# Short-Term Hourly Load Forecasting Using Abductive Networks

Radwan E. Abdel-Aal, *Member, IEEE*

*Abstract*—**Short-term load modeling and forecasting are essential for operating power utilities profitably and securely. Modern machine learning approaches, such as neural networks, have been used for this purpose. This paper proposes using the alternative technique of abductive networks, which offers the advantages of simplified and more automated model synthesis and analytical input-output models that automatically select influential inputs, provide better insight and explanations, and allow comparison with statistical and empirical models. Using hourly temperature and load data for five years, 24 dedicated models for forecasting next-day hourly loads have been developed. Evaluated on data for the sixth year, the models give an overall mean absolute percentage error (MAPE) of 2.67%. Next-hour models utilizing available load data up to the forecasting hour give a MAPE of 1.14%, outperforming neural network models for the same utility data. Two methods of accounting for the load growth trend achieve comparable performance. Effects of varying model complexity are investigated and proposals made for further improving forecasting performance.**

*Index Terms*—**Abductive networks, artificial intelligence, forecasting, GMDH, load forecasting, modeling, neural network applications, neural networks, power system planning, power systems.**

## I. Introduction

ACCURATE load forecasting is a key requirement for the planning and economic and secure operation of modern power systems. Short-term load forecasting (STLF) [one hour to one week] [1] is important for scheduling functions, such as generator unit commitment, hydrothermal 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. Another area of application involves security and load flow studies, including contingency planning, load shedding, and load security strategies. With ever-increasing load capacities, a given percentage forecasting error amounts to greater losses in real terms. Recent changes in the structure of the utility industry due to deregulation and increased competition also emphasize greater forecasting accuracies. STLF activities include forecasting the daily peak load, total daily energy, and daily load curve as a series of 24 hourly forecasted loads.

Traditionally, power utilities have relied in the past on a few highly experienced in-house human experts to perform judgmental forecasts manually [2] using techniques such as the similar-day method. Increased demand on the accuracy, speed, and frequency of the forecasts have gradually led to forecast automation. Conventional techniques for forecasting the load curve included both static and dynamic methods. Static methods model the load as a linear combination of explicit time functions, usually in the form of sinusoids and polynomials [3]. The more accurate dynamic models take into account other important factors such as recent load behavior, weather parameters, and random variations. Techniques in this category include univariate time series models such as the Box-Jenkins integrated autoregressive moving average (ARIMA) [4]. Such methods suffer from limited accuracy because they ignore important weather effects. In addition to being time consuming, they also require extensive user intervention and may be numerically unstable [5]. Multivariate causal models use multiple regression to express the load as a function of exogenous inputs including weather and social variables [6]. In addition to the complexity of the modeling process, regression models are often linear devices that attempt to model distinctly nonlinear relationships [7]. Even when a nonlinear relationship is attempted, it is difficult to determine empirically the correct complex relationship that exists between the load and the other explanatory inputs.

A recent trend in handling such problems that are difficult to solve analytically has been to resort to computational intelligence approaches. The availability of large amounts of historical load and weather data at power utilities has encouraged the use of data-based modeling approaches such as genetic algorithms and neural networks. With such techniques, the user does not need to explicitly specify the model relationship. This enhances their use in automatic knowledge discovery without bias or influence by prior assumptions. With neural networks, 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. Feedforward neural networks trained with error back-propagation have been widely used for load modeling and forecasting (e.g., [7]–[10]). However, the technique suffers from a number of limitations, including difficulty in determining optimum network topology and training parameters [8]. There are many choices to be made in determining numerous critical design parameters with little guidance available [7], and designers often resort to trial and error approaches [9], 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 overfitting and en-

Manuscript received April 13, 2003. This work was supported by the Research Institute of King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

The author is with the Center for Applied Physical Sciences, Research Institute, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia (e-mail: radwan@kfupm.edu.sa).

Digital Object Identifier 10.1109/TPWRS.2003.820695

0885-8950/04$20.00 © 2004 IEEE

sure adequate generalization with new data. Another limitation is the black box nature of neural network models. The models give little insight into the modeled relationship and the relative significance of various inputs, thus providing poor explanation facilities [10]. The acceptability of and confidence in an automated load forecasting tool in an operational environment appear to be related to its transparency and its ability to justify obtained results to human experts [11].

