Electric Load Forecasts by Metaheuristic Based Back Propagation Approach

Papia Ray¹, Sabha Raj Arya² and Shobhit Nandkeolyar¹

¹*Department of Electrical Engineering, Veer Surendra Sai University of Technology, Burla, Odisha, India*

²*Department of Electrical Engineering, Sardar Vallabhbhai National Institute of Technology, Surat-395007, India*

E-mail: papia_ray@yahoo.co.in; {sabharaj79; nshobhit91}@gmail.com

Received 11 April 2017; Accepted 29 June 2017;
Publication 18 August 2017

Abstract

The prediction of system load demands a day ahead or a week ahead is called Short Term Load Forecasting. Artificial Neural Network based STLTF model has gained significance because of transparency in its modelling, simplicity of execution, and superiority of its performance. The neural model consists of weights whose optimal values are found out by means of different optimization techniques. In this paper, Artificial Neural Network trained by different methods like Back Propagation, Genetic Algorithm, Particle Swarm Optimization, Cuckoo Search model and Bat algorithm is utilized for load forecasting. A thorough analysis of the different techniques is carried out here in order to assess their extent and capability to yield result, by means of dissimilar models, in altered situations. The simulation results indicate that Bat Algorithm based Back Propagation model leads to least forecasting error in comparison to other techniques. However, Cuckoo Search method based Back Propagation model also gives less error relatively, which is very much permissible.

Keywords: Short Term Load Forecasting, Metaheuristic, Genetic Algorithm, Particle Swarm Optimization, Cuckoo Search, Bat Algorithm.

Journal of Green Engineering, Vol. 7, 61–82.

doi: 10.13052/jge1904-4720.7124

© 2017 River Publishers. All rights reserved.

List of Abbreviations

ANN	Artificial Neural Network
STLF	Short Term Load Forecasting
BP	Back Propagation
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
CS	Cuckoo Search model
BA	Bat algorithm

1 Introduction

Electric load demand forecasting has always been a main area of concern in any power utility industry. Usually models are developed based on the reminiscent of past local weather conditions and the knowledge of previous load demand data in order to perform the load forecasting. Such forecasts are generally aimed towards short-time period prediction for example one-day advance prediction, as the prediction for a longer interval, in case of mid-term or long-term load forecasting, will be less reliant due to propagation of error. The precision of predicting the load data has a huge implication over an electrical utility's operations and its cost of production. Accurate forecasting of load is therefore extremely vital, given the continuous variations happening within the power industry, as a result of deregulation.

STLF has always been a very important part of the Energy Management System as it falls within the primary exercise carried out for the entire routine operations' scheduling, may it be daily or weekly [1]. It is important to examine the distinctiveness of the load demand data and the factors that affect the load, in order to achieve a certain degree of exactness and quickness in the forecasting [2, 3]. Usually, STLF techniques have been divided into either outmoded techniques or techniques that are presently in practice. Various outmoded techniques for load forecasting have been able to achieve varying rates of success [4, 5]. Previous traditional methods for load demand forecasting like, regression method [6], time series method [7], pattern recognition [8], Kalman filters model [9], etc., were in regular use for a long period of time, thus corroborating preciseness that is scheme dependent. These traditional methods are multi-model methods and are pooled utility based, yielding satisfactory results for systems in which they are utilized [10]. However, they can't fittingly portray the complex nonlinear relations that exist among the load demand and the factors that influence it like period, weather

conditions or time of day, which are generally subjected to model variations. Present load forecasting methods such as Expert System [11], Artificial neural network based method [12–14], Fuzzy logic based method [15], and Hybrid Wavelets-Kalman filter [16] are some of the advanced techniques, which give unfettering outcomes.

Among them, ANN strategy is particularly engaging as it can manage the nonlinear dependencies that exist among the load demands and the factors influencing it, forthrightly. Neural networks have the capability to sort out non-linear curves suitably. ANN maps the input and output relations by the help of approximate linear or non-linear mathematical functions. In order to build a neural network for demand forecasting one has to pick from Back propagation, Hopfield and Boltzmann machine; feed forward or backward model, the connectivity among units and layers, and the number of arrangements to be applied to the model [17]. The most acknowledged ANN's training technique for load forecasting is BP. BP utilizes available input and output data and adjusts its weights with the help of some observable functions known as loss functions. This process is called as supervised learning. In case of neural network with unsupervised learning, pre-operational training is not required.

This paper is organized as follows: Section-2 illustrates the ANN model for STLF. Section-3 describes the advanced training methods, used in this paper, with improvements. Section-4 discusses the characteristics of power system load and the determination of input/output variables in the STLF process. Section-5 contains the simulation results. Section-6 concludes this entire work.

2 Artificial Neural Networks

ANN based techniques are projected as methods that doesn't demand explicit structures to speak about the sophisticated connections existing among the load demand data and the factors influencing it. Hence, they are a decent option to deal with the STLF issue. The current section will discuss the ANN model that is used for the STLF process.

