Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 54 (2008) 903–917
computers & industrial engineering
www.elsevier.com/locate/caie

Univariate modeling and forecasting of monthly energy demand time series using abductive and neural networks

R.E. Abdel-Aal *

Computer Engineering Department, King Fahd University of Petroleum and Minerals, P.O. Box 1759, Dhahran 31261, Saudi Arabia

Received 23 July 2007; accepted 29 October 2007
Available online 6 November 2007

Abstract

Neural networks have been widely used for short-term, and to a lesser degree medium and long-term, demand forecasting. In the majority of cases for the latter two applications, multivariate modeling was adopted, where the demand time series is related to other weather, socio-economic and demographic time series. Disadvantages of this approach include the fact that influential exogenous factors are difficult to determine, and accurate data for them may not be readily available. This paper uses univariate modeling of the monthly demand time series based only on data for 6 years to forecast the demand for the seventh year. Both neural and abductive networks were used for modeling, and their performance was compared. A simple technique is described for removing the upward growth trend prior to modeling the demand time series to avoid problems associated with extrapolating beyond the data range used for training. Two modeling approaches were investigated and compared: iteratively using a single next-month forecaster, and employing 12 dedicated models to forecast the 12 individual months directly. Results indicate better performance by the first approach, with mean percentage error (MAPE) of the order of 3% for abductive networks. Performance is superior to naïve forecasts based on persistence and seasonality, and is better than results quoted in the literature for several similar applications using multivariate abductive modeling, multiple regression, and univariate ARIMA analysis. Automatic selection of only the most relevant model inputs by the abductive learning algorithm provides better insight into the modeled process and allows constructing simpler neural network models with reduced data dimensionality and improved forecasting performance.

Keywords: Abductive networks; Neural networks; Energy demand; Time series forecasting; Medium-term load forecasting; Univariate time series analysis

1. Introduction

Load forecasting plays an important role in the planning and economic and secure operation of modern power systems. Long-term forecasts (5–20 years) are required for resource planning, utility

* Tel.: +966 3 860 4320; fax: +966 3 860 3059.
E-mail address: radwan@kfupm.edu.sa

0360-8352/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2007.10.020

expansion, and staff hiring. Medium-term forecasts (1 month to 5 years) are used for purchasing fuel and revising electricity tariffs. Short-term load forecasting (STLF) (1 h to 1 week) is important for scheduling functions, such as generator unit commitment, hydro-thermal coordination, short-term maintenance, fuel allocation, power interchange, transaction evaluation, as well as network analysis functions, such as dispatcher power flow and optimal power flow. This paper is concerned with medium-term energy demand forecasting. Demand for electricity changes with time, weather, socio-economic variables, and population demographics. Techniques for demand forecasting in the medium to long range include the use of multivariate causal (econometric) models (Liu, Ang, & Goh, 1991) and multiple regression models (Barakat & Eissa, 1989) to express the load as a function of exogenous inputs such as weather and social/economic variables. In addition to the complexity of the modeling process, regression models are basically linear devices that attempt to model distinctly nonlinear relationships (Hippert, Pedreira, & Souza, 2001). Even when a nonlinear relationship is attempted for the model, it is difficult to determine empirically the correct complex relationship that actually exists between demand and the other explanatory inputs.

Univariate analysis of the time series alone was proposed for short-term load forecasting (Toyada, Chen, & Inouse, 1970), but forecasting accuracy has been limited due to ignoring important exogenous factors such as weather and day type. However, with medium and long-term forecasting, integration of daily loads into monthly or yearly data dilutes the effects of such factors, and forecasting may be based solely on previous values of the time series (Gonzalez-Romera, Jaramillo-Moran, & Carmona-Fernandez, 2007). As all the external factors have already impacted the generation of the observed time series, it can be argued that the time series embodies all necessary information to model the underlying generating process. Univariate time series models have proved more accurate than multivariate abductive network models in forecasting the monthly energy consumption in the Eastern Province of Saudi Arabia (Abdel-Aal, Al-Garni, & Al-Nassar, 1997). External factors often require economic and demographic data that may be unavailable or difficult to obtain. To use the model for forecasting into the future, forecasts for such factors may be required, and the forecasting accuracy would depend on the quality of such forecasts. Moreover, using only the demand time series can reduce the data dimensionality for the problem being modeled, which improves generalization and forecasting performance. Conventional univariate time series analysis approaches include the Box-Jenkins integrated autoregressive moving average (ARIMA) method (Abdel-Aal & Al-Garni, 1997), logistic modeling (Barakat & Alrashed, 1992), and Kalman filtering (Al-Hamadi & Soliman, 2006). However, such methods are time consuming, require extensive user intervention, and may be numerically unstable (AlFuhaid, El-Sayed, & Mahmoud, 1997).

A recent trend for handling problems that are difficult to solve analytically has been to resort to artificial intelligence and machine-learning techniques, which range from knowledge-based expert systems to data-based approaches such as fuzzy logic, genetic algorithms, and neural networks. Data-based techniques are non-parametric, i.e., the user does not need to explicitly specify the model relationship. This enhances their use in knowledge discovery by freeing them from bias or influence by prior assumptions on the modeled relationship, which are required in many empirical and regression modeling approaches. Complex nonlinear input–output relationships can be modeled automatically through supervised learning using a database of solved examples. Once synthesized, the model can generalize to perform predictions of outputs corresponding to new cases, previously unseen during training. This offers a number of advantages over conventional approaches, including reduced development time and increased tolerance to noise, uncertainty, and missing data, reduced need for theoretical knowledge on the modeled phenomenon, freedom from assumptions on the probability distributions of input variables, and the fact that no programming is required. Intensive computations are needed only once during model synthesis, while predictions by the models synthesized are fast and straightforward. Neural networks have been used extensively for STLF, (e.g., Khotanzad, Afkhami-Rohani, & Maratukulam, 1998; Shimakura et al., 1993), and to a lesser extent for medium-term (Elkateb, Solaiman, & Al-Turki, 1998; Ghiassi, Zimbra, & Saidane, 2006; Gonzalez-Romera et al., 2007; Islam, Al-Alawi, & Ellithy, 1995) and long long-term load forecasting (Al-Saba & El-Amin, 1999; Kermanshahi & Iwamiya, 2002). In the majority of cases, multivariate modeling is employed using a number of time series of explanatory variables that influence the load. For long-term

forecasting, such factors include population, number of households, number of clients, number of air conditioning units sold/installed, and amount of CO_2 pollution in the environment, as well as economic indicators, e.g., gross national product (GNP), gross domestic product (GDP), index of industrial production (IIP), real per capita income, and real personal income, electricity price, crude oil price. For medium-term forecasting, explanatory factors included weather parameters, e.g., temperature, relative humidity, heating degree-days (HDD), and cooling degree-days (CDD). Univariate modeling of the load time series with neural networks based solely on the load data without using any weather data was used only in a few cases, (e.g., Gonzalez-Romera et al., 2007).

