Genet Program Evolvable Mach
DOI 10.1007/s10710-016-9268-6

Dynamic feedback neuro-evolutionary networks for forecasting the highly fluctuating electrical loads

Gul Muhammad Khan[1] · Faheem Zafari[2]

Received: 11 November 2015 / Revised: 24 February 2016

Abstract A computationally efficient and accurate forecasting model for highly dynamic electric load patterns of UK electric power grid is proposed and implemented using recurrent neuro-evolutionary algorithms. Cartesian genetic programming is used to find the optimum recurrent structure and network parameters to accurately forecast highly fluctuating load patterns. Fifty different models are trained and tested in diverse set of scenarios to predict single as well as more future instances in advance. The testing results demonstrated that the models are highly accurate as they attained an accuracy of as high as 98.95 %. The models trained to predict single future instances are tested to predict more future instances in advance, obtaining an accuracy of 94 %, thus proving their robustness to predict any time series.

Keywords Very short term electric load forecasting (VSTLF) · Recurrent neural networks · Cartesian genetic programming evolved recurrent neural network (CGPRNN) · Neuro-evolution

✉ Gul Muhammad Khan
gk502@uetpeshawar.edu.pk

Faheem Zafari
fahim.zafari@nwfpuet.edu.pk

[1] Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan

[2] Department of Computer and Information Technology, Purdue University, West Lafayette, IN, USA

Published online: 18 May 2016

1 Introduction

Due to inefficient utilization of Power plants and other resources of energy, the cost per unit of electrical energy has increased over the years [1]. The efficient utilization of the electrical power generation plant, which can be achieved by maintaining a balance between the demand and generation of electricity [2], can assist in reducing or eliminating the electricity cost problems. To maintain the balance, there is a need for a forecasting system that can predict the demand of electricity at a specific time in future. The advent of smart grids/meters allows to obtain the load data in a specific area that can be used to forecast the electrical load for the future. Once the load is forecasted, the electrical power generation plant can then be used accordingly to generate the specific amount of electrical power. This facilitates the efficient and optimum utilization of the resources through proper fuel scheduling and maintenance of the power generation system [1]. In the absence of a forecasting system, the electrical power plant usually produces more than demand. However, the storage of excessive power is not efficient thus causing the system to be over-utilized and the cost per unit generation to increase. On the other hand, if lesser power is generated than the demand, the voltage goes down and the losses on the distribution lines increases. Also tripping at various distribution points damage various home appliances and cuts off the electric power supply to them. Hence load forecasting is really an integral part of modern electrical power distribution. Research has continued for an optimum load forecasting methods.

Numerous electrical load forecasting techniques have been used over the years such as fuzzy logic, auto-regression and vector regression models [3]. Artificial neural networks (ANNs) tend to perform better compared to other models because of its ability to learn and adapt to the scenario. The comparison of fuzzy logic, neural networks and autoregression model shows that ANNs and fuzzy logic are superior compared to autoregression model. Numerous ANN models have been proposed; each having its own advantages. Some of the models include feed-forward neural network, cascaded neural networks, self-supervised neural network and recurrent neural networks. Obtaining a neural network with optimum performance for handling the load forecasting is a tedious task. However, evolutionary techniques can play a significant role in automatic generation of a neural network that tends to perform better in terms of accuracy, in load forecasting.

Load forecasting was initially classified into three broad categories:

- Long term load forecasting, which is for the duration of over a year
- Medium term load forecasting, which is for a week duration
- Short term load forecasting, which is for a duration of 1 h to 1 week [1].

However, the increase in the demand for energy and the need for a secure, flexible and efficient electricity infrastructure, requires forecasting at very small intervals such as half hour, or even a few minutes. The data obtained through smart grids and smart meters has now made it possible for researches to produce forecasting models that can predict the load for very short term of half an hour. This facilitates the forecasting of future load in a specific area for very short duration, hence the power producing plant is able to adjust itself to the requirements of the locality.

Because very short term load forecasting is of significant importance, and is expected to play an integral role in load forecasting, various researches have proposed models for very short term load forecasting (VSTLF). Models based on VSTLF are much more robust compared to other traditional approaches.

The ANN based model discussed in this paper is known as Cartesian genetic programming evolved recurrent neural networks (CGPRNN) [1] and the current work is an extension of [4]. The number of feedback paths are varied for a range of networks to obtain a network with the best possible prediction performance. The models are initially trained using the historical load data obtained from UK National Grid. These models are trained to forecast the next half an hour in advance. After the training phase, the models are tested to forecast the load for next half an hour, 12 and 24 h for a period of a month; based on the load data of past 24 h using sliding windows mechanism. The testing results show that these models are highly efficient in load forecasting having an accuracy as high as 98.95 %. A thorough review of the work done related to load forecasting, and neuro-evolution is presented in the next sections.

2 Load forecasting

Various statistical models have been proposed in order to predict the future load requirements including support vector regression, local prediction framework for load forecasting, autoregressive model, auto-regressive moving average model (ARMA) and autoregressive moving average with exogenous variable (ARMAX) and Kalman filtering [5]. Kalman filtering based load forecasting are one of the oldest predictive techniques utilized for power load forecasting. One of the most common form of Kalman filtering based predictive technique is the phase locked loop filter [6]. It utilizes pattern recognition techniques along with weather patterns to predict hourly electric load. A hybrid technique involving a combination of the Kalman Filter along with Elman Neural Network was proposed in [6]. It used Kalman filter to predict the linear parameters and Elman Neural Network to predict the the non-linear parameters. Al-Hamadi and Soliman [7] utilized a technique based on the hybridization of the Kalman filter and fuzzy logic to predict the short term peak load based on the current weather patterns as well as the recent past history. Lim [8] developed an improved short-term load forecasting algorithm for an arbitrary education institute for fluctuations in the daily, weekly, and yearly load patterns. He analyzed and correlated it with temperature trends during the respective periods. An optimal exponential smoothing coefficient according to the selected period was used for the building load forecasts. The estimated optimal exponential smoothing coefficient derived for each selected period was then compared with past load patterns. The proposed algorithm was verified by simulation of the electric demands showing that the forecasting accuracy of the proposed algorithm is improved comparing with traditional exponential smoothing analysis having a mean absolute percentage error (MAPE) value of 3.61 %. Equation 1 shows the mathematical equation for MAPE. L_{Fi} is the ith forecasted value while L_{Ai} is the ith actual value. N is the number of instances.

