisldc.org/Loaddata.aspx). The data has been tested for the presence of chaos and hence, short-time predictability. The prediction of the load demand has been done with the help of an ANN and principles of phase space reconstruction.

A. Chaos in Power Load Demand Time Series

1. Nonlinearity

   The nonlinearity of the time series of power load demand can be observed from the plot of Fig. 1.

2. Power Spectrum and Power Spectral Density

   Broadband spectrum of Fig. 2. indicate the presence of chaos for the load demand time series.

3. Lyapunov Exponent

   The most common test for the presence of chaos in a system is the presence of positive Lyapunov exponent. To determine largest Lyapunov exponent, the optimal time delay ($\tau$) and embedding dimension are determined by using auto mutual information and Cao’s method respectively. Auto mutual function of load time series is calculated by using Open TSTOOL [9] software package. First the minimum of this function occurs for embedding delay = 5 shown in Fig. 3. This is the value of optimum delay used for reconstruction of the time series.

Using Cao’s method, the optimum value of embedding dimension can be determined. It is the value beyond which $E_l(m)$ stops changing. Using open TSTOOL box [9], it has been found to be equal to five.

Using optimal values of time delay and embedding dimension, the value of the largest Lyapunov exponent is calculated by the slope of prediction error curve as shown in Fig. 4, for intermediate values of time steps, $k$. Using this method, the largest Lyapunov constant for the reconstructed time series is 0.0297, a finite positive value, hence indicates the presence of chaos [24].

4. Correlation Dimension

   Correlation dimension (d) from the reconstructed time series (using optimal delay and embedding dimension) is calculated using Open TSTOOL [9] software package. Using a maximum relative search radius of 0.1 (relative to the attractor size), the correlation dimension is ‘2.165’. The chaos is also characterized by fractal value of the correlation dimension.

B. forecasting of Load Time Series

1. Forecasting of Load Time Series Using ANN trained by using backpropagation algorithm

   First 822 entries from the load time series of the form $\{x_1, x_2, \ldots, x_{1272}\}$ are used to provide the training data and next 450 entries are used for testing the accuracy of forecasting for a
lead time of one hour. The accuracy of prediction is quantified in terms of various performance measures such as MAPE (Mean Absolute Percentage Error), REP (Relative Error Percentage) and RMSRE (Root Mean Square Relative Error).

Table I

<table>
<thead>
<tr>
<th>Training Function &amp; Description</th>
<th>Performance Measure</th>
<th>MAPE</th>
<th>REP</th>
<th>RMSRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian regulation backpropagation</td>
<td>Trainlm</td>
<td>2.5790</td>
<td>3.3610</td>
<td>3.4752</td>
</tr>
<tr>
<td>Levenberg-Marquardt backpropagation</td>
<td>Trainlm</td>
<td>2.973</td>
<td>3.870</td>
<td>3.963</td>
</tr>
<tr>
<td>Gradient descent backpropagation</td>
<td>Trainlm</td>
<td>3.191</td>
<td>3.863</td>
<td>4.152</td>
</tr>
</tbody>
</table>

In [24], this backpropagation training of ANN has been done with only one training function i.e., Levenberg-Marquardt. But, here in this work, in addition, two more ANN training functions, i.e., Bayesian regulation backpropagation & Gradient descent backpropagation have also been used. The performance of each training function is shown in TABLE I, which indicates that Bayesian regulation propagation results in best performance. To further improve the performance of ANN model, an optimization technique, i.e., genetic algorithm (GA) is used as discussed in next subsection.

2. Forecasting of Load Time Series by Using ANN evolved by using a genetic algorithm

Here genetic algorithm is used to evolve the complete structure of artificial neural network used for forecasting, it decide value of all the weight & biases of ANN having five neurons in hidden layer & one neuron in output layer. There are 36 parameters (b1, b2, ..., b36) to be optimized; which include 25 synaptic weights (b1, b2, ..., b25) associated with five neurons of the hidden layer & 15 synaptic weight (b26, b27, ..., b36) associated with neurons of output layer & six bias (b1, b2, ..., b6) for neurons of hidden layer & output layer. Using GA parameters as discussed in section IV, genetic algorithm is used to evolve ANN model. In 10 iterations the objective function is minimized as shown in Fig. 5.