In spite of the wide-spread use of neural networks, the technique suffers from a number of limitations, including difficulty in determining optimum network topology and training parameters (Alves Da Silva, Rodrigues, Rocha Reis, & Moulin, 2001). There are many choices to be made in determining numerous critical design parameters with little guidance available and designers often resort to a trial and error approach which can be tedious and time consuming. Such design parameters include the number and size of the hidden layers, the type of neuron transfer functions for the various layers, the training rate and momentum coefficient, and training stopping criteria to avoid over-fitting and ensure adequate generalization with new data. Another limitation is the black box nature of neural network models, giving poor explanation facilities and providing little insight into the modeled relationship and the relative significance of various inputs (Matsui, Iizaka, & Fukuyama, 2001). The acceptability of, and confidence in, automated load forecasting tools in an operational environment are related to their transparency and their ability to justify results to human experts (Lewis III, 2001).

Self-organizing abductive or polynomial networks (Montgomery & Drake, 1991) based on the group method of data handling (GMDH) learning algorithm (Farlow, 1984) have been proposed as an alternative modeling approach that alleviates many of the above limitations of neural networks. The method offers the advantages of improved forecasting performance (Abdel-Aal, 2004), faster model development requiring little or no user intervention, faster convergence during model synthesis without the problems of getting stuck in local minima, automatic selection of relevant input variables, and automatic configuration of model structures (Alves Da Silva et al., 2001). With the resulting model represented as a hierarchy of polynomial expressions, analytical model relationships can be derived. Such relationships provide insight into the modeled phenomena, highlight contributions of various inputs, and allow comparison with previously used empirical or statistical models. The technique automatically avoids over-fitting by using a proven regularization criterion based on penalizing model complexity without requiring a dedicated validation data set during training, as is the case with many neural network paradigms. Various forms of GMDH-based networks were used in multivariate modeling for short-term (Abdel-Aal, 2004; Alves Da Silva et al., 2001; Dillon, Morsztyn, & Phula, 1975; Sforna, 1995) and medium-term (Abdel-Aal et al., 1997) load forecasting. No applications could be traced in the literature for GMDH-based univariate time series load forecasting, although the technique has been used for univariate processing of time series in other areas, including modeling discharge currents accompanying tree development in dielectrics (Fujimori, Kajiya, & Mitsuboshi, 1998) and forecasting bilinear time series (Bielinska & Nabaglo, 1994).

This paper compares the performance of abductive networks and neural networks on forecasting the monthly energy demand for 1year through univariate modeling of the demand time series for the previous 6 years. The upward trend in the demand time series is removed prior to modeling so that trained networks do not have to extrapolate beyond the range of data experienced during training, which improves forecasting performance (Virili & Freisleben, 2000). Two approaches were adopted with each of the two learning paradigms: iterative use of a single model that forecasts the demand for next month, and direct application of twelve dedicated models, each forecasting the demand for a month in the evaluation year. We demonstrate improved performance of neural network models when trained on a reduced subset of input features selected automatically by the abductive network learning algorithm. Following a short description of the abductive and neural network learning algorithms in Section 2, the demand data set used and the technique adopted for handling the time series upward trend are described in Section 3, together with the approaches for two baseline naïve (flat) forecasts. Section 4 describes the development and use of next-month demand forecasters employing both neural and abductive networks.

Section 5 covers dedicated multiple models that forecast the full evaluation year in one go. Concluding remarks are given in Section 6.

2. Methods

2.1. GMDH and AIM abductive networks

AIM (abductory inductive mechanism) (AbTech Corporation, 1990) is a supervised inductive machine-learning tool for automatically synthesizing abductive network models from a database of inputs and outputs representing a training set of solved examples. As a GMDH algorithm, the tool can automatically synthesize adequate models that embody the inherent structure of complex and highly nonlinear systems. The automation of model synthesis not only lessens the burden on the analyst but also safeguards the model generated from being influenced by human biases and misjudgments. The GMDH approach is a formalized paradigm for iterated (multi-phase) polynomial regression capable of producing a high-degree polynomial model in effective predictors. The process is 'evolutionary' in nature, using initially simple (myopic) regression relationships to derive more accurate representations in the next iteration. To prevent exponential growth and limit model complexity, the algorithm selects only relationships having good predicting powers within each phase. Iteration is stopped when the new generation regression equations start to have poorer prediction performance than those of the previous generation, at which point the model starts to become overspecialized and therefore unlikely to perform well with new data. The algorithm has three main elements: representation, selection, and stopping. It applies abduction heuristics for making decisions concerning some or all of these three aspects.

To illustrate these steps for the classical GMDH approach, consider an estimation database of n_e observations (rows) and $m+1$ columns for m independent variables $(x_1, x_2, \ldots, x_m)$ and one dependent variable y. In the first iteration we assume that our predictors are the actual input variables. The initial rough prediction equations are derived by taking each pair of input variables $(x_i, x_j; i, j = 1, 2, \ldots, m)$ together with the output y and computing the quadratic regression polynomial (Farlow, 1984):

$$y = A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_ix_j. \tag{1}$$

Each of the resulting $m(m-1)/2$ polynomials is evaluated using data for the pair of x variables used to generate it, thus producing new estimation variables $(z_1, z_2, \ldots, z_{m(m-1)/2})$ which would be expected to describe y better than the original variables. The resulting z variables are screened according to some selection criterion and only those having good predicting power are kept. The original GMDH algorithm employs an additional and independent selection set of n_s observations for this purpose and uses the regularity selection criterion based on the root mean squared error r_k over that data set, where

$$r_k^2 = \sum_{\ell=1}^{n_s}(y_\ell - z_{k\ell})^2 \Bigg/ \sum_{\ell=1}^{n_s} y_\ell^2; \quad k = 1, 2, \ldots, m(m-1)/2. \tag{2}$$

Only those polynomials (and associated z variables) that have r_k below a prescribed limit are kept and the minimum value, $r_{\min}$, obtained for r_k is also saved. The selected z variables represent a new database for repeating the estimation and selection steps in the next iteration to derive a set of higher-level variables. At each iteration, $r_{\min}$ is compared with its previous value and the process is continued as long as $r_{\min}$ decreases or until a given complexity is reached. An increasing $r_{\min}$ is an indication of the model becoming overly complex, thus over-fitting the estimation data and performing poorly in predicting the new selection data. Keeping model complexity checked is an important aspect of GMDH-based algorithms, which keep an eye on the final objective of constructing the model, i.e., using it with new data previously unseen during training. The best model for this purpose is that providing the shortest description for the data available (Barron, 1984). Computationally, the resulting GMDH model can be seen as a layered network of partial quadratic descriptor polynomials, each layer representing the results of an iteration.