$$MAPE = \left(\frac{1}{N} \sum_{i=1}^{i=N} \left| \frac{L_{Fi} - L_{Ai}}{L_{Ai}} \right| \right) \times 100 \tag{1}$$

Ramos et al. [9] developed a load forecasting method for short-term load forecasting (STLF), based on Holt–Winters exponential smoothing and an artificial neural network (ANN). His main contributions was the application of Holt–Winters exponential smoothing approach to the forecasting problem and, as an evaluation of the past work, data mining techniques were also applied to short-term Load forecasting. Both ANN and Holt–Winters exponential smoothing approaches were compared and evaluated obtaining a MAPE of 7.6 % being the best performance. Support vector machine (SVM) based supervised learning methods have also been employed for Daily Load Forecasting. Chen et al. [10] developed an SVM based model for daily peak load estimation based on a lead time of 31 days. The error recorded for the proposed model was within 2–3 %. Pai and Hong [11] introduced a load forecasting model that was a combination of the genetic algorithm (GA) and recurrent SVMs (RSVMG). The GA in the model predicted the free parameters of the SVM. The proposed model produced better results than SVMs, artificial neural networks (ANNs) and regression models.

Hong et al. [12] proposed three key elements of long term load forecasting: predictive modeling, scenario analysis, and weather normalization. The predictive models attained high accuracy from hourly data, in comparison to classical methods of forecasting using monthly or annual peak data. Further development of probabilistic forecasts through cross scenario analysis has enhanced the results. They have achieved an accuracy of 4.2 % on average. Chen et al. [13] proposed a two-stage identification and restoration method to detect the typical patterns of inaccurate measurement and abnormal disturbance based on statistical criteria independent with normal distribution in first stage and historical trend in second stage using frequency domain decomposition. The deviations of the data measurements from the typical daily curve obey normal distribution and were used as criteria in the second stage. The effectiveness of the proposed methodology has been confirmed by examples in real bus load forecasting systems obtaining a MAPE of 1.8 %. Hooshmand et al. [14] proposed a novel 2-step algorithm for STLF. During the first step, ANN was used in combination with wavelet transform to forecast the load for the next 24 h. The inputs applied are weather parameters, and the load data of the previous day. During the second step, a combination of wavelet transform, adaptive neural fuzzy inference system and similar hour method were used to improve the accuracy of the results obtained in step 1. The experimental results showed that the model has a MAPE value of as low as 1.603 %.

Mandal et al. [15] used the similar days approach along with ANN for short term load forecasting. Since learning the entire similar days data is a cumbersome job, the model uses Euclidean norm along with weighted factors for evaluating the similarity between the previously searched days and the forecasted day. The model's accuracy is further increased by adding temperature as a climate factor. Panday et al. [16] proposed procuring power through energy exchange based on forecasting of day-ahead load demand. They discussed the role of ANN in day-ahead hourly forecast of the power system load in Uttar Pradesh Power Corporation India Limited (UPPL) so as to minimize the error in demand forecasting. A new artificial neural network (ANN) has been designed to compute the forecasted load of UPPL. The data used in the modeling of ANN are hourly historical electricity load data. The ANN model is trained on hourly data from UPPCL from April, 2014 to June, 2014 and tested on out-of-sample data of 2 weeks. Simulation results obtained showed that a day-ahead hourly forecast of load using proposed ANN was very accurate having an average MAPE of 3.05 %. Sahay et al. [17] designed a new artificial neural network (ANN) to compute the forecasted load. The data used in the modeling of ANN were hourly historical data of the temperature and electricity load. The ANN model was trained on hourly data from Ontario Electricity Market from 2007 to 2011 and tested on out-of-sample data from 2012. Simulation results obtained showed that day-ahead hourly forecasts of load using proposed ANN generated an average MAPE of 2.05 % with temperature and 2.23 % without temperature. In short, ANN models tend to forecast the load requirements accurately compared to other models, that makes it an attractive area of research.

3 Neuro-evolution

The evolution of various attributes of a neural network is called neuro-evolution (NE). The attributes, which might be evolved are node activation functions, connection weights, network topology and connection type. The genotype represents these parameters and is evolved until the desired fitness is achieved that is called the phenotypic behavior. The NE design is affected by the encoding choice as the solutions search space is dependent on it. In NE, a single or a combination of network parameters may be evolved. The evolution of only connection weights results in a fixed topology that restricts the solution space of the network and evolution necessarily does not result in the optimum solution for a specific problem.

Topology and weight evolved artificial neural network (TWEANN) increased network efficiency by evolving both connection weights and topology [18]. The experimental results shown in [18] highlight the fact that evolving both topology and weights provide a better network when compared to evolving only weights or topology alone. The symbiotic adaptive neural network evolution (SANE) simultaneously evolves neuron population and network topology. An extension of SANE known as enforced sub-population (ESP) evolves the sub-population of neurons that are present in the hidden layer instead of evolving a single neuron population [19]. The three major problems in neuro-evolution are

- Tracking the genes that contain historical markings and allow an easy crossover among various topologies.
- Starting a simple structure and then increasing its complexity through generations.

- Using speciation for innovation protection.

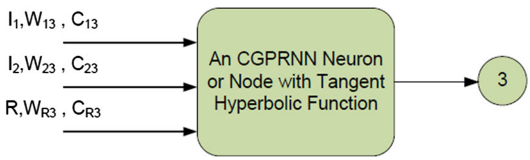
These problems were solved by NE of augmenting topologies (NEAT) which was developed by Stanley and Miikkulainen [20]. NE has been influential in providing ANNs that can be used to perform various tasks. Section 4 presents an insight into CGPRNN, which is actually a model obtained by evolving a recurrent ANN using Cartesian genetic programming (CGP).