2.1 ANN Model

ANNs consist of several extremely interrelated and basic essentials called as neurons. The ANN model is shown in Figure 1. Its exact equation is given by

$$O_j = f_j \sum_{i=1}^n (w_{ji} * x_i) \quad (1)$$

Figure 1 Mathematical model of ANN.

where, " O_j " = neuron output; " f_j " = transfer function, which typically utilizes a sigmoid function and are differentiable and non-decreasing in nature; " w_{ji} " is a weight that is changeable and signifies the joining strong point; " x_i " is the input to the neurons.

Feed Forward Neural Network (FFNN) is a type of ANN which usually comprises of three different layers. They are the input layer, hidden layer and the output layer. Signal, in this kind of network, propagates towards the forward direction in the following manner: first the signal travels from the input layer to the hidden layer and then from the hidden layer to the output layer. Errors are calculated in the respective nodes of output layer, by comparing the results obtained with the actual output. These calculated errors back propagate throughout the network and updates the real arithmetic weights. The choice of the input variables is based on the historical data available, such that they are consistent to the issues that affect the load. Here, output is the forecasted load which is the 24-hour load demand, for this situation. The successful execution of load forecasting is influenced by the choice of inputs, hidden nodes, scaling techniques, transfer function, and preparation. Hence, they must be selected sensibly.

2.2 Training

ANN undergoes training process in order to learn and map the input-output patterns. So, during the training process, the ANN's weights get updated until the mean square error (MSE) of the entire network falls below a certain threshold value, which is decided at the beginning of the whole

procedure. Usually, the ANN model is trained by BP learning algorithm. For updating the weight matrices, while the learning process is going on, in a multivariable optimization problem which involves numerical processes due to its non-linearity, the following equation is used

$$W^{t+1} = -\eta \frac{\partial E}{\partial w} + \alpha w^t \quad (2)$$

where, “ W^{t+1} ” is the next set of weights, “ w^t ” is the previous weight change, “ η ” is the rate of learning of the ANN model, “ α ” is the momentum factor, and “ w ” is the weight vector.

The methodology for load forecasting that utilizes an ANN model is illustrated in Figure 2. Union issues might arise on the immediate utilization of system information, as the variables have altogether dissimilar reaches. Two scaling plans chalked out and utilized. Firstly, all the input data variables “ X_i ” and output data variables “ Y_i ” are scaled to reside within the $[0, 1]$ region. The input and the output variables are scaled by the help of given expressions

$$\bar{X}_i^{(k)} = X_i^k / \max(X_i^{(k)}) \quad (3)$$

$$\bar{Y}_i^{(k)} = Y_i^{(k)} / \max(Y_i^{(k)}) \quad (4)$$

where, “ k ” is the index of input and output vector/pattern.

The biases as well as weights of each and every layer are assigned at the time when the neural system is designed. The associated system weights are updated till the best possible change, which connects the previous input and

Figure 2 ANN based demand forecasting procedure.

output cases, is found out. After simulation, the output of the neural network needs to be de-scaled in order to produce the required forecasted load output. As the properties of the load varies, error perception is critically aimed towards anticipating procedure. Hence, the Mean Absolute Percentage Error (MAPE) is calculated here as

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|X_t - X_f|}{X_t} * 100 \quad (5)$$

where, “ X_t ” is the real load and “ X_f ” is the forecasted load.

3 Metaheuristic Methods and Their Improved Algorithms

The various metaheuristic based BP algorithms used in this paper for the training of ANN network is discussed below.

3.1 Genetic Algorithm Based Back Propagation

GA is a global search technique based on stochastic approach which imitates the nature’s evolution process [18]. The entire procedure commences with initialization i.e., making arbitrary and reasonable guesses for the chromosomes. Based on the problem domain, the chromosomes are binary encoded, real encoded etc. The efficient exploitation of the solution space is ensured by two key controlling parameters which are the probability of crossover process and the probability of mutation process. This process yields new solution by continuously evolving and then the GA terminates when the stopping criteria is met.

The various aspects of this method are:

1. Coding

The parameters that symbolizes a possible answer to the problem i.e., genes, are concatenated to form a chromosome. In most traditional GA codes, the chromosomes are encoded into binary alphabets. A real coding scheme is adopted in this paper instead of binary encoding. An early population consisting of “ p ” chromosomes is produced arbitrarily, where “ p ” is the size of the population.

2. Weight Extraction

For regulating the fitness variable for respective chromosomes, elimination of their weights is performed. A chromosome is characterized by $x_1, x_2, \dots, x_d$,

$\dots, x_L$ and $x_{kd+1}, x_{kd+2}, \dots, x_{(k+1)d}$ denotes the k th gene ($k \geq 0$) within the chromosome. The real weight “ W_k ” is assumed from the equation

$$W_k = \left\{ -\frac{x_{kd+2} * 10^{d-2} + x_{kd+3} * 10^{d-3} + \dots + x_{(k+1)d}}{10^{d-2}} \right\}, \text{ if } 0 \leq x_{kd+1} \leq 5 \quad (6)$$

$$W_k = \left\{ \frac{x_{kd+2} * 10^{d-2} + x_{kd+3} * 10^{d-3} + \dots + x_{(k+1)d}}{10^{d-2}} \right\}, \text{ if } 5 \leq x_{kd+1} \leq 9 \quad (7)$$

3. Fitness Function

Fitness function is a measure of the quality of the solution and is problem dependent. In this paper, the fitness function is characterized as shown

$$fitness = \frac{1}{(1 + MAPE)} \quad (8)$$

where, “ $MAPE$ ” is the Mean Absolute Percentage Error.