A number of GMDH methods have been proposed which operate on the whole training data set thus avoiding the use of a dedicated selection set. The adaptive learning network (ALN) approach, AIM being

an example, uses the predicted squared error (PSE) criterion (Barron, 1984) for selection and stopping to avoid model over-fitting, thus eliminating the problem of determining when to stop training in neural networks. The criterion minimizes the expected squared error that would be obtained when the network is used for predicting new data. AIM expresses the PSE error as:

$$\mathrm{PSE} = \mathrm{FSE} + \mathrm{CPM}(2K/n)\sigma_p^2, \tag{3}$$

where FSE is the fitting squared error on the training data, CPM is a complexity penalty multiplier selected by the user, K is the number of model coefficients, n is the number of samples in the training set, and σ_p^2 is a prior estimate for the variance of the error obtained with the unknown model. This estimate does not depend on the model being evaluated and is usually taken as half the variance of the dependent variable y (Barron, 1984). As the model becomes more complex relative to the size of the training set, the second term increases linearly while the first term decreases. PSE goes through a minimum at the optimum model size that strikes a balance between accuracy and simplicity (exactness and generality). The user may optionally control this trade-off using the CPM parameter. Larger values than the default value of 1 lead to simpler models that are less accurate but may generalize well with previously unseen data, while lower values produce more complex networks that may over-fit the training data and degrade actual prediction performance.

AIM builds networks consisting of various types of polynomial functional elements. The network size, element types, connectivity, and coefficients for the optimum model are automatically determined using well-proven optimization criteria, thus reducing the need for user intervention compared to neural networks. This simplifies model development and reduces the learning/development time and effort. The models take the form of layered feed-forward abductive networks of functional elements (nodes) (AbTech Corporation, 1990), see Fig. 1. Elements in the first layer operate on various combinations of the independent input variables (x's) and the element in the final layer produces the predicted output for the dependent variable y. In addition to the main layers of the network, an input layer of normalizers convert the input variables into an internal representation as Z scores with zero mean and unity variance, and an output unitizer unit restores the results to the original problem space. The used version of AIM supports the following main functional elements:

(i) A white element which consists of a constant plus the linear weighted sum of all outputs of the previous layer, i.e.,:

$$\text{“White” Output} = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n, \tag{4}$$

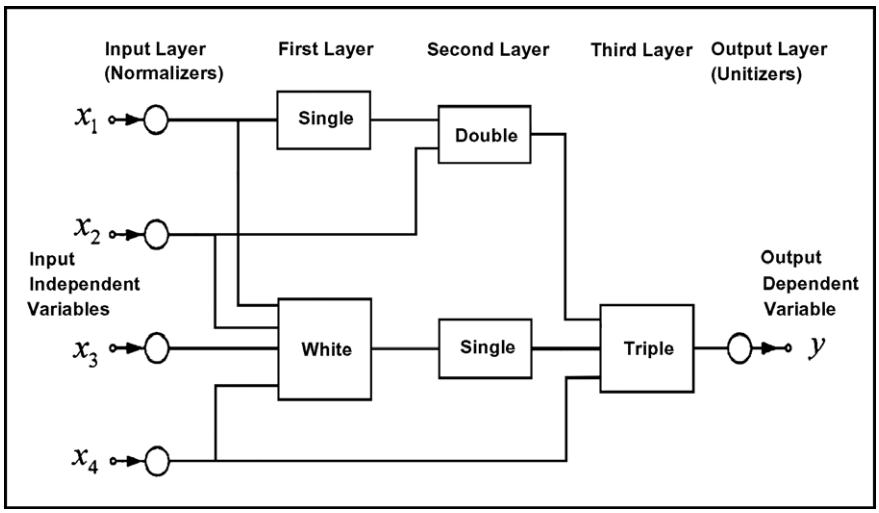


Fig. 1. A typical AIM abductive network model showing various types of functional elements.

where $x_1, x_2, \ldots, x_n$ are the inputs to the element and $w_0, w_1, \ldots, w_n$ are the element weights.

(ii) Single, double, and triple elements which implement a third-degree polynomial expression with all possible cross-terms for one, two, and three inputs, respectively; for example,

$$\text{“Double” Output} = w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + w_6x_1^3 + w_7x_2^3. \tag{5}$$

2.2. Multilayer perceptron (MPL), back propagation, neural networks

The MLP neural network consists of simple processing elements (artificial neurons) arranged in layers: an input layer receiving the input variables, one or more hidden layers performing the required nonlinear input–output mappings, and an output layer producing the network outputs. Each neuron receives weighted inputs from all neurons in the preceding layer. Let W_{ij} be the weight associated with the link from neuron i in one layer to neuron j in the next downstream layer. The neuron sums all weighted inputs and, with reference to a threshold value, uses a nonlinear activation function to determine its output. The modeling problem is solved by training on a set of solved examples in the form of input–output records. Training attempts to minimize the error between known and calculated network outputs over all training examples through optimizing the network weights. The mean square error (MSE) criterion is given by:

$$E = \frac{1}{2}\left[\sum_p \sum_k |t_{kp} - O_{kp}|^2\right], \tag{6}$$

where t_{kp} and O_{kp} are the true and observed outputs, respectively, for neuron k in the output layer when input vector $\mathbf{x}_p$ corresponding to the pth training record is applied to the network. Training with the back propagation algorithm involves iterative application of the training records, determining observed output errors for neurons in the output layer, back propagating these errors to all previous layers, and adjusting the weights so as to minimize the error criterion. The output from neuron j in a given layer (other than the input layer) is calculated as:

$$O_j = f\left(\sum_i W_{ij}O_i\right), \tag{7}$$

where i indicates a neuron in the preceding layer and f is the activation function for neuron j. The activation function is often a sigmoid function of the form:

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{8}$$

With the gradient descent approach to error minimization, weights are changed in proportional to the error gradient, i.e.,

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}}, \tag{9}$$

where η is a constant that determines the learning rate. To improve convergence characteristics, weight changes are also related to changes introduced in the previous iteration. At the nth iteration, the change in W_{ij} for the link from neuron i to neuron j is given by (Park, El-Sharkawi, Marks, Atlas, & Damborg, 1991):

$$\Delta W_{ij}(n) = \varepsilon\delta_j O_i + \alpha\Delta W_{ij}(n-1), \tag{10}$$

where ε is the learning rate, α is the momentum factor, and δ_j is the error signal for the destination neuron j. When neuron j is in the output layer, δ_j is given by:

$$\delta_j = (t_j - O_j)O_j(1 - O_j). \tag{11}$$

When neuron j is in a hidden layer, δ_j is given by:

$$\delta_j = O_j(1 - O_j)\sum_k \delta_k W_{jk}, \tag{12}$$

where k indicates neurons in the succeeding layer next to that containing neuron j.