4 Cartesian genetic programming evolved recurrent neural network (CGPRNN)

Recurrent networks are of great significance when dealing with a wide domain of nonlinear and dynamic system. Khan et al. [5] proposed Cartesian genetic programming evolved recurrent artificial neural network (CGPRNN) for solving non-linear control problems. CGPRNN utilizes Cartesian genetic programming (CGP) for evolution of RNNs. Cartesian genetic programming is a genetic programming technique devised by Vašíček and Sekanina [21] and Rothermich and Miller [21, 22]. In CGP, a two dimensional graphic representation is used to generate a digital circuit or computer program. CGP is a highly flexible and efficient genetic programming technique. CGP utilizes arrays and Cartesian framework for representation of its architecture [5]. CGPRNN is a neuro-evolutionary algorithm used for evolving recurrent neural networks. It utilizes the superior capability of CGP for generating a recurrent ANN. The difference between CGPRNN and other classes of CGPANN is that it utilizes a feedback mechanism i.e. one or many outputs are fed back into the system as input(s). The difference between the architecture of CGPRNN and traditional RNN is that the neurons derived by CGPRNN network are not connected entirely. Also, all the input layer neurons are not supplied with program inputs. This feature allows CGPRNN to produce topologies, which is efficient in terms of computational cost and hardware implementation.

CGPRNNs genotype consists of nodes which represent the neurons of the RNN. As shown in Fig. 1, there are certain inputs, connections functions and weights affiliated with the nodes. The inputs supplied are of 3 types: the program inputs, the inputs coming from previous nodes and the input coming from the feedback. In the first layer of the genotype in CGPRNN, the inputs are either recurrent or the system inputs. For the following layers, the existence of a feedback path depends upon the fact whether the selection of a feedback input which is to be utilized as a node input, is selected through mutation or not. A node is said to be connected if its connection

Fig. 1 CGPRNN node with 3 inputs



value is one while it is said to be disconnected if the connection value is zero. The connections weights are generated randomly between -1 and $+1$. However, the feedback input weight is always $+1$. All the inputs along with the weights of the connected inputs are multiplied and then added. It is then forwarded to an activation function, which is either linear or non-linear such as log-sigmoid, linear, tangent hyperbolic or step function. The production of output at every node depends upon the activation function. The node output is then utilized as the system output or as an input to the next node. The output(s) of the genotype is either any node output(s) or the program input(s). The output of the genotype can also be used as the feedback into the nodes if the recurrent input is already connected. The CGPRNN genotype is then evolved continuously through mutation until a desired fitness is achieved. The state unit weights along with the connections are frozen and the resultant genotype is then transformed to the ultimate RNN [5].

Equation 2 shows which system input is connected with the z input of the y neuron in the x genotype. The pseudo random generator (PRG) is used to choose whether to connect the system input, the output of previous node or a recurrent output as an input with a specific input of a node. *SOR* is the recurrent system output which is fed back into the system as an input. The number of recurrent paths is given by Eq. 3, in which N_r is the maximum feedback paths.

$$geno(x, y, z) = PRG[I : geno(x, y-1)\ldots geno(x, 1)\ldots SOR] \tag{2}$$

$$R = [1, 2, 3, 4\ldots N_r] \tag{3}$$

Let j be the system output given in Eq. 4.

$$j = [1, 2, 3\ldots N] \tag{4}$$

where N is the total number of outputs. The weights to a specific recurrent output is assigned using Eq. 5

$$W(SOR(R,j)) = PRG[-1\ldots+1] \tag{5}$$

R is a specific feedback while j is already defined in Eq. 4. So the weight to a specific recurrent output is assigned using a pseudo random generator, which generates weights for a specific recurrent output.

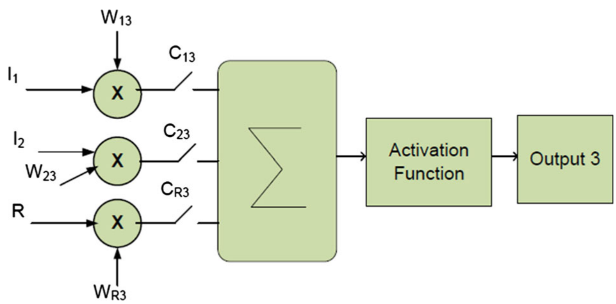


Fig. 2 Internal structure of a CGPRNN node

Figure 1 shows a CGPRNN node with 3 inputs. Figure 2. shows the internal view of a CGPRNN node. There are three unconnected inputs, I_1, I_2, and R which are multiplied with the corresponding weights W_{13}, W_{23}, W_{R3}. After multiplying the inputs with their corresponding weights, they are all added and supplied into an activation function. The activation function produces the output for the specific node. Figure 3a shows the genotype for a 2×2 CGPRNN network. Figure 3b shows the block diagram representation of the genotype shown in Fig. 3a, while Fig. 3c shows the graphical representation of the genotype in Fig. 3a. Figure 4 shows the phenotype for the CGPRNN genotype shown in Fig. 3. Equations 6 and 7 show the mathematical expression for the CGPRNN phenotype in Fig. 4.

$$O_3 = logsigmoid(w_{13}I_1 + w_{23}I_2 + w_{63}I_{3R}) \tag{6}$$

where I_R is the recurrent input i.e. the feedback from output 6

$$O_6 = tanh(w_{36}O_3 + w_{26}I_2 + w_{66}I_{6R}) \tag{7}$$

The next section will present the application of the proposed algorithm for VSTLF, providing the experimental setup, and the detailed results and analysis.

5 Application of CGPRNN to very short term load forecasting (VSTLF)

In this section, the exploration of CGPRNN for its capabilities to produce an efficient prediction model is presented and its performance is evaluated to predict the highly varying load data. This section comprises of a range of subsections describing the experimental setup, results and analysis for various scenarios.