Table II

<table>
<thead>
<tr>
<th>SR.NO</th>
<th>FINAL WEIGHT</th>
<th>SR.NO</th>
<th>FINAL WEIGHT</th>
<th>SR.NO</th>
<th>FINAL WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.296700</td>
<td>13</td>
<td>-0.371300</td>
<td>25</td>
<td>-0.981300</td>
</tr>
<tr>
<td>2</td>
<td>0.007500</td>
<td>14</td>
<td>-0.663200</td>
<td>26</td>
<td>-0.386400</td>
</tr>
<tr>
<td>3</td>
<td>-0.560300</td>
<td>15</td>
<td>1.315600</td>
<td>27</td>
<td>-0.618900</td>
</tr>
<tr>
<td>4</td>
<td>1.324100</td>
<td>16</td>
<td>-0.473100</td>
<td>28</td>
<td>-0.898100</td>
</tr>
<tr>
<td>5</td>
<td>-0.412700</td>
<td>17</td>
<td>0.148500</td>
<td>29</td>
<td>-0.907900</td>
</tr>
<tr>
<td>6</td>
<td>1.522900</td>
<td>18</td>
<td>-0.074400</td>
<td>30</td>
<td>-0.569900</td>
</tr>
<tr>
<td>7</td>
<td>-0.490900</td>
<td>19</td>
<td>-1.128600</td>
<td>31</td>
<td>1.040700</td>
</tr>
<tr>
<td>8</td>
<td>-0.269300</td>
<td>20</td>
<td>-0.906400</td>
<td>32</td>
<td>-0.797700</td>
</tr>
<tr>
<td>9</td>
<td>0.849300</td>
<td>21</td>
<td>1.430100</td>
<td>33</td>
<td>0.393900</td>
</tr>
<tr>
<td>10</td>
<td>-1.327400</td>
<td>22</td>
<td>-0.853100</td>
<td>34</td>
<td>-0.293400</td>
</tr>
<tr>
<td>11</td>
<td>-1.544300</td>
<td>23</td>
<td>0.750700</td>
<td>35</td>
<td>-0.127400</td>
</tr>
<tr>
<td>12</td>
<td>0.925500</td>
<td>24</td>
<td>0.028500</td>
<td>36</td>
<td>-0.116500</td>
</tr>
</tbody>
</table>

The final solution vector, ‘b’ is given in TABLE II. The performance of the GA evolved ANN model for MAPE, REP, & RMSE is shown in Table III. The simulation results for actual and predicted values are shown in Fig. 7; the performance of the said model is quite good as most of predicted values overlap the actual values of the load demand.

Table III

<table>
<thead>
<tr>
<th>Training Techniques</th>
<th>Performance Measure</th>
<th>MAPE</th>
<th>REP</th>
<th>RMSRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN+GA</td>
<td>2.2830</td>
<td>2.9315</td>
<td>3.1171</td>
<td></td>
</tr>
</tbody>
</table>

Use of Genetic algorithm improves result in 9.96% improvement in forecasting results backpropagation trained ANN. An overall comparison of various techniques used in this work for forecasting purpose is given in Table 4.

Fig. 5. Performance analysis of fitness value with genetic algorithms.

Fig. 7 Actual and Predicted Values of Load Demand.
VI. CONCLUSION & FUTURE SCOPE

In this paper, it is concluded that electric load demand time series is chaotic & hence short time predictable, backpropagation trained artificial neural network model has been successfully used for prediction of electric load demand of Delhi region. The performance of this forecasting model has been further improved by using GA to evolve the structure of ANN. In future, weather condition and weekdays may be taken into consideration to improve forecasting results.

Acknowledgment

The authors are thankful to the department of Electrical Engineering, DCR University of Science and Technology, Murthal, Sonepat, Haryana, India for providing the facilities to complete this work. They also thank the source of data analyzed in this paper, i.e., State Load Dispatch Centre (SLDC), Delhi.

| TABLE IV. COMPARISON OF BACKPROPAGATION TRAINED & GA TRAINED ANN |
|---------------|---------------|----------------|
| PERFORMANCE MEASURE | PERCENTAGE IMPROVEMENT |
| | ANN (X) | ANN+GA (Y) | ((X-Y)/X)*100 |
| MAPE | 2.5790 | 2.2830 | 11.47 |
| REP | 3.3610 | 2.9315 | 12.77 |
| RMSRE | 3.4752 | 3.1171 | 10.30 |

References