The learning rate and the momentum factor influence the speed and stability of network training. The process continues until the error criterion on the training set is reduced below a specified limit. To improve generalization on new out-of-sample data, early stopping criteria are often employed where a separate test dataset is used to validate the resulting model and training is stopped when error on that dataset starts to increase indicating the start of over-fitting.

3. Material

3.1. The data set

The original data set used consists of measured hourly load and temperature data for the Puget power utility, Seattle, USA, over the period 1 January 1985 to 12 October 1992. The data set was provided by Professor A.M. El-Sharkawi, University of Washington, Seattle, USA. A few missing load and temperature data, indicated as 0's in the original data set, were filled-in by interpolating between neighboring values. Monthly energy demand data were calculated by integrating the hourly loads for each month. We used the data for the first six years (1985–1990) for model synthesis and that for the following year (1991) for model evaluation. Table 1 summarizes the load data for the 7-year period and indicates an average year-to-year annual growth rate of 3.25%. The 7-year monthly demand time series raw data in tera watt hours (TWH) is plotted in Fig. 2(a). The plot clearly shows a 12-month seasonality and a significant upward trend. With such a trend, data values for the last (evaluation) year in the time series is expected to exceed those for the preceding years used for training. The abductive and neural network modeling techniques employed are good interpolators, but perform poorly in extrapolating beyond the data range encountered during training. Gonzalez-Romera et al. (2007) have solved this problem through splitting the load time series into a trend time series and a fluctuation time series which were modeled and forecasted separately and the results combined to produce the final load forecast. To avoid the complexity of extracting the trend through smoothing and avoid duplicating the modeling and forecasting effort, we adopt a simpler approach in this paper. Here, all monthly demand data were first normalized so that all years have an annual energy demand equal to that of the last training year (1990). Let the annual energy demand for year i be E_i, the normalization factor f_i for that year is defined as:

$$f_i = E_{1990}/E_i; \quad i = 1985, 1986, \ldots, 1991. \tag{13}$$

For the evaluation year 1991, we use an estimated value for the annual energy demand, $\widehat{E}_{1991}$, because in practice no actual data would be available for the evaluation year. This estimated value was obtained by extrapolating a straight line fit for the annual energy demands over the preceding 6 years (1985–1990).

Table 1
Summary of the 7-year energy demand data showing year-to-year growth and the normalization factors used for handling trend

Year, i	Total energy demand, E_i (MWH)	Annual growth (year-to-year)	Normalization factors, f_i
1985	16,310,645	1	1.187
1986	16,017,335	0.982	1.209
1987	16,510,405	1.031	1.172
1988	17,563,434	1.061	1.102
1989	18,434,815	1.052	1.050
1990	19,357,130	1.050	1
1991			
Actual	19,523,246	1.009	0.991
Estimated	19,719,570	1.019	0.982
Average annual growth 1986–1991 (Actual)		1.0325	

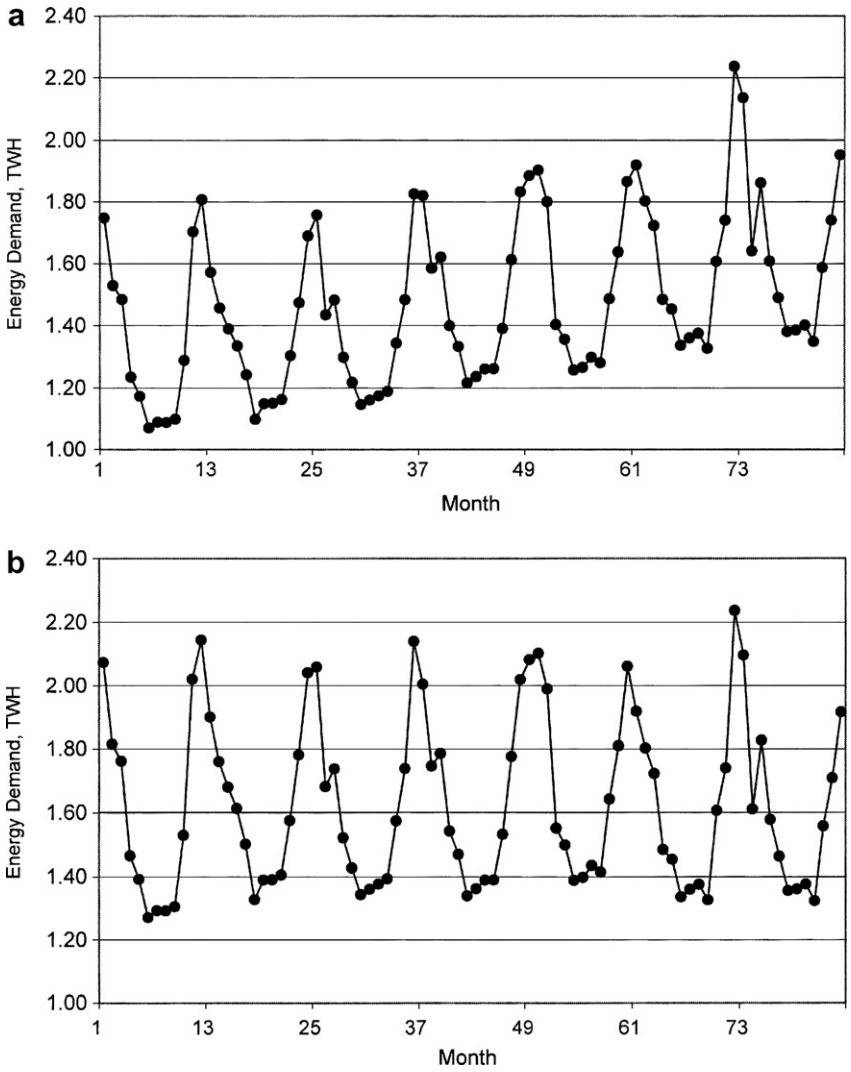


Fig. 2. The 7-year monthly energy demand time series: (a) raw data showing upward trend, (b) data normalized for trend removal.

Normalization factors for the 7 years are listed in the last column of Table 1. Monthly demands were normalized by multiplying actual values for each year by the corresponding normalization factor. Fig. 2(b) shows the trend-normalized time series that was used to train and evaluate all forecasting models throughout this paper. Forecasted values for the evaluation year 1991 were first de-normalized through dividing by f_{1991} before comparison with actual 1991 monthly demand data.