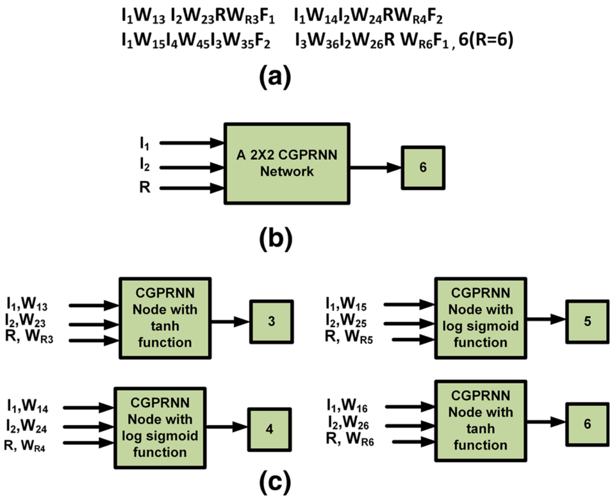


Fig. 3 **a** The genotype for a 2×2 CGPRNN network. **b** Block representation of genotype in (**a**). **c** Graphical representation of the genotype in (**a**)

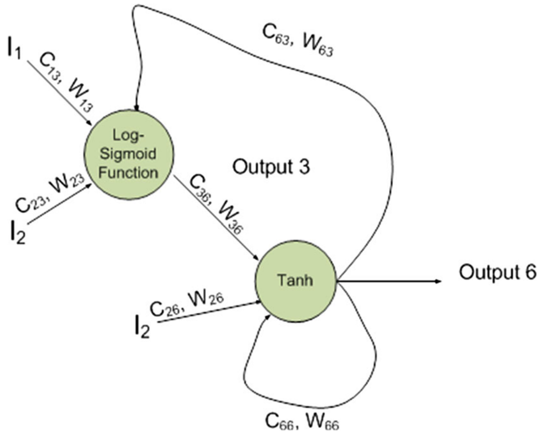


Fig. 4 Phenotype for the CGPRNN genotype shown in Fig. 1

5.1 Experimental setup

The half hourly historical data of electrical loads obtained from the UK power grid was used to train CGPRNN networks. The load of January, 1997 was used to train the system, while testing was done for the the entire year 1998. Initially a random population of CGPRNN is generated. The number of inputs into the network are 48 (24 h) for CGPRNN network, plus the number of feedback inputs. The mutation rate (μ_r) was 10 % as it provides better results and fast learning [2]. Single row is used for CGPRNN (Number of Nodes = Number of Columns), as it results in infinite graphs and better evolutionary results. An evolutionary strategy of $1 + \lambda$ is used where λ is set at 9 in this case [5]. Mean absolute percentage error (MAPE) is used

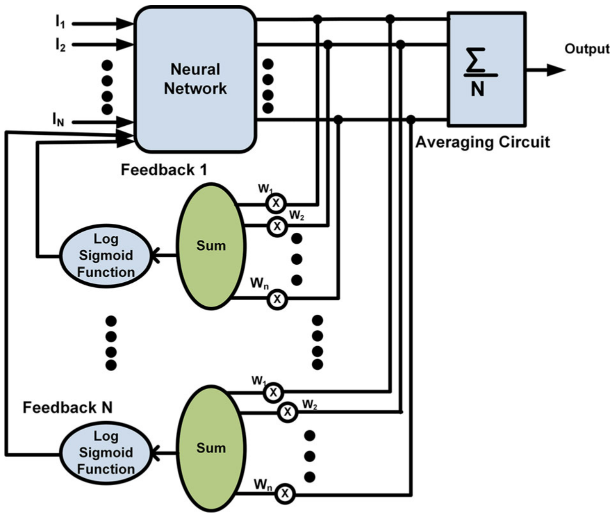


Fig. 5 CGPRNN network with N feedback paths

as the evaluation parameter for determining the fitness of the individual network. The network with the best MAPE value is promoted to the next generation. The selected network acts as parent for the next generation, which is mutated to produce offspring. The process continues until the desired fitness is attained or the maximum number of generation is reached. Each experiment is performed for one million generation.

Initially the CGPRNN model is evaluated for its performance. Various network architectures with different numbers of nodes ranging from 50 to 500 with an increment of 50 nodes are explored. The model was trained for half hourly load forecasting. However, the model was tested to predict the load for next half an hour, 12 and 24 h for an entire month using the historical data of past 24 h. Network with both 24 and 48 feedback paths were used. Figure 5 shows a CGPRNN network with N feedback paths. Figure 6 shows how the sliding window mechanism works for the network when the next half hour load is predicted based on the historical data of past 24 h. The next half an hour is predicted, then the window slides and then utilizes the forecasted half an hour along with the previous data for predicting the next half an hour. The next subsection will provide detailed results and analysis for all these scenarios of the proposed model.

5.2 Results and analysis

In this subsection, we will discuss the performance of CGPRNN under various experimental conditions. Table 1 highlights the testing results for a CGPRNN network with 24 feedback paths and number of nodes ranging between 50 and 500 with an increment of 50 nodes; which is used to predict the next half hour load for a month based on the load data of past 24 h. A MAPE value as low as 1.128 % is obtained. Table 1 also shows the average MAPE across months and nodes demonstrating the best performance of CGPRNN on average. Standard deviation (SD) across various nodes and months is also calculated. The SD across the number of nodes is almost the same apart from the 200 node network. Across the months, it is evident that it is highest in the months of June, July and August. This is due to high fluctuation in load statistics that makes it difficult to predict.

Table 2 shows the MAPE values for CGPRNN network with 24 feedback paths and different number of nodes. The model predicts next 24 h of load based on the past 24 h of load data, for a month. The month of March and July (with 250 nodes)

Fig. 6 Sliding windows mechanism based on 24 h historical data

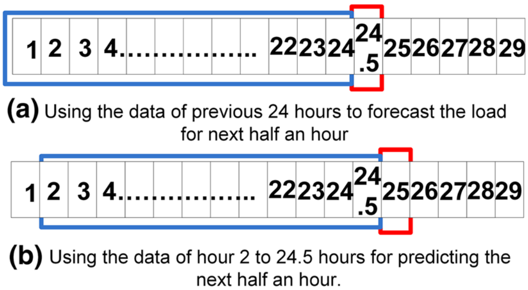


Table 1 The testing results for CGPRNN network with 24 feedback path scenario and 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 nodes; which uses load data of past 24 h to forecast the load for next half an hour for a month