To overcome such limitations, we propose using abductive networks [12] as an alternative machine learning technique to electric load forecasting. We have previously used this approach to model and forecast the monthly domestic electric energy consumption [13], and in forecasting the minimum and maximum daily temperatures [14], [15]. The approach is based on the self-organizing group method of data handling (GMDH) [16]. The potential for GMDH in load forecasting has been realized long ago [17]. However, the technique was somewhat neglected in the literature due to its heuristic nature and limited set of elementary functions [18], as well as the multiple-input-single-output nature of the resulting models and the difficulty of fine-tuning them. Compared to neural networks, however, the method offers the advantages of 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 [8]. With the model represented as a hierarchy of polynomial expressions, resulting analytical model relationships can 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 overfitting by using a proven regularization criterion based on penalizing model complexity, without requiring a separate validation data set during training, as in many neural network paradigms.

This paper applies modern GMDH implementations to hourly load forecasting, illustrating modeling simplicity and adequate forecasting accuracy and highlighting unique explanation capabilities not provided by neural networks. Following a brief description of GMDH and the abductive network modeling tool in Section II, the load and temperature data set used is described in Section III. Next-day hourly load forecasters that predict the full 24-h load curve for a day in one go at the end of the preceding day are described in Section IV. Models were developed using two different approaches to account for the trend of load growth. Next-hour load forecasters that predict the load hour by hour utilizing all data available up to the forecasting hour are presented in Section V. Results are also given when such models are iteratively used to forecast the full next-day load curve. Section VI makes comparisons with neural network models developed using the same data.

## II. GMDH and Aim Abductive Networks

Abductory inductive mechanism (AIM ) [19] 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 (multiphase) 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 $(x_1, x_2, \dots, x_m)$ 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 $(x_i, x_j;\ i, j = 1, 2, \dots, m)$ together with the output $y$ and computing the quadratic regression polynomial [16]:

$$y = A + Bx_i + Cx_j + D{x_i}^2 + E{x_j}^2 + Fx_ix_j. \qquad (1)$$

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 $(z_1, z_2, \dots, z_{m(m-1)/2})$ 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

$${r_k}^2 = \frac{\sum\limits_{\ell=1}^{n_s} (y_\ell - z_{k\ell})^2}{\sum\limits_{\ell=1}^{n_s} {y_\ell}^2}; \quad k = 1, 2, \dots, \frac{m(m-1)}{2}. \qquad (2)$$

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 overfitting 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 which provides the shortest description for the data available [20]. 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 [20] for selection and stopping to avoid model overfitting, 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

$$PSE = FSE + CPM\left(\frac{2K}{n}\right){\sigma_p}^2 \tag{3}$$

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 ${\sigma_p}^2$ 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 $y$ [20]. 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 tradeoff 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 overfit 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 considerably reduces the learning/development time and effort. The models take the form of layered feedforward abductive networks of functional elements (nodes) [19], 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:

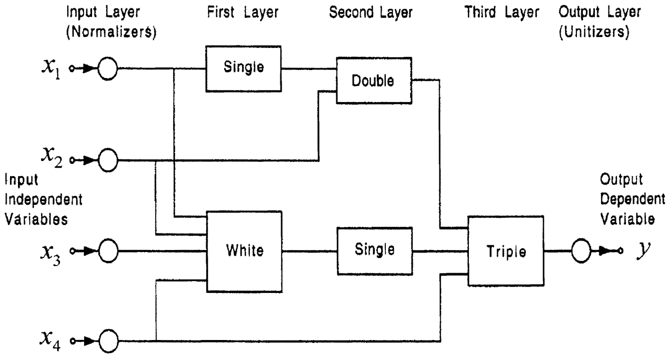


Fig. 1. Typical AIM abductive network model showing various types of functional elements.

i) A white element which consists of a constant plus the linear weighted sum of all outputs of the previous layer, that is

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

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

$$\begin{aligned}\text{“Double” Output} &= w_0 + w_1x_1 + w_2x_2\\ &+w_3{x_1}^2 + w_4{x_2}^2 + w_5x_1x_2 + w_6{x_1}^3 + w_7{x_2}^3.\end{aligned} \tag{5}$$