3.2. Naïve forecasts

In order to justify the forecasting effort, results are often compared with simple (naïve) forecasts obtained easily from the available data. One naïve forecast (Naïve-I) often used in practice (Thiesing & Vornberger, 1997) is based on persistence, and assumes the demand at each month to be equal to the actual demand recorded on the preceding month. Using the monthly demand data described above, persistence-based forecasts give a maximum and mean absolute percentage errors of 20% and 7.1%, respectively. In view of the strong seasonality of the time series, another possible naïve forecast (Naïve-II) would assume the demand at each month in the 1991 forecasting year as equal to the demand in the same month of the previous year

(1990) divided by the estimated normalization factor f_{1991}, see Table 1. This forecasting method proved superior to persistence forecasting, with the maximum and mean absolute percentage errors being 16.67% and 4.68%, respectively.

4. Iterative forecasting using a single next-month model

We have developed abductive and neural network models that forecast the monthly energy demand for 1 year ahead from the monthly energy demand time series for 6 years. In one approach to the problem, a single model is developed to forecast the demand one month ahead using a number of previous demand values as model inputs. The model is then iteratively used twelve times to forecast the demand for a full year ahead. The value forecasted at a given iteration is fed as a model input in the next iteration. The optimum size of the sliding time window containing the time series samples used as model inputs is an important parameter referred to as the embedding dimension of the time series (Frank, Davey, & Hunt, 2000). Unduly small time windows may not allow the model to capture enough information on the underlying system generating the series. Unnecessarily large time windows increase the dimensionality of the data and reduce the number of records available for training. Both factors contribute to over-fitting by the synthesized model, leading to poor generalization during forecasting. A number of methods have been proposed for finding the correct embedding dimension for a given time series (Frank et al., 2000). For seasonal time series, it is important to include samples at time lags equal to the seasonality interval (Chatfield, 1998). The annual seasonality that is evident from the time series plots in Fig. 2 suggests that a window size of 12 samples should be adequate. For a model that forecasts the demand on the next month $D(m+1)$ from demand at the previous 12 months, a record in the training/evaluation datasets takes the following form:

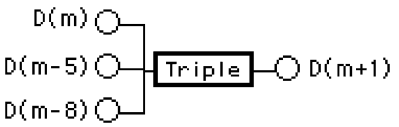
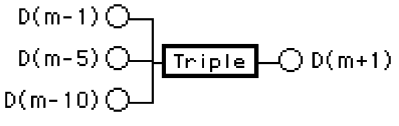
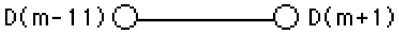
Inputs	Output
$D(m-11),\ldots,D(m-2),D(m-1),D(m)$	$D(m+1)$

This provides 60 training examples from the 72 monthly demand data for the six training years.

4.1. Abductive networks modeling

Table 2 shows the structure and performance results for the abductive network models synthesized for the next-month demand $D(m+1)$ at three levels of model complexity, including the default CPM value of 1, a

Table 2
Structure and performance results for the abductive network models synthesized for forecasting next-month demand at three levels of model complexity

CPM	Model synthesized		Absolute forecasting error over the evaluation year (%)	
	Structure	Time lags selected	Maximum	Mean
1		1, 6, 9	13.80	3.52
0.5		2, 6, 11	7.93	3.19
2		12	11.44	4.25
Naïve forecasts	Naïve-I	1	20	7.10
	Naïve-II	12	16.67	4.68

Performance of the two naïve forecasts is included for comparison.

more complex model at CPM = 0.5, and a simpler model at CPM = 2. The first two models are nonlinear functions in only three inputs selected automatically by the learning algorithm out of the twelve inputs available during training. The selected inputs roughly represent the beginning, the middle, and the end of the 12 months interval preceding the modeled month. Table 2 shows also the time lags corresponding to the selected inputs. Selection of time lags as long as 12 months by some models shows that the size chosen for the sliding time window to model the time series is not larger than necessary. The simplest model obtained with CPM = 2 is a linear model consisting of a 'wire' element connecting the normalizer element of the only input selected to the unitizer element of the output. Substituting symbolically for the equations of both elements and performing trend de-normalization gives the following simple relationship for the forecasted monthly demand:

$$D(m+1) = 0.070130 + 0.975281D(m-11). \tag{14}$$

This 'wire' model allows prediction of the monthly demand in the forecasting year using only the demand at the same month in the previous year, and is therefore similar to the seasonality-based Naïve-II model described in Section 3.2.

Each of the three models was used iteratively for twelve times to forecast the monthly demand over the evaluation year 1991. In each iteration, the forecasted demand is made available as input to the model in the next iteration, which could lead to accumulation of the forecasting error. It is noted that the third model at CPM = 2 uses a time lag of 12, and when forecasting only up to 12 months ahead, none of the forecasted values will be required by the model. Table 2 shows the maximum and mean values for the absolute percentage error. The model with CPM = 0.5 gives the best performance, with a MAPE error of 3.19% for next year forecast, which is significantly superior to the two approaches for naïve forecasts. The 'wire' abductive model synthesized with CPM = 2 also outperforms the similarly structured Naïve-II model. Fig. 3(a) is a plot comparing actual and forecasted demands obtained with this model over the evaluation year. For the similar univariate approach applied to the Spanish electric consumption data, Gonzalez-Romera et al. (2007) report a MAPE error of 2% only for next month predictions using 23 years data for training and 5 years for evaluation and applying complex techniques for trend extraction and modeling. Monthly forecasts up to a year ahead was not reported in that study. In another application, data for 5 years on the electric energy consumption in the Eastern Province of Saudi Arabia was used to develop forecasting models for the monthly demand for the 6th year. A multivariate regression model that used the additional time series of temperature, humidity, global solar radiation, and population gave a MAPE error of 9.2% (Al-Garni, Zubair, & Nizami, 1994). An abductive network multivariate model using a monthly index and the additional time series of temperature and humidity gave a MAPE value of 5.6% (Abdel-Aal et al., 1997), while a univariate Box-Jenkins ARIMA model for the same energy consumption data gave a MAPE value of 3.8% (Abdel-Aal & Al-Garni, 1997).

4.2. Neural networks modeling

Multilayer perceptron neural networks trained using the back propagation learning algorithm were used to develop models for forecasting next-month energy demand using the PathFinder software for Windows. The models were trained and evaluated on the same data used for the abductive models, with 20% of the training data reserved for cross validation. All twelve inputs were used for network training and default values for network design and training parameters were employed. The sigmoid transfer function was used for neurons in both the hidden and the output layer. Three values (4, 5, and 6) for the number of neurons in the single hidden layer were tried in order to investigate the effects of model complexity on forecasting performance. Table 3 shows the maximum and mean absolute percentage forecasting errors over the evaluation year for the various models developed. Results indicate that the model with medium complexity (5 hidden neurons) gives best average forecasting accuracy, with a MAPE value of 3.77. This neural network model is inferior to the best abductive network model with CPM = 0.5 (Table 2) which uses only three out of the twelve inputs. The last neural network model in Table 3 is a model trained on only the three inputs ($D(m-1)$, $D(m-5)$, and $D(m-10)$) selected by that abductive network model as an optimum subset of inputs. With a MAPE of 3.09, this simple neural model (3–5–1) gives the best forecasting performance so