Nodes	50	100	150	200	250	300	350	400	450	500	Average	SD
Jan	**1.256**	2.003	1.321	1.883	1.283	2.101	1.567	1.618	1.346	1.328	1.571	0.319
Feb	**1.211**	1.936	1.248	1.694	1.211	1.849	1.468	1.628	1.268	1.269	1.478	0.279
Mar	**1.181**	1.781	1.204	1.613	1.195	1.747	1.465	1.507	1.245	1.240	1.418	0.236
Apr	2.232	3.630	2.066	6.051	**1.932**	4.378	2.210	2.547	3.067	2.209	3.032	1.319
May	3.379	5.614	2.928	10.837	**2.869**	7.166	3.109	4.269	4.851	3.228	4.825	2.529
Jun	4.735	7.920	3.994	16.189	**3.977**	10.209	4.099	6.421	6.899	4.440	6.888	3.860
Jul	4.545	7.708	**3.810**	16.159	3.842	10.067	3.916	6.202	6.721	4.220	6.719	3.904
Aug	5.617	9.146	**4.704**	19.597	4.706	12.185	4.673	7.660	8.226	5.211	8.172	4.707
Sep	3.558	5.998	**2.988**	11.477	2.956	7.650	3.041	4.318	5.029	3.311	5.033	2.737
Oct	1.963	3.333	1.810	4.466	**1.736**	3.927	1.946	2.254	2.395	1.915	2.574	0.978
Nov	**1.128**	1.979	1.186	1.830	1.153	2.188	1.421	1.483	1.230	1.199	1.480	0.385
Dec	**1.232**	1.973	1.330	1.938	1.239	2.022	1.545	1.509	1.327	1.271	1.539	0.321
Average	2.670	4.418	2.383	7.811	2.341	5.457	2.538	3.451	3.634	2.570		
Standard deviation (SD)	1.630	2.730	1.259	6.721	1.280	3.824	1.183	2.253	2.597	1.455		

Bold values show the best performance for the given number of nodes

Table 2 The testing results for CGPRNN network with 24 feedback path scenario and 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 nodes; which uses load data Of past 24 h to forecast the load for next 24 h for a month

Nodes	50	100	150	200	250	300	350	400	450	500
Jan	8.962	9.713	8.952	9.061	8.884	9.634	8.612	8.712	8.78	8.747
Feb	7.987	8.467	7.928	7.901	7.907	8.323	7.607	7.814	7.80	7.781
Mar	**7.355**	**7.570**	**7.309**	**7.220**	7.401	**7.430**	7.282	7.18	7.903	**7.245**
Apr	7.939	9.645	7.969	12.576	7.373	10.282	7.441	7.152	9.40	7.801
May	8.623	10.918	8.443	16.167	7.890	12.364	7.722	7.256	10.64	8.350
Jun	8.719	11.515	8.357	19.050	7.638	13.737	7.270	6.723	11.42	8.301
Jul	8.162	10.806	7.849	18.923	7.347	13.297	7.087	6.570	10.73	7.799
Aug	8.367	11.510	7.899	20.364	**7.290**	14.390	**6.831**	**6.371**	11.50	7.881
Sep	8.729	10.924	8.443	16.235	8.018	12.430	7.765	7.262	10.50	8.350
Oct	9.309	9.959	9.143	10.735	9.260	10.071	9.123	9.367	9.44	9.072
Nov	8.583	9.315	8.438	8.591	8.520	9.341	8.208	8.506	8.23	8.280
Dec	8.867	10.402	9.018	9.267	8.470	10.102	7.940	7.117	8.79	8.549

Bold values show the best performance for the given number of nodes

provide the optimum results while the best result is obtained for the month of March with 200 nodes i.e. the MAPE value is 7.220. The MAPE values in Table 2 are higher as compared to Table 1 because the model was trained to predict the next half hour only, while the results in Table 2 are for predicting 24 h of load.

We have also tested the capability of 24 feedback network to predict the next 12 h data based on the 24 h data history for a period of one month. Table 3 provides the MAPE values for the networks with number of nodes varying between 50 and 500 with a step of 50 nodes. Most of the optimum results are for the month of March

Table 3 The testing results for CGPRNN network with 24 feedback path scenario and 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 nodes; which uses load data of past 24 h to forecast the load for next 12 h for a month

Nodes	50	100	150	200	250	300	350	400	450	500
Jan	7.277	7.904	7.380	7.926	7.243	7.243	8.442	7.357	7.192	7.245
Feb	7.119	7.352	7.115	7.255	7.054	7.054	7.454	6.926	6.930	7.021
Mar	**6.427**	**6.490**	**6.452**	**6.527**	**6.446**	**6.446**	**6.570**	**6.481**	6.766	**6.413**
Apr	7.127	8.525	7.014	10.966	6.621	6.621	8.718	6.573	6.380	6.879
May	8.075	10.123	7.746	14.467	7.218	7.218	10.610	6.943	6.641	7.624
Jun	8.619	11.032	8.161	17.124	7.418	7.418	11.803	6.867	6.400	8.050
Jul	8.421	10.669	8.000	17.212	7.314	7.314	11.604	6.788	6.305	7.875
Aug	8.612	11.172	8.067	18.487	7.339	7.339	12.452	6.632	**6.185**	7.971
Sep	8.439	10.322	8.018	14.499	7.618	7.618	10.750	7.169	6.760	7.900
Oct	8.235	9.117	8.048	9.755	8.031	8.031	9.044	7.970	8.003	7.959
Nov	7.188	7.799	7.166	7.683	7.189	7.189	8.486	7.174	7.321	7.067
Dec	7.310	8.337	7.541	8.203	7.125	7.125	8.701	6.970	6.387	7.196

Bold values show the best performance for the given number of nodes

and August (single result for a network with 450 nodes). A comparison of Tables 2 and 3 highlight the fact that the overall MAPE values are better for the model when predicting the next 12 h for a month compared to predicting the next 24 h for a month. This is because 12 h is a shorter time period compared to 24 h and lesser error accumulates in 12 h compared to 24 h.