## III. Data Set

The data set used consists of measured hourly load and temperature data for the Puget power utility, Seattle, WA, over the period from January 1, 1985 to October 12, 1992. It is made available in the public domain by Professor A. M. El-Sharkawi, University of Washington, Seattle [21]. We used the data for five years (1985–1989) for model synthesis and that of the following year (1990) for model evaluation. A few missing load and temperature data, indicated as 0's in the data set, were filled in by interpolating between neighboring values. Table I summarizes the load data for the six-year period and indicates an average annual growth rate of 3.5%. The mean hourly load decreased slightly in 1986, but has then kept steadily increasing. For the evaluation year of 1990, we use an estimated hourly mean because, in practice, no actual data would be available for the evaluation year. This mean was estimated using a straight line fit for the mean hourly loads of only the previous four years (1986–1989) having a steady increase in the load. Two approaches were attempted in accounting for the trend of load growth. In the first approach, all hourly load data were first normalized so that all years have an annual hourly mean load equal to that of the last training year (1989). Let the mean hourly load for year $i$ be $M_i$, the normalization factor $f_i$ for that year is given by

$$f_i = \frac{M_{1989}}{M_i}. \tag{6}$$

These normalization factors are given in the second column from right in Table I. Normalization was performed by multiplying the hourly load values for each year by the corresponding normalization factor. With the second approach, no normalization of the hourly load data was necessary, and load growth was represented by an additional model input defined for year $i$ as

$$r_i = \frac{M_i}{M_{1985}}. \qquad (7)$$

Values for this input are given in the last column of Table I. This approach reduces the data preprocessing work required.

## IV. Next-Day Hourly Load Forecasters

We have developed 24 models that forecast the full hourly load curve for the following day (d) in one go at the end of the preceding day $(\mathrm{d}-1)$. A model is dedicated for forecasting the load, EL (d,h), for each hour of the day. The models were trained using data for five years (1985–1989) and evaluated on the year 1990. All models use the same set of inputs which includes: 24 hourly loads at day $(\mathrm{d}-1)$ $(\mathrm{L1}, \mathrm{L2}, \ldots, \mathrm{L24})$, the measured minimum (Tmin) and maximum (Tmax) air temperatures on day $(\mathrm{d}-1)$, the forecasted minimum (ETmin) and maximum (ETmax) air temperatures on day (d), and the day type for forecasting day (d). The day type was coded as four mutually exclusive binary inputs representing a working day (Monday to Friday) [WRK], a Saturday (SAT), a Sunday (SUN), and an official holiday (HOLI). Tmin and Tmax were taken as the minimum and maximum values of the 24 hourly temperatures provided for the day. In the absence of forecasted data for the minimum and maximum air temperatures for the following day, we used actual values instead, which would be the case with ideal temperature forecasts. We have investigated the effect of introducing Gaussian noise depicting temperature forecasting errors that would be present in practice. A record in the training dataset for the model for hour h $(\mathrm{h}=1,2,\ldots,24)$ includes 32 input variables and takes the form of Table II. Prior to training and evaluation, all hourly load data for inputs and output were normalized to 1989 loads by multiplying by the normalization factors shown in Table I. For model evaluation, the effect of normalization was first removed by dividing forecasted loads for the year 1990 by the estimated normalization factor for that year, and then comparing the results with the year's actual load. To avoid the effect of discontinuities at year boundaries (input loads for day $(\mathrm{d}-1)$ being multiplied by a different normalization factor from that of output load for day (d) as those two days fall in different years), the first day of each year was excluded as a forecasted day in both training and evaluation. This has left us with 1821 training records (1985–1989, 1988 being a leap year) and 364 evaluation records in 1990. Training was performed using the default value $CPM = 1$ for the complexity penalty multiplier.

Fig. 2 shows the abductive network model synthesized for forecasting the load at hour 1 (midnight). This is a one-element model that uses only loads L3, L20, and L24 of the preceding day. Neither temperature nor day-type inputs feature in the model, which is a nonlinear function of the load time series only. Activities at that time of the day do not vary much from day to day. The figure shows the resulting equations for all functional elements, and the predicted output is calculated by substituting in the given set of five equations. Equation of the Triple element indicates the nonlinear nature of the model.