Based on the above ideas, GA based BP model proposed in this paper follows the steps listed below:

Step-1: The length of the chromosome, the population size of the generation, and initial generation of parameter sets are initialized.

Step-2: Equation (8) is used to evaluate each individual’s fitness value.

Step-3: By the crossover and mutation process new individuals are generated and then the new generation’s fitness value is evaluated.

Step-4: Roulette wheel assortment scheme is used to combine the individuals, which helps in obtaining an individual having higher fitness value.

Step-5: Check if the condition for termination is achieved or not. If the condition is achieved, then go to Step-6; else repeat Step-3 and Step-4.

Step-6: From the above steps, we obtain the optimal individual. Hence the best initial guess for the weights of the ANN model are found out to perform the STLF using BP method.

3.2 Particle Swarm Optimization Based Back Propagation

The PSO technique was suggested by Eberhart and Kennedy in [19]. This method was evolved by carefully inspecting the social behaviour of flock of birds and school of fishes. Each individual’s behaviour in a swarm is dependent on its self-velocity as well as its neighbour’s velocity. As a result of this resultant velocity, the particle reaches an innovative position. For a

D-dimensional problem, $x_{i1}, L, x_{id}, L, x_{iD}$, denotes each particle in a PSO model having “ m ” particles. These particles represent a possible solution to the problem. The upgradation of the velocity and position of each individual particle in a swarm is done with the help of these equations

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_{1j}^t(p_{ij}^t - x_{ij}^t) + c_2r_{2j}^t(p_{gj}^t - x_{ij}^t) \quad (9)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (10)$$

where, “ w ” denotes the inertia weight factor; “ c_1 ” is the cognitive coefficient; “ c_2 ” is the social coefficient; “ r_{1j} ” and “ r_{2j} ” are two separate random quantities whose values lie between 0 and 1. “ c_1 ” and “ c_2 ” are the indicators of relative proportion of cognition and social interaction respectively.

For the j th dimension vector $P_i = (P_{i1}, K, P_{ij}, K, P_{gD})$ indicates to the position of the i th particle with the best fitness achieved so far i.e., “ $pbest$ ” and vector $P_g = (P_{g1}, K, P_{gj}, K, P_{gD})$ denotes swarm’s best position where the particle’s data is closest to the target, denoted by “ $gbest$ ”.

The PSO based BP model which is proposed in this paper follows the following steps:

Step-1: The neuron network as well as the architecture of the proposed ANN-BP model is defined, and before proceeding to the next steps some values are assigned to the following variables: the weight matrix “ w_0 ”, and the range of “ w_0 ”; rate of learning “ η ”; inertia weight factor “ w ”; particle size; the local optimal position of the particle “ $pbest$ ”; the global optimal position of the particle “ $gbest$ ”. The values of “ c_1 ”, “ c_2 ” as well as “ i ” is set as unity. The stopping criteria is also decided at the beginning.

Step-2: Define the fitness function according to the proposed method as

$$fitness = \frac{1}{(1 + MAPE)} \quad (11)$$

where, “ $MAPE$ ” is mean absolute percentage error, which is an indicator of each particle’s figure of merit in the swarm.

If the current value of fitness is better in comparison to “ $pbest$ ” then the current fitness value is assigned as the new “ $pbest$ ” otherwise the previous value of “ $pbest$ ” is retained.

Step-3: The extreme value of “ $pbest_i$ ” is selected as the present global best of the particle “ $gbest$ ”.

Step-4: Two arbitrary values are considered for “ r_1 ” and “ r_2 ” and Equations (9) and (10) are used to upgrade the velocity as well as position.

Step-5: Set the value of i as $i+1$.

Step-6: If the condition for maximum iteration is met or if the desired aim is achieved then the iteration is terminated and the particle whose location is denoted by the global position “*gbest*” is the optimal solution, otherwise go to Step-2.

3.3 Cuckoo Search Based Back Propagation

One of the most modern nature based metaheuristic procedure i.e., CS was developed in 2009 by X. S. Yang and S. Deb, CS algorithm is based on the parasitic behaviour of several species of cuckoo. In comparison to isotropic arbitrary walks, Lévy flights improves this process to a greater extent.

Lévy flight is generally characterized as an arbitrary walk where the step length has a probability distribution which is not exponentially bounded also called a heavy-tailed probability distribution. It has been suggested in many studies that the flight characteristic of insects and birds have the resemblance features of Lévy flight. A novel finding by Reynolds and Frye demonstrates that fruit flies or *Drosophila melanogaster* uses a sequence of straight flight paths interrupted by sudden right angular turns or bends thus leading to a Lévy flight kind of irregular-scale free search pattern, in order to explore its surrounding landscape.