Table 4 highlights the testing results for a CGPRNN network with 48 feedback paths while the number of nodes ranges between 50 and 500 nodes with a step of 50 nodes. The network was used to forecast the next half an hour based on the load data of past 24 h. Most of the accurate results are for the month of March and July, while the best result is for the month of July with a network of 450 nodes that has a MAPE value of 1.046 %. A comparison of Tables 1 and 4 highlights the fact that the network with 48 and 24 feedback path perform almost identical.

The 48 feedback network is also tested to predict more instances of future load in advance. Table 5 presents the MAPE values for CGPRNN network with 48 feedback paths, while the number of nodes increases from 50 to 500 nodes with an increment of 50 nodes which use the data of past 24 h to predict the load for next 24 h, although the network was trained with load data of half hour. This is why the MAPE values are comparatively higher than Table 4.

Table 6 presents the MAPE values for CGPRNN network with 48 feedback paths with varying number of nodes using the data of past 24 h to predict the load for next 12 h, although the network was trained to predict the load data of half hour. The MAPE values are comparatively lower than Table 5 due to the aforementioned reasons.

Table 7 provides a general comparison amongst CGPRNN and various other prediction models which have been used for half hourly load forecasting. The models are used for very short term forecasting. The models in [14, 23] forecast the half hourly electricity load for next 24 h while our CGPRNN model forecasts it for an entire month. The model in [24] predicts the half hourly for an entire month with

Table 4 The testing results for CGPRNN network with 48 feedback path scenario and 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 nodes; which uses load data of past 24 h to forecast the load for next half an hour for a month

Nodes	50	100	150	200	250	300	350	400	450	500
Jan	2.251	1.377	2.280	2.210	1.415	1.376	1.417	1.249	1.477	**1.173**
Feb	1.964	1.413	1.986	2.034	1.404	1.416	1.418	1.255	1.480	**1.182**
Mar	1.699	1.326	1.752	1.767	1.331	1.276	1.328	1.137	1.438	**1.081**
Apr	3.124	2.078	3.202	5.849	2.188	2.483	2.381	1.972	2.288	**1.897**
May	5.320	3.031	5.359	9.726	3.156	3.769	3.431	**2.852**	3.225	2.877
Jun	8.008	4.134	8.068	13.929	4.287	5.255	4.667	**3.932**	4.291	4.060
Jul	7.967	3.978	7.982	13.897	4.094	5.050	4.501	**3.776**	4.103	3.901
Aug	9.831	4.768	9.871	16.539	5.021	6.184	5.420	**4.634**	4.892	4.858
Sep	5.856	3.020	5.787	10.212	3.097	3.965	3.643	**2.917**	3.206	2.977
Oct	3.206	1.788	3.280	4.322	1.842	2.128	2.205	1.738	2.009	**1.652**
Nov	2.325	1.258	2.331	2.119	1.276	1.231	1.294	1.127	1.345	**1.046**
Dec	2.309	1.322	2.305	2.465	1.405	1.392	1.377	1.272	1.376	**1.177**

Bold values show the best performance for the given number of nodes

Table 5 The testing results for CGPRNN network with 48 feedback path scenario 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 nodes; which uses load data of past 24 h to forecast the load for next 24 h for a month

Nodes	50	100	150	200	250	300	350	400	450	500
Jan	9.639	8.657	9.338	10.118	8.735	9.229	8.859	8.882	8.518	9.024
Feb	8.204	7.637	8.053	8.876	7.701	8.250	7.870	7.868	7.551	7.992
Mar	**7.466**	**7.400**	**7.521**	**7.561**	**7.396**	**7.464**	**7.289**	**7.282**	**7.325**	**7.396**
Apr	8.671	7.291	8.963	14.199	8.229	8.241	7.918	7.986	7.316	7.644
May	9.761	7.602	9.877	18.358	8.571	9.094	8.459	8.475	7.579	8.242
Jun	10.212	7.129	10.254	21.520	8.328	9.342	8.405	8.398	7.062	8.208
Jul	9.711	6.992	9.843	21.399	7.911	8.699	7.888	7.897	6.915	7.728
Aug	10.278	6.726	10.369	22.721	7.811	9.028	8.003	7.936	6.657	7.811
Sep	9.787	7.648	9.968	18.355	8.537	9.128	8.635	8.486	7.597	8.321
Oct	9.779	9.296	9.930	11.273	9.601	9.320	9.411	9.302	9.080	9.290
Nov	9.435	8.382	9.010	9.331	8.377	8.763	8.541	8.472	8.146	8.686
Dec	9.913	7.573	9.134	11.758	7.894	9.579	8.653	8.723	7.503	8.785

Bold values show the best performance for the given number of nodes

Table 6 The testing results for CGPRNN network with 48 feedback path scenario and 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 nodes; which uses load data of past 24 h to forecast the load for next 12 h for a month

Nodes	50	100	150	200	250	300	350	400	450	500
Jan	8.777	7.222	8.570	8.959	7.602	7.454	7.310	7.380	7.175	7.331
Feb	7.506	6.847	7.689	8.176	7.075	7.322	6.981	7.063	6.816	7.100
Mar	**6.640**	**6.433**	**6.955**	**6.891**	**6.660**	**6.486**	**6.364**	**6.399**	**6.483**	**6.424**
Apr	7.294	6.445	7.596	12.369	7.102	7.3937	7.016	6.946	6.493	6.814
May	8.314	6.825	8.411	16.176	7.744	8.568	7.784	7.705	6.845	7.608
Jun	8.747	6.694	8.612	18.957	7.966	9.275	8.201	8.116	6.715	8.061
Jul	8.518	6.625	8.432	19.094	7.829	9.043	7.961	7.952	6.645	7.897
Aug	8.840	6.469	8.878	20.199	7.773	9.303	8.127	8.018	6.491	8.012
Sep	8.506	7.063	8.899	16.090	7.954	8.865	8.220	8.013	7.021	7.949
Oct	8.742	7.965	9.147	10.349	8.475	8.351	8.172	8.129	7.845	8.087
Nov	8.891	7.205	8.535	8.423	7.447	7.297	7.226	7.240	7.086	7.268
Dec	8.968	6.606	8.616	10.306	6.985	7.825	7.290	7.357	6.572	7.261

Bold values show the best performance for the given number of nodes

a MAPE value as low as 1.55 %. The results show that CGPRNN is superior to all the other models, having a MAPE value as low as 1.046 %.