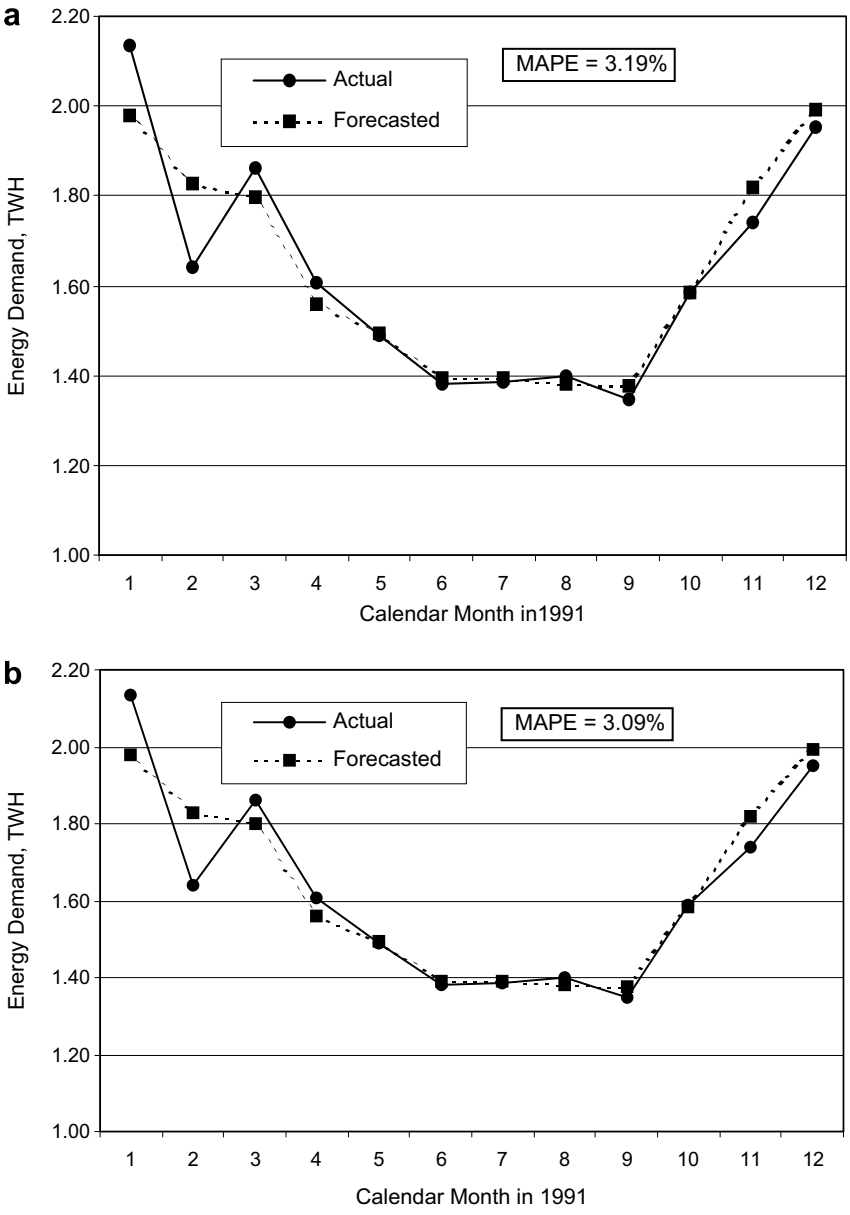


Fig. 3. Plots of actual and forecasted demand over the evaluation year with iterated application of a single next-month forecaster. (a) Using optimum abductive network model with CPM = 0.5, (b) using a 3–5–1 neural network model employing only the three model inputs automatically selected by the abductive model of (a).

far, and achieves 18% reduction in the MAPE value for the more complex neural model (12–5–1) that uses all twelve inputs. A smaller subset of inputs consisting of good predictors helps reduce the high data dimensionality problem which causes the model to over-fit the training data and degrades model performance on out-of-sample data during actual forecasting. This effect becomes particularly important with the relatively small value for the ratio between the number of training records available and the number of model inputs. With 48 records used for actual training of the neural network (the remaining 12 records being reserved for cross-validation), the 3–5–1 model raises this ratio from 4 to 16, which reduces over-fitting and improves forecasting performance.

Table 3
Structure and performance results for neural network models synthesized for forecasting next-month demand

Inputs used	Model structure	Absolute forecasting error over the evaluation year (%)	
		Maximum	Mean
All 12 inputs	12–4–1	11.38	4.06
	12–5–1	13.02	3.77
	12–6–1	12.55	4.07
3 inputs only: $D(m-1)$, $D(m-5)$, $D(m-10)$	3–5–1	11.52	3.09
Naïve Forecasts	Naïve-I	20	7.10
	Naïve-II	16.67	4.68

The 3-input neural model uses an optimum subset of inputs selected automatically for the abductive model with CPM = 0.5 in Table 2. Performance of the two naïve forecasts is included for comparison.

5. Direct forecasting using 12 dedicated models

In this forecasting approach, 12 dedicated models are developed to forecast the monthly demand over the full evaluation year directly in one go, with each model forecasting the demand for one month of the year. The models forecast the demand on month i, $D(m+i)$, $i = 1, 2, 3, \ldots, 12$, in terms of historical demand data available at month m. The forecasting lead times for the 12 models are 1, 2, 3, ..., 12 months, respectively. A record in the training dataset takes the form:

Inputs	Outputs
$D(m-11), \ldots, D(m-2), D(m-1), D(m)$	$D(m+1), D(m+2), D(m+3), \ldots, D(m+12)$

This provides 49 training examples from the 72 monthly demand data for the six training years for each of the 12 dedicated models. Compared to the iterative approach described in Section 4, this approach has the disadvantages of fewer training records and longer forecasting lead times, both of which degrade forecasting accuracy. In a practical setting, it would also be more difficult to develop and maintain twelve models as compared to a single model in the iterative approach. However, the approach has the advantage that forecasted values are not re-cycled for calculating new forecasts, and therefore forecasting errors do not accumulate.