For the sake of straightforwardness, the three immaculate principles which are utilized are given underneath.

1. Only one egg can be laid at a time by a cuckoo, and the egg is laid in an arbitrary selected nest.
2. The nests containing superior eggs advances to the superseding generation.
3. The probability that the egg placed by a cuckoo will be identified by the host bird is equal to “ P_a ”. Depending upon its choice, the host bird may either discard the egg that was laid by the cuckoo or just leave the nest there unattended and shape a new nest for itself.

According to these rules, an estimate can be made that “ P_a ” fraction of the “ n ” host nests are replaced by new nests. The main steps of the CS algorithm can be summarized by the pseudo code as follows:

Objective function $obj(x)$, $x = (x_1, x_2, \dots, x_d)$

An initial population of n host nests x_i is produced

While ($k < \text{maximum generation}$) or (stopping condition)

A cuckoo is randomly selected

A solution is produced using Lévy flights and then its superiority is found out

Fitness value obj_i is found out
 A nest j amongst n nest is arbitrary selected
 If ($obj_i > obj_j$)
 j is substituted by new solution
 A fraction P_a of the original nests are abandoned by the host bird and new nests are built in its place
 The most appropriate solutions, or the nests with superior solutions is kept
 The current prominent nest or solution is determined by grading the obtained solutions
 End
 Here, “ i ”, “ j ” and “ k ” are variables which are used as counters.
 While generating new solution “ $y(t+1)$ ” for a cuckoo “ i ”, a Lévy flight is accomplished

$$y_i^{(t+1)} = y_i^{(t)} + \alpha \oplus Lévy(\lambda) \quad (12)$$

where, “ α ” represents the step size and its value depends on the problem. Generally, the step size is considered equal to $L/10$, where “ L ” is the characteristic scale of the problem.

A random walk is expressed by the stochastic expression of Equation (12). Usually, Markov chain process is used to define an arbitrary walk whose successive position is dependent on the current position, represented by the first term of this equation, while the second term represents the transition probability. An operator is used in the second term for performing entry wise multiplication, which is represented by “ $\oplus$ ”. Random walk based on Lévy flight is a more efficient of exploring the search space because the step length is much longer. The step length in this scheme can be evaluated from the Lévy distribution as

$$Lévy \Theta u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (13)$$

Lévy walk yields several solutions around the best solution, which quickens the local search. However, far field randomization should be used to produce a considerable fraction of the new solutions and its location should be far from the present best solution. This in turn will protect the solution form getting trapped in a local optimum.

3.4 Bat Algorithm Based Back Propagation

BP is a new metaheuristic algorithm, used in optimization problems, that was modelled by Xin She Yang in 2010 [20]. The BA was inspired by the echolocation ability of the microbats, which use sound waves of varying

frequencies, loudness and pulse rates. During flights, microbats depends on their sound echolocation skills to avoid different obstructions or to discover their preys. Echolocation is a process in which echoes are generated by ultrasonic sound waves. The rebounding echoes are processed by the brain and the auditory system of the bat, and are compared with the generated outgoing waves in order to produce concise images of its environment. Thus, due to these phenomena the bat is able to identify and classify its prey, even in pitch darkness. If the bat is closer to its prey then the rate of pulse is higher and the loudness level of the sound waves it creates is lower. In BA, a microbat is assumed to be a particle having its own distinctive rate of pulse and loudness level.

The rules that were applied to implement the BA algorithm are as listed below:

1. Bats generally use echolocation process in order to sense the distance between other objects and themselves. Moreover, they can discriminate between food or prey and other obstructions present.
2. The arbitrary position and velocity of bats during their flight are represented by " x_i " and " v_i " respectively. A bat is selected which emit sound waves having a fixed minimum frequency " f_{min} ", varying wavelength " λ " and level of loudness " A_0 ". The rate of emission of these emitted waves " r " takes a value that resides within the $[0, 1]$ range, depending on the proximity of their targets from themselves.
3. It is assumed that the level of loudness of the sound waves is decaying from a higher positive value " A_0 " to a fixed lower value " A_{min} ".

BA yields probable solutions, in the case of single-objective optimization problems, as virtual microbats. The velocities " v_i ", positions " x_i " and frequencies " f_i " of the microbats can be computed by the help of following equations

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (14)$$

$$v_i^{k+1} = v_i^k + (x_i^k - x^*)f_i \quad (15)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (16)$$

where, " β " takes an arbitrary value and is distributed uniformly between 0 and 1; " f_{max} " and " f_{min} " denotes the maximum and minimum frequencies respectively; the initial position vector " x_i^0 " is a random vector uniformly distributed within $(x_{i,min}, x_{i,max})$; the initial velocity vector " v_i^0 " is a vector

of all zeros; vector “ x^* ” is the global best solution, got by linking all the objective functions at individual iterations.

A new solution is generated for local search for each “bat” by random walk around the latest best solution.

$$x_{i,next}^k = x_{i,previous}^k + \varepsilon < A_i^k > \quad (17)$$

where, “ ε ” takes an arbitrary value and is distributed uniformly between 0 and 1; “ A_i^k ” represents the average level of loudness of all the bats at time step “ k ”.