5.3 Further analysis

The CGPRNN systems with both 24 and 48 feedback paths perform almost identically. CGPRNN model with 24 and 48 feedback paths are also evaluated in

Table 7 Comparison of various networks used for electric load forecasting

S.no.	Model	MAPE (%)
1	CGPRNN (proposed model)	**1.046**
2	ANN method [23]	1.01
3	Hooshmand et al. [14] Model	1.603
4	ANN based model in [24]	1.55

Bold value shows the best performance for the given number of nodes

forecasting the load for next 12 and 24 h for a month on the basis of the past 24 h load data. Since the model was trained only for predicting the next half hour load, so forecasting the next 12 and 24 h results in more erroneous outcomes compared to forecasting the next half hour. To highlight the cost efficiency of the model, Fig. 7 shows a CGPRNN network with 50 nodes and 24 feedback paths that predicts the next half an hour load based on the load data of past 24 h (48 inputs). Despite the presence of 24 feedback paths, only a single feedback path is utilized. This is due to the fact that our utilized model evolves to obtain an accurate and cost efficient system. Figure 7 is a testament to the fact that we necessarily do not need highly complex systems to obtain accurate performance, but it is the combination of nodes, weights and connections that must be optimally connected to form a network that results in high accuracy. CGPRNN is superior is due to the fact that recurrent connectivity, inter node connectivity, and selection amongst many input is flexible in comparison to any other known algorithm. The final phenotype of the system in Fig. 7 can be modeled using mathematical expression as provided by Eqs. 8–13.

$$Y_{50} = \frac{8I_{47} + 2O_{100}}{10} \tag{8}$$

$$\begin{aligned} O_{100} = w_1 I_1 + w_{47A} I_{47} + w_{47B} I_{47} + 3f_{100}(I_{47} + I_1 \\ + 3f_{87}(I_{14}, I_{32}, I_{38})) + 2f_{83}(I_{20}, I_{41}, I_{42}, \\ 2f_{80}(2I_1, 2I50, I_{21})) \end{aligned} \tag{9}$$

$$\begin{aligned} f_{100} = logsigmoid(w_{47A} I_{47} + w_{47B} I_{47} + w_1 I_1 + \\ 2f_{87}(I_{14}, I_{32}, I_{38}, 2f_{83}(I_{20}, I_{41}, I_{42}, \\ 2f_{80}(2I_1, 2I_{50}, I_{21}))) \end{aligned} \tag{10}$$

$$\begin{aligned} f_{87} = logsigmoid(w_{14} I_{14} + w_{32} I_{32} + w_{38} I_{38} \\ + 2f_{83}(I_{20}, I_{41}, I_{45}, 2f_{80}(2I_1, 2I_{50}, I_{21}))) \end{aligned} \tag{11}$$

$$\begin{aligned} f_{83} = logsigmoid(w_{20} I_{20} + w_{41} I_{41} \\ + w_{45} I_{45} + 2f_{80}(2I_1 + 2I_{50} I_{21})) \end{aligned} \tag{12}$$

$$\begin{aligned} f_{80} = logsigmoid(w_{1A} I_1 + w_{1B} I_1 \\ + w_{50A} I_{50} + w_{50B} I_{50} + w_{21} I_{21}) \end{aligned} \tag{13}$$

The input I_{50} is the feedback input which is given by Eq. 14

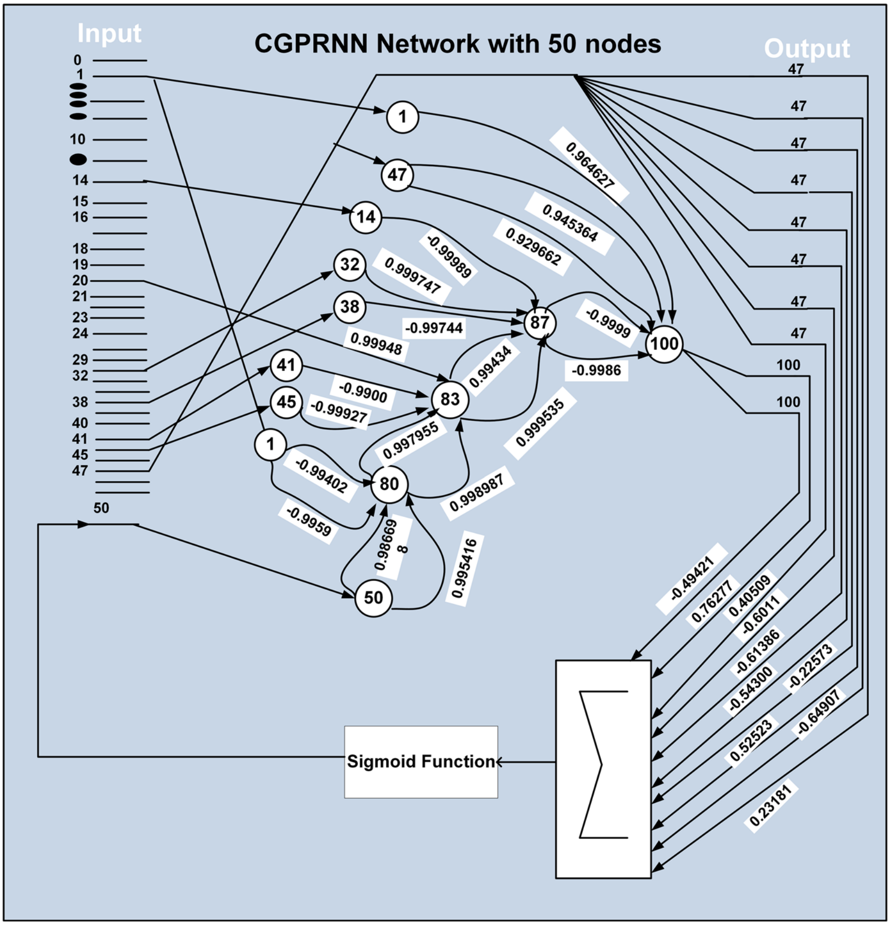


Fig. 7 A CGPRNN Network with 50 nodes and 24 feedback paths which predicts the next half an hour for a month based on past 24 h