5.1. Abductive networks modeling

Table 4 lists the inputs selected and the corresponding time lags for the twelve abductive network models synthesized for the ith month demand $D(m+i)$; $i = 1, 2, \ldots, 12$ at the default model complexity (CPM = 1). All models use a single element, which takes the form of a wire, double, or triple element for 1, 2, and 3 inputs,

Table 4
Summary of inputs selected and corresponding time lags for the 12 dedicated abductive network models that directly forecast the demand over the full evaluation year

Modeled output	Selected inputs	Corresponding time lags
$D(m+1)$	$D(m-11)$	12
$D(m+2)$	$D(m-10)$	12
$D(m+3)$	$D(m-1)$, $D(m-3)$	4, 6
$D(m+4)$	$D(m-1)$, $D(m-8)$, $D(m-9)$	5, 12, 13
$D(m+5)$	$D(m-1)$	6
$D(m+6)$	$D(m)$	6
$D(m+7)$	$D(m-5)$, $D(m-8)$	12, 15
$D(m+8)$	$D(m-1)$, $D(m-4)$	9, 12
$D(m+9)$	$D(m-3)$	12
$D(m+10)$	$D(m-2)$	12
$D(m+11)$	$D(m-4)$, $D(m-7)$	15, 18
$D(m+12)$	$D(m)$, $D(m-2)$	12, 14

respectively. The time lag of 12 months is used by eight of the 12 modules, indicating that the load at a given month is strongly influenced by the load at the same month of the preceding year, which is expected with the strong annual seasonality exhibited. Four models use the 6-month time lag or odd multiples of it. Fig. 4(a) is a plot comparing actual monthly demand with forecasted demand obtained using these twelve abductive network models over the evaluation year. The MAPE value of 4.84% is considerably inferior to that of 3.52% shown in Table 2 for the iterative approach using a single next-month abductive model at the same CPM value. This is attributed to the fewer training examples (49 versus 60) and longer forecast lead times (1–12 months versus 1 month, respectively).

5.2. Neural networks modeling

Twelve separate neural network models were synthesized for the ith month demand ($D(m+i)$); $i = 1, 2, \ldots, 12$ using all 12 inputs and with 6 neurons in the hidden layer. Fig. 4(b) is a plot comparing actual monthly

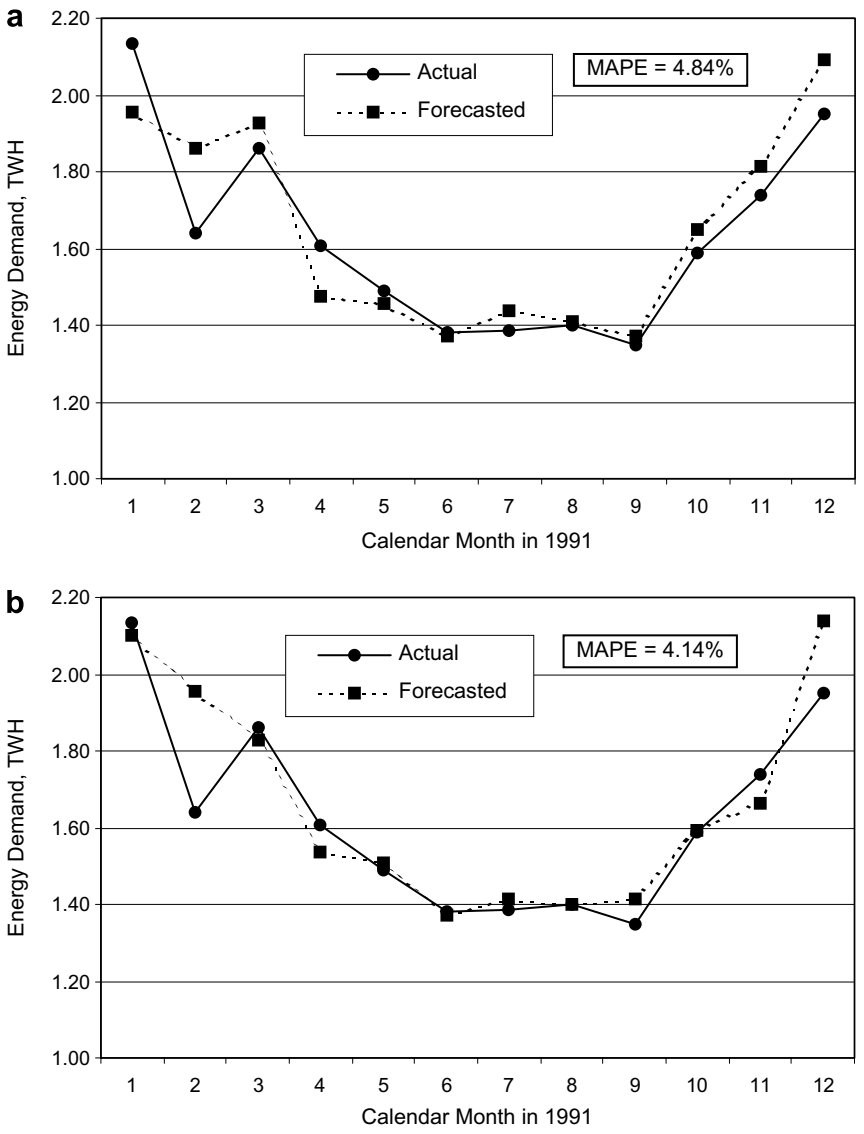


Fig. 4. Plots of actual and forecasted demand over the evaluation year with dedicated twelve forecasting models, one for each month. (a) Using abductive network models with CPM = 1, (b) using neural network models of the structure 12–6–1.

demand with forecasted demand obtained with the 12 neural network models over the evaluation year. The MAPE value of 4.14 is slightly inferior to that of 4.07 shown in Table 3 for the iterative approach using a single next-month model with the same number of hidden neurons.

6. Conclusions

We have demonstrated the use of abductive and neural networks for modeling and forecasting the univariate time series of monthly energy demand. Trained on data for 6 years, models were used to forecast data for the seventh year. Compared to multivariate methods, univariate modeling has the advantage that it does not require additional time series data for other external factors that are often unknown and their data may be difficult to obtain. Moreover, in practical implementations, forecasts for all external time series may be required, which compounds the forecasting effort and increases uncertainty. Forecasting accuracies are superior to those quoted in the literature for similar applications using multiple regression analysis, multivariate modeling, and univariate ARIMA analysis. A simple approach has been used for normalizing for the upward trend in the time series prior to modeling in order to avoid poor extrapolation performance associated with many machine-learning techniques. Two approaches were investigated for obtaining 12-month forecasts: iteratively using a single next-month model, or using 12 dedicated models each forecasting the demand for one month. The former approach has proved more accurate, with both abductive and neural networks giving adequate forecasting accuracies (MAPE values of 3.2% and 3.8%, respectively). Abductive network models select effective inputs and can be simpler than neural network models, which improves interpretability, gives better insight into the modeled system, and allows the derivation of analytical model relationships. We have demonstrated that optimum subsets of model inputs selected automatically through abductive modeling can be used to reduce data dimensionality and therefore improve the forecasting performance of neural network models. Dimensionality reduction is particularly important for reducing model over-fitting and improving generalization in applications with a small number of training examples, which is the case in many medium and long-term demand forecasting applications. Forecasting performance exceeds that of several similar applications reported in the literature.