The rate of pulse emission “ r_i ” and the value of loudness level “ A_i ” can be found out in each iteration by the following expressions

$$A_i^{k+1} = \alpha A_i^k \quad (18)$$

$$r_i^{k+1} = r_i^0 (1 - e^{-\gamma k}) \quad (19)$$

where, “ α ” and “ γ ” are two constants, which usually takes their values as $0 < \alpha < 1$ and $\gamma > 0$; “ γ ” is a constant that controls the algorithm’s rate of convergence; initial loudness “ A_i^0 ” is a number that is arbitrarily chosen between 1 and 2; “ r_i^0 ” is the initial rate of emission of sound waves, which takes an arbitrary value between 0 and 1.

Random walk is a kind of modification process that prevents the solution from getting stuck in local minima. BA is quicker in comparison to other nature based optimization methods since inertia weight is not required to regulate the velocity of each virtual bat or particle. The bat’s velocity is updated by the help of an inertia weight factor “ w ”. This increases the precision of the planned BA. The efficient weight can be found out as given

$$v_i^{k+1} = w * v_i^k + (x^* - x_i^k) f_i \quad (20)$$

where, “ $w=w_{constant}$ ” which is a constant value of inertia weight.

The process of updating the bat’s position and velocity being similar to the standard PSO, a lower value of inertia weights leads to a local optimum whereas a higher value leads to global optimum.

4 Load Characteristics and Input/Output Variables

Usually, the electrical load demand can be expressed as the sum of following 4 components, at all times

$$TL = T_n + T_w + T_s + T_r \quad (21)$$

where, “ TL ” corresponds to the net load demand of the system; “ T_n ” represents the usual portion of the power system load that is assumed to be occurring consistently throughout the year; “ T_w ” relates to the climate related component of the load; “ T_s ” is the exceptional-occasion related portion of the power system load demand, which is present due to occurrences of unusual or abnormal occasions; “ T_r ” refers to an irregular portion of the load, which is similar to an unexplained noise factor.

The factors that influences the future load demand must be considered as input variables to the load forecasting process. Load demand keeps changing from hour to hour. So, an indicator “ $H(i)$ ”, where $i = 1$ to 24 , is considered. Furthermore, weather also plays a crucial role in load forecasting. Therefore, the past data i.e., the previous day’s load demands and weather condition are taken as the input variables. We have expressed the weather condition of a day mathematically in the following manner: bright sunny day is assigned 0 , an overcast or cloudy day is assigned 0.5 , and a rainy day is assigned 1 . Therefore, the input variable is a 27-dimensional vector which can store the hourly load demand data and also the weather condition. Moreover, since the target vector is the 24-load demands of the day for which the load forecasting is being done, a 24-dimensional vector is taken as the output variable.

5 Case Study

The hourly load demand data and the realistic weather data of the Xingtai Power Plant, situated in the Hebei territory of China, is considered in this paper to evaluate the effectiveness of the proposed methods of STLF.

5.1 Sample Dataset

The hourly load demands and the weather-related data over the period of 10th June to 30th June, 2006 constitutes the historical dataset. The dataset is separated into training dataset, authentication dataset and testing dataset as shown in Table 1. The complete load demand data for the aforementioned duration of Xingtai Power Plant is listed in Table 2.

Table 1 Division of Data sets	
Data Sets	Period
Training Data	10th June–21st June, 2016
Validation Data	22nd June–28th June, 2016
Testing Data	30th June 2016