$$I_{50} = \sum_{i=1}^{10} w_{Oi} O_i \tag{14}$$

6 Conclusion

We have presented the newly introduce neuro-evolutionary algorithms: Cartesian genetic programming evolved recurrent neural networks (CGPRNN) and explored its capabilities to produce an accurate prediction model for highly dynamic load patterns of Electric load over very short period of time (half an hour). The network is trained on half hourly electrical load data to produce the prediction models that can forecast the load of next half hour for an entire month using historical data of past 24 h. The experimental results demonstrated pre-eminence of the algorithms over the machine learning algorithms introduced to date, especially in the case of forecasting the electric load on very short term bases obtaining an accuracy of as high as 98.95 %. These prediction models are also tested for their robustness in scenarios other than the one they are trained on, to predict more future instances of load patterns in advance, obtaining up to 94 % accurate results. Thus proving the models to be not only accurate but also robust and can be used to predict the patterns in time series data in general. The algorithm seems to have great potential in the field of forecasting and can be used to forecast the wind speed, population growth, foreign currency exchange rates, and weather forecasting.

References

1. G.M. Khan, R. Arshad, S.A. Mahmud, F. Ullah, Intelligent bandwidth estimation for variable bit rate traffic. IEEE Trans. Evol. Comput. **19**(1), 151–155 (2015)
2. C. Kadilar, M. Simsek, C.H. Aladag, Forecasting the exchange rate series with ann: the case of Turkey. Istanb. Univ. Economet. Stat J. **9**(1), 17–29 (2009)
3. E. El-Attar, J. Goulermas, Q. Wu, Forecasting electric daily peak load based on local prediction, in *Power & Energy Society General Meeting, 2009 (PES'09)* (IEEE, 2009), pp. 1–6
4. G.M. Khan, F. Zafari, S.A. Mahmud, Very short term load forecasting using Cartesian genetic programming evolved recurrent neural networks (cgprnn), in *12th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, (IEEE, 2013), pp. 152–155
5. M.M. Khan, G.M. Khan, J.F. Miller, Evolution of optimal ANNs for non-linear control problems using Cartesian genetic programming, in *Proceedings of the 2010 International Conference on Artificial Intelligence Intelligence (IC-AI 2010)*, July 12–15, 2010, Las Vegas, NV, pp. 339–346
6. F. Zhao, H. Su, Short-term load forecasting using Kalman filter and elman neural network, in *2nd IEEE Conference on Industrial Electronics and Applications (ICIEA)* (IEEE, 2007), pp. 1043–1047
7. H. Al-Hamadi, S. Soliman, Fuzzy short-term electric load forecasting using Kalman filter. IEE Proc. Gener. Transm. Distrib. **153**(2), 217–227 (2006)
8. J.-H. Lim, O.-S. Kwon, K.-B. Song, J.-D. Park, Short-term load forecasting for educational buildings with temperature correlation, in *Fourth International Conference on Power Engineering, Energy and Electrical Drives (POWERENG)* (IEEE, 2013), pp. 405–408
9. S. Ramos, J. Soares, Z. Vale, Short-term load forecasting based on load profiling, in *Power and Energy Society General Meeting (PES)* (IEEE, 2013), pp. 1–5
10. B.-J. Chen, M.-W. Chang, C.-J. Lin, Load forecasting using support vector machines: a study on eunite competition 2001. IEEE Trans. Power Syst. **19**(4), 1821–1830 (2004)
11. P.-F. Pai, W.-C. Hong, Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. Electr. Power Syst. Res. **74**(3), 417–425 (2005)
12. T. Hong, J. Wilson, J. Xie, Long term probabilistic load forecasting and normalization with hourly information. IEEE Trans. Smart Grid **5**(1), 456–462 (2014)
13. X. Chen, C. Kang, X. Tong, Q. Xia, J. Yang, Improving the accuracy of bus load forecasting by a two-stage bad data identification method. IEEE Trans. Power Syst. **29**(4), 1634–1641 (2014)
14. R.-A. Hooshmand, H. Amooshahi, M. Parastegari, A hybrid intelligent algorithm based short-term load forecasting approach. Int. J. Electr. Power Energy Syst. **45**(1), 313–324 (2013)
15. P. Mandal, T. Senjyu, N. Urasaki, T. Funabashi, A neural network based several-hour-ahead electric load forecasting using similar days approach. Int. J. Electr. Power Energy Syst. **28**(6), 367–373 (2006)
16. A.K. Pandey, K.B. Sahay, M. Tripathi, D. Chandra, Short-term load forecasting of uppcl using ann, in *6th IEEE Power India International Conference (PIICON)* (IEEE, 2014), pp. 1–6
17. K.B. Sahay, N. Kumar, M. Tripathi, Short-term load forecasting of ontario electricity market by considering the effect of temperature, in *6th IEEE Power India International Conference (PIICON)* (IEEE, 2014), pp. 1–6
18. X. Yao, Evolving artificial neural networks. Proc. IEEE **87**(9), 1423–1447 (1999)

19. F. Gomez, J. Schmidhuber, R. Miikkulainen, Accelerated neural evolution through cooperatively coevolved synapses. J. Mach. Learn. Res. **9**, 937–965 (2008)
20. K.O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies. Evol. Comput. **10**(2), 99–127 (2002)
21. Z. Vašíček, L. Sekanina, Hardware accelerator of Cartesian genetic programming with multiple fitness units. Comput. Inform. **29**(6), 1359–1371 (2012)
22. J.A. Rothermich, J.F. Miller, Studying the emergence of multicellularity with Cartesian genetic programming in artificial life, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Late Breaking Papers* (Morgan Kaufmann Publishers, 2002), pp. 397–403
23. M. Akole, B. Tyagi, Artificial neural network based short term load forecasting for restructured power system, in *International Conference on Power Systems, 2009 (ICPS'09)* (IEEE, 2009), pp. 1–7
24. S. Fan, R.J. Hyndman, Short-term load forecasting based on a semi-parametric additive model. IEEE Trans. Power Syst. **27**(1), 134–141 (2012)