The forecasting model for hour 12 shown in Fig. 3 is a more complex four-layer model that uses both forecasted extreme temperatures and the SUN day-type input. The model also uses the following loads for the preceding day: last load (L24), the load at the same forecasting hour (L12), as well as L7 and L22. Fig. 3 shows also the performance of the model in the form of scatter and time series plots of the actual and forecasted data at that hour over the evaluation year. The scatter plot shows a best line fit and the value of the Pearson's correlation coefficient as 0.98, and the time series plot shows the mean absolute percentage error (MAPE) as 2.40%. Table III summarizes the model structure for all of the 24 models, listing the model inputs selected and the number of layers and elements. The table indicates that loads for hours 1 and 2 are not much affected by weather or day type, but are primarily determined by the load

TABLE I
Summary of the Six-Year Load Data Showing Year-to-Year Growth and the Factors Used by the Two Methods for Dealing With Trend

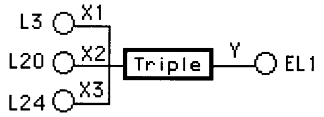
| Year, $i$ | | Total Annual Load, (MWH) | Mean Hourly Load, $M_i$ (MW) | Annual Load Growth (year-to-year) | Normalization Factors, $f_i$ (Method 1) | Trend Input Values, $r_i$ (Method 2) |
|---|---|---|---|---|---|---|
| 1985 | | 16,310,645 | 1862 | 1 | 1.130 | 1 |
| 1986 | | 16,017,335 | 1828 | 0.982 | 1.151 | 0.982 |
| 1987 | | 16,510,405 | 1885 | 1.031 | 1.116 | 1.012 |
| 1988 | | 17,563,434 | 2000 | 1.061 | 1.052 | 1.074 |
| 1989 | | 18,434,815 | 2104 | 1.052 | 1 | 1.130 |
| 1990 | Actual | 19,357,130 | 2210 | 1.050 | 0.952 | 1.187 |
| | Estimated | 19,184,400 | 2190 | 1.041 | 0.961 | 1.176 |
| Average Load Growth 1986-1990 (Actual) | | | | 1.035 | | |

TABLE II
Training Record Layout for Next-Day Forecasting Model for Hour h

| Inputs | | | | Output |
|---|---|---|---|---|
| 24 hourly loads for day (d-1) | Extreme Temperatures for day (d-1) | Forecasted Extreme Temperatures for day (d) | Day type code for day (d) | Load for hour (h) on day (d) |
| L1, L2, ..., L24 | Tmin, Tmax | ETmin, ETmax | WRK, SUN, SAT, HOLI | L(d,h) |

- Normalizer Equations:

X1 = - 4.52 + 0.00303 L3
X2 = - 4.66 + 0.00295 L20
X3 = - 5.61 + 0.00315 L24



- Triple Equation:

Y = 0.125 X1 + 0.868 X3 – 0.115 X1 X2 + 0.0506 X1 X3 + 0.0582 X2 X3

- Unitizer Equation:

EL1 = 1600 + 312 Y

Fig. 2. Structure and equations for the next-day load forecaster for hour 1.

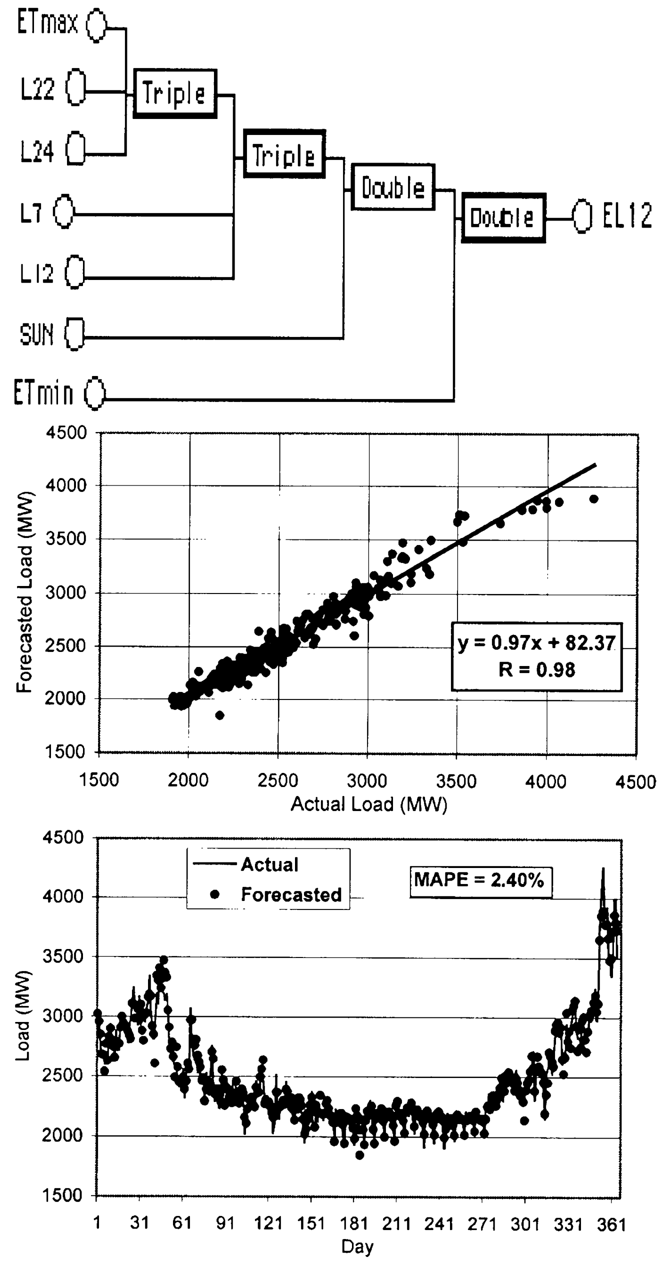


Fig. 3. Structure and performance of the next-day load forecasting model for hour 12 over the evaluation year.

TABLE III
SUMMARY OF THE ABDUCTIVE NETWORK MODELS FOR THE 24 NEXT-DAY HOURLY LOAD FORECASTERS

| Day (d) Forecasting Hour | Model Inputs | | | | Model Structure | |
|---|---|---|---|---|---|---|
| | Day (d-1) Load at Hours: | Temperature | | Day (d) Day Type | Number of Layers | Total Number of Elements |
| | | Day (d-1) | Day (d) | | | |
| 1 | 3,20,24 | | | | 1 | 1 |
| 2 | 17,21,24 | | | | 1 | 1 |
| 3 | 1-13,15,17-24 | Tmin | ETmin | WRK | 1 | 1 |
| 4 | 1-24 | Tmin | ETmin | WRK | 1 | 1 |
| 5 | 1,3-5,7-24 | Tmin, Tmax | ETmin | WRK, SAT, SUN | 1 | 1 |
| 6 | 1-24 | Tmin, Tmax | ETmin, ETmax | WRK, SAT, SUN | 2 | 2 |
| 7 | 1-24 | Tmin, Tmax | ETmin, ETmax | WRK, SAT, SUN | 2 | 2 |
| 8 | 1-24 | Tmin, Tmax | ETmin, ETmax | WRK, SAT, SUN | 3 | 3 |
| 9 | 1-6,8-10,12-24 | Tmin | ETmin, ETmax | WRK, SAT, SUN | 2 | 2 |
| 10 | 1-24 | Tmin, Tmax | ETmin, ETmax | WRK, SUN | 3 | 3 |
| 11 | 7,15,21,24 | | ETmax | SUN | 3 | 3 |
| 12 | 7,12,22,24 | | ETmin, ETmax | SUN | 4 | 4 |
| 13 | 8,12,22 | Tmax | ETmin, ETmax | WRK, SUN | 4 | 4 |
| 14 | 2,8,13,22 | Tmax | ETmin, ETmax | WRK, SUN | 4 | 4 |
| 15 | 7,9,13,16,18,22 | Tmax | ETmin, ETmax | WRK | 4 | 5 |
| 16 | 1-15,17-24 | Tmax | ETmin, ETmax | WRK, SUN | 4 | 4 |
| 17 | 2,7,18,22 | Tmax | ETmax | WRK | 4 | 4 |
| 18 | 1-11,13-22,24 | Tmin, Tmax | ETmax | WRK, SAT | 2