Table 2 Sample Data

Date	Power Load (MW)	Weather Load
6.10	897 878 826 830 824 854 1037 1094 1176 1272 1300 1317 1281 1304 1286 1287 1286 1178 1034	0.2385 0.2125 0
6.11	930 892 890 846 832 890 1059 1136 1181 1273 1331 1359 1321 1250 1223 1259 1299 1336 1364 1343 1354 1383 1271 1131	0.2152 0.2101 0
6.12	1025 982 944 921 916 987 1142 1246 1277 1359 1408 1441 1460 1380 1342 1322 1378 1379 1390 1389 1408 1345 965 796	0.2415 0.1027 0
6.13	750 733 703 697 718 716 820 937 976 1048 1115 1165 1153 1006 957 949 959 1023 1052 1066 1074 1055 937 843	0.2421 0.1423 0
6.14	776 788 750 754 766 785 956 1052 1139 1240 1273 1335 1321 1254 1241 1274 1333 1345 1349 1346 1351 1338 1237 1096	0.2154 0.1212 0
6.15	970 930 901 898 882 968 1129 1238 1272 1344 1400 1412 1427 1337 1285 1333 1362 1395 1432 1388 1379 1371 1283 1134	0.2523 0.3124 0
6.16	1044 998 959 952 975 1075 1276 1316 1381 1448 1498 1559 1549 1456 1407 1437 1506 1509 1518 1445 1453 1440 1338 1194	0.2103 0.2126 0
6.17	1066 1028 983 981 1000 1080 1305 1398 1438 1534 1559 1583 1583 1515 1498 1512 1547 1589 1611 1623 1589 1587 1493 1315	0.2156 0.2470 0
6.18	1223 1154 1122 1087 1099 1199 1386 1466 1515 1594 1620 1678 1619 1565 1512 1537 1591 1628 1649 1613 1647 1650 1568 1391	0.2380 0.2416 0
6.19	1250 1194 1175 1122 1085 1215 1395 1453 1513 1612 1672 1723 1698 1657 1608 1600 1567 1627 1608 1513 1486 1477 1420 1304	0.2351 0.3215 0
6.20	1169 1136 1070 1060 1057 1137 1330 1408 1470 1541 1595 1640 1566 1550 1533 1564 1580 1572 1585 1567 1509 1493 1406 1244	0.2419 0.2780 0
6.21	1144 1096 1039 983 938 1016 1222 1358 1443 1539 1570 1571 1518 1443 1408 1470 1511 1532 1517 1519 1440 1380 1290 1129	0.2411 0.2801 0
6.22	1039 985 977 934 944 1037 1227 1332 1461 1548 1597 1625 1571 1453 1429 1477 1526 1528 1514 1478 1411 1377 1307 1138	0.2512 0.2456 0
6.23	1056 991 982 949 938 1033 1243 1322 1430 1536 1587 1622 1544 1447 1408 1451 1540 1567 1565 1548 1501 1480 1374 1224	0.2123 0.1476 0
6.24	1102 1039 990 951 947 1037 1249 1353 1419 1543 1608 1591 1549 1423 1392 1432 1504 547 1580 1486 1400 1373 1251 1095	0.2416 0.2134 0

Table 2 Continued

6.25	996 948 925 881 908 984 1227 1317 1410 1513 1578 1566 1525 1449 1369 1430 1471 1442 1384 1287 1261 1311 1224 1077	0.2751 0.2347 0
6.26	994 938 939 901 912 991 1182 1310 1356 1488 1513 1533 1490 1435 1384 1444 1497 1581 1576 1551 1474 1448 1379 1252	0.2415 0.2556 0
6.27	1135 1079 1033 999 988 1091 1290 1392 1445 1557 1608 1599 1557 1465 1401 1434 1501 1579 1561 1585 1537 1520 1441 1326	0.2315 0.2647 0
6.28	1196 1104 993 821 760 728 729 800 838 934 973 1047 1069 1018 1013 1079 1092 1116 1083 1096 1060 1112 1036 954	0.2372 0.2502 1
6.29	861 828 800 798 787 799 845 912 982 1090 1122 1181 1174 1122 1092 1151 1199 1204 1207 1167 1177 1238 1168 1033	0.2134 0.2199 0
6.30	943 914 907 875 873 872 931 976 1062 1144 1213 1263 1231 1196 1150 1190 1212 1231 1223 1228 1245 1317 1214 1081	0.2385 0.2125 0

Before processing the input data i.e., during the pre-processing phase, the dataset is normalized to reside within the range [0,1]. With normalized information, ANN yields better execution results due to the fact that it becomes difficult to apprehend the complex relation existing between the input and the target, if the data ranges are improperly arranged. For observing the improved correctness in the load forecasting and to get the forecasted values in their desired form, the output values are again processed back according to the normalized values.

5.2 Simulation Results

The simulation was performed using the MATLAB 9 software package. Table 3 shows the values of the actual load demand, the forecasted load demand and the percentage of error between the actual and the predicted values. The errors related to different schemes are listed in Table 4.

The forecasting of load was performed by the execution of conservative BP model, GA-BP model, PSO-BP model, CS-BP model and BA-BP model, and their forecasting accurateness were evaluated and compared. Resilient BP was utilized for the training of the neural network as it is a direct adaptive way that facilitates faster learning. The various parametric values selected for the different techniques are given in the appendix. A comparison of the actual demand values and the forecasted values, using the aforementioned techniques, is shown in Figure 3.

Table 3 Demand Forecasting for different schemes

Time (h)	Actual Load (MW)	BP			GA-BP			PSO-BP			CS-BP			BA-BP		
		Forecasted			Forecasted			Forecasted			Forecasted			Forecasted		
		Load (MW)	Error (%)		Load (MW)	Error (%)		Load (MW)	Error (%)		Load (MW)	Error (%)		Load (MW)	Error (%)	
1	943	923	2.12		931	1.27		932	1.16		940	0.32		942	0.11	
2	914	891	2.52		896	1.97		906	0.88		912	0.22		914	0	
3	907	883	2.65		915	0.88		900	0.77		904	0.33		906	0.11	
4	875	853	2.51		862	1.48		866	1.03		873	0.23		873	0.23	
5	873	876	0.34		869	0.46		870	0.34		876	0.34		873	0	
6	872	862	1.15		858	1.60		868	0.46		870	0.23		872	0	
7	931	920	1.18		945	1.50		919	1.29		928	0.32		930	0.11	
8	976	978	0.20		978	0.20		963	1.33		972	0.41		975	0.10	
9	1062	1079	1.60		1055	0.66		1060	0.19		1060	0.19		1061	0.09	
10	1144	1166	1.92		1145	0.09		1134	0.87		1140	0.35		1144	0	
11	1213	1236	1.90		1229	1.32		1202	0.91		1210	0.25		1212	0.08	
12	1236	1286	1.82		1268	0.40		1251	0.95		1261	0.16		1263	0	
13	1231	1224	0.57		1223	0.65		1231	0		1230	0.08		1231	0	
14	1196	1195	0.08		1183	1.09		1185	0.92		1194	0.17		1196	0	
15	1150	1166	0.35		1137	1.13		1138	1.04		1152	0.17		1150	0	
16	1190	1179	0.92		1198	0.67		1201	0.92		1189	0.08		1190	0	
17	1212	1193	1.57		1202	0.82		1217	0.41		1210	0.16		1210	0.16	
18	1231	1232	0.08		1226	0.41		1232	0.08		1230	0.08		1230	0.08	
19	1223	1223	0		1224	0.08		1215	0.65		1220	0.08		1224	0.08	
20	1228	1216	0.98		1228	0		1233	0.41		1224	0.32		1228	0	
21	1245	1223	1.77		1232	1.04		1234	0.88		1243	0.16		1244	0.08	
22	1317	1305	0.91		1315	0.15		1306	0.84		1315	0.15		1317	0	
23	1214	1194	1.65		1222	0.66		1201	1.07		1212	0.16		1213	0.08	
24	1081	1060	1.94		1089	0.74		1074	0.65		1080	0.10		1080	0.10	

Table 4 Errors for different schemes

Schemes	Max. Error	Mean Error	Avg. Percentage Error
BP	24	16.5	1.28
GA Based BP	18	8	0.80
PSO Based BP	13	9.5	0.75
CS Based BP	4	2	0.21
BA Based BP	2	1	0.06

Figure 3 Comparison of forecasted load of different schemes.

From Table 4, it is evident that the BA-BP method leads to the least average of percentage error, which is 0.06, as compared to other schemes. However, CS-BP forecasting scheme also gives an average percentage error of 0.21, which is quite admissible. These two methods are more effective, economical,

and hence can be utilized to enhance the accuracy of the load forecasting process, up to a great extent.

6 Conclusion

The key determination of this work is the exploration of different methods for STLF, which are computational-intellectual approaches in nature. The accuracy of the load forecasting procedure has a massive impact on the operation and the production cost of an electrical utility. Precise load demand forecasting is hence crucial, for which ANN is used here. The hybridized training methods such as CS-BP and BA-BP algorithms were found to be achieving better performance than the conventional BP method, the GA-BP method, or the PSO-BP method. GA-BP method has been found to be decent in terms of providing reasonably superior solutions in reasonably fewer number of iterations. New generation of the preceding population is produced in case of GA by the crossover and mutation process. This may lead to loss of good traits of a chromosome. However, the particle's position and velocity are updated in accordance to the desired criteria in case of PSO algorithm, which helps it in yielding better results than GA. CS algorithm, on the other hand, being quite immune to variation in parameters leads to more encouraging results than the conventional GA and PSO, and finds the global optima quite competently with higher rates of success. BA is faster than other optimization methods because in order to regulate the velocity of virtual bats, the inertia weight factor is not required, hence it helps to achieve the best result among other techniques.

Appendix

PSO-BP parameters

Particle size = 20, Number of iteration = 100, Inertia weight varies from 0.9 to 0.5 as iteration progresses, the cognitive and the social coefficients c_1 and c_2 varies from 0.5 and 2.5 or vice versa as iteration progresses.

CS-BP parameters

$n = 15$ nests, $\alpha = 1$, $P_a = 0.25$

BA-BP parameters

Number of bats = 20; Size of external archive = 100; $f_{max} = 2$, $f_{min} = 0$; $w_{max} = 0.4$, $w_{min} = 0.2$; $\alpha = 0.8$; number of iterations = 25000.

Table A1 Conventional BP parameters

Network Type	MLFNN
Training Algorithm	Back Propagation
Numbers of Layers	3
Hidden Nodes	60–80
Hidden Layer Activation Function	Logsig, Tansig
Output Layer Activation Function	Purelin
Training Parameter Goal	4×10^{-9}
Performance Function	MAPE
Epochs	10000
Learning Rate	0.1

Table A2 GA based BP parameters

Population Size	40
Crossover	0.9
Mutation	0.01
Fitness Function	$F = (\frac{1}{1+MAPE})$
Number of Generation	100

References

- [1] Papalexopoulos, A. D., Hao, S., and Peng, T. M. (1994). An implementation of a neural network based load forecasting model for the EMS. *IEEE Trans. Power Syst.* 9, 1956–1962.
- [2] Chen, H. (1996). A practical on-line predicting system for short-term load. *East China Electric Power* 24.
- [3] Chen, H. (1997). *An Implementation of Power System Short-Term Load Forecasting*. China: Power System Automation.
- [4] Slutsker, I., Nodehi, K., Mokhtari, S., Burns, K., Szymanski, D., and Clapp, P. (1998). Market participants gain energy trading tools. *IEEE Comput. Appl. Power* 11, 47–52.
- [5] Moghram, I., and Rahman, S. (1989). Analysis and evaluation of five short-term load forecasting techniques. *IEEE Trans. Power Syst.* 4, 1484–1491.
- [6] Papalexopoulos, A. D., and Hesterberg, T. C. (1990). A regression-based approach to short-term system load forecasting. *IEEE Trans. Power Syst.* 5, 1535–1547.
- [7] Hagan, M. T., and Behr, S. M. (1987). The time series approach to short-term load forecasting. *IEEE Trans. Power Syst.* 2, 785–791.

- [8] Dhdashti, A. S., Tudor, J. R., and Smith, M. C. (1982). Forecasting of hourly load by pattern recognition: a deterministic approach. *IEEE Trans. Power Apparatus Syst.* 101, 3290–3294.
- [9] Toyada, J., Chen, M., and Inoue, Y. (1970). An application of state estimation to short-term load forecasting, I and II. *IEEE Trans. Power Syst.* 89, 1678–1688.
- [10] Chen, H., and Liu, J. (1998). “A weighted multi-model short-term load forecasting system,” in *Proceedings of the IEEE International Conference on Power System Technology*, New York, NY, Vol. 1, 557–561.
- [11] Rahman, S., and Bhatnagar, R. (1988). An expert system based algorithm for short-term load forecast. *IEEE Trans. Power Syst.* 3, 392–399.
- [12] Lu, C. N., Wu, H. T., and Vemuri, S. (1993). Neural network based short term load forecasting. *IEEE Trans. Power Syst.* 8, 337–342.
- [13] Dash, P. K., Satpathy, H. P., Liew, A. C., and Rahman, S. (1997). A real-time short-term load forecasting system using functional link network. *IEEE Trans. Power Syst.* 12, 675–680.
- [14] Vermaak, J. (1998). Recurrent neural networks for short-term load forecasting. *IEEE Trans. Power Syst.* 13, 126–132.
- [15] Papadakis, S. E. (1998). A novel approach to short-term load forecasting using fuzzy neural network. *IEEE Trans. Power Syst.* 13, 480–492.
- [16] Zheng, T., Girgis, A. A., and Makram, E. B. (2000). A Hybrid wavelet-Kalman filter method for load forecasting. *Electr. Power Syst. Res.* 54, 11–17.
- [17] Yang, H.-T., and Huang, C.-M. (1998). A new short-term load forecasting approach using self-organizing fuzzy ARMAX models. *IEEE Trans. Power Syst.* 217–225.
- [18] Pham, D. T., and Karaboga, D. (2000). “Intelligent optimization techniques, genetic algorithm, tabu search,” in *Simulated Annealing and Neural Networks*, eds D. T. Pham and D. Karaboga (Heidelberg: Springer-Verlag).
- [19] Hassnain, S., and Khan, A. (2007). Short term load forecasting using particle swarm optimization based ANN approach. *IEEE Int. Joint Conf. Neural Netw.* 1, 1476–1481.
- [20] Yang, X. S. (2010). “A new metaheuristic bat-inspired algorithm,” in *Proceedings of the IEEE International Conference on Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (Heidelberg: Springer), 65–74.

Biographies

Papia Ray received her Bachelor of Engineering (Electrical Engineering) degree from Government Engineering College, Bihar and Master of Technology (Power Systems) from National Institute of Technology, Jamshedpur and Ph.D. degree from Indian Institute of Technology, Delhi in 2013. She is presently serving as Assistant Professor in Electrical Engineering Department of Veer Surendra Sai University of Technology, Burla, Odisha. She is a Member of IEEE and Institution of Engineers and Life Member of ISTE.

Sabha Raj Arya received Bachelor of Engineering (Electrical Engineering) degree from Government Engineering College Jabalpur, in 2002, Master of Technology (Power Electronics) from Motilal National Institute of Technology, Allahabad, in 2004 and Ph.D. degree from Indian Institute of Technology (I.I.T) Delhi, New Delhi, India, in 2014. He is joined as Assistant Professor, Department of Electrical Engineering, Sardar Vallabhbhai National Institute of Technology, Surat. His fields of interest include power quality, design of power filters and distributed power generation. He received Two National Awards namely INAE Young Engineer Award from Indian National Academy of Engineering, POSOCO Power System Award from Power Grid Corporation of India in the year of 2014 for his research work. He is also received Amit Garg Memorial Research Award-2014 from I.I.T Delhi from the high impact publication in a quality journal during the session 2013–2014. He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE).

Shobhit Nandkeolyar is presently an Adjunct Professor in the department of Electrical Engineering at Parala Maharaja Engineering College (PMEC), Berhampur. He has completed his M.Tech (Master of Technology) in Electrical Engineering from Veer Surendra Sai University of Technology (VSSUT), Burla with speciality in Power System Engineering. He holds a B.Tech (Bachelor of Technology) degree in Electrical Engineering from Indira Gandhi Institute of Technology (IGIT), Sarang. His area of research consists of Power System optimization, Power System Protection, FACTS Devices and Power System Reliability.