Acknowledgement

The author wishes to acknowledge the support of the Research Institute and the Department of Computer Engineering at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

References

Abdel-Aal, R. E. (2004). Short term hourly load forecasting using abductive networks. *IEEE Transactions on Power Systems, 19*, 164–173.

Abdel-Aal, R. E., & Al-Garni, A. Z. (1997). Forecasting monthly electric energy consumption in eastern Saudi Arabian using univariate time series analysis. *Energy – The International Journal, 22*, 1059–1069.

Abdel-Aal, R. E., Al-Garni, A. Z., & Al-Nassar, Y. N. (1997). Modeling and forecasting monthly electric energy consumption in eastern Saudi Arabian using abductive networks. *Energy – The International Journal, 22*, 911–921.

AbTech Corporation. (1990). AIM user's manual. Charlottesville, VA.

AlFuhaid, A. A., El-Sayed, M. A., & Mahmoud, M. S. (1997). Cascaded artificial networks for short-term load forecasting. *IEEE Transactions on Power Systems, 12*, 1524–1529.

Al-Garni, A. Z., Zubair, S. M., & Nizami, J. S. (1994). A regression model for electric-energy consumption forecasting in Eastern Saudi Arabia. *Energy – The International Journal, 19*, 1043–1049.

Al-Hamadi, H. M., & Soliman, S. A. (2006). Fuzzy short-term electric load forecasting using Kalman filter. *IEE Proceedings-Generation Transmission and Distribution, 153*, 217–227.

Al-Saba, T., & El-Amin, I. (1999). Artificial neural networks as applied to long-term demand forecasting. *Artificial Intelligence in Engineering, 13*, 189–197.

Alves Da Silva, A. P., Rodrigues, U. P., Rocha Reis, A. J., & Moulin, L. S. (2001). NeuroDem – A neural network based short term demand forecaster. In *IEEE power tech conference*, Porto, Portugal.

Barakat, E. H., & Alrashed, S. A. (1992). Long-range peak-demand forecasting under conditions of high growth. *IEEE Transactions on Power Systems, 7*, 1483–1486.

Barakat, E. H., & Eissa, M. A. M. (1989). Forecasting monthly peak demand in fast growing electric utility using a composite multiregression-decomposition model. *IEE Proceedings C, 136*, 35–41.

Barron, A. R. (1984). Predicted squared error: A criterion for automatic model selection. In S. J. Farlow (Ed.), *Self-organizing methods in modeling: GMDH type algorithms* (pp. 87–103). New York: Marcel-Dekker.

Bielinska, E. M., & Nabaglo, I. E. (1994). Comparison of different methods of bilinear time series prediction. In *Third IEEE conference on control applications* (pp. 1835–1839).

Chatfield, C. (1998). Time series forecasting with neural networks. In *Proceedings of the 1998 IEEE signal processing society workshop on neural networks for signal processing* (pp. 419–427).

Dillon, T. S., Morsztyn, K., & Phula, K. (1975). Short term load forecasting using adaptive pattern recognition and self organizing techniques. In *Proceedings of the fifth power system computer conference*, Cambridge.

Elkateb, M. M., Solaiman, K., & Al-Turki, Y. (1998). A comparative study of medium-weather-dependent load forecasting using enhanced artificial/fuzzy neural network and statistical techniques. *Neurocomputing, 23*, 3–13.

Farlow, S. J. (1984). The GMDH algorithm. In S. J. Farlow (Ed.), *Self-organizing methods in modeling: GMDH type algorithms* (pp. 1–24). New York: Marcel-Dekker.

Frank, R. J., Davey, N., & Hunt, S. P. (2000). Input window size and neural network predictors. In *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks* (pp. 237–242).

Fujimori, S., Kajiya, T., & Mitsuboshi, K. (1998). GA analysis of discharge currents. In *IEEE international symposium on electrical insulation* (pp. 530–533).

Ghiassi, M., Zimbra, D. K., & Saidane, H. (2006). Medium term system load forecasting with a dynamic artificial neural network model. *Electric Power Systems Research, 76*, 302–316.

Gonzalez-Romera, E., Jaramillo-Moran, M. A., & Carmona-Fernandez, D. (2007). Forecasting of the electric energy demand trend and monthly fluctuation with neural networks. *Computers & Industrial Engineering, 52*, 336–343.

Hippert, H. S., Pedreira, C. E., & Souza, R. C. (2001). Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems, 16*, 44–55.

Islam, S. M., Al-Alawi, S. M., & Ellithy, K. A. (1995). Forecasting monthly electric load and energy for a fast growing utility using an artificial neural network. *Electric Power System Research, 34*, 1–9.

Kermanshahi, B., & Iwamiya, H. (2002). Up to year 2020 load forecasting using neural nets. *International Journal of Electrical Power & Energy Systems, 24*, 789–797.

Khotanzad, A., Afkhami-Rohani, R., & Maratukulam, D. (1998). ANNSTLF – Artificial neural network short-term load forecaster - Generation three. *IEEE Transactions on Power Systems, 13*, 1413–1422.

Lewis III, H. W. (2001). Intelligent hybrid load forecasting system for an electric power company. In *Proceedings of the Mountain Workshop on Soft Computing in Industrial Applications* (pp. 25–27).

Liu, X. Q., Ang, B. W., & Goh, T. N. (1991). Forecasting of electricity consumption: A comparison between an econometric model and a neural network model. In *IEEE international joint conference on neural networks* (pp. 1254–1259).

Matsui, T., Iizaka, T., & Fukuyama, Y. (2001). Peak load forecasting using analyzable structured neural network. In *IEEE power engineering society winter meeting* (pp. 405–410).

Montgomery, G. J., & Drake, K. C. (1991). Abductive reasoning networks. *Neurocomputing, 2*, 97–104.

Park, D. C., El-Sharkawi, M. A., Marks, R. J., II, Atlas, L. E., & Damborg, M. J. (1991). Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems, 6*, 442–449.

Sforna, M. (1995). Searching for the electric load-weather temperature function by using the group method of data handling. *Electric Power Systems Research, 32*, 1–9.

Shimakura, Y., Fujisawa, Y., Maeda, Y., Makino, R., Kishi, Y., Ono, M., et al. (1993). Short-term load forecasting using an artificial neural networks. In *Second international forum on applications of neural networks to power systems* (pp. 233–238).

Thiesing, F. M., & Vornberger, O. (1997). Sales forecasting using neural networks. In *International conference on neural networks* (pp. 2125–2128).

Toyada, J., Chen, M., & Inouse, Y. (1970). An application of state estimation to short-term load forecasting, Part I. Forecasting model and II implementation. *IEEE Transactions on Power Systems, 89*, 1678–1688.

Virili, F., & Freisleben, B. (2000). Nonstationarity and data preprocessing for neural network predictions of an economic time series. In *Proceedings of the IEEE international joint conference on neural networks* (pp. 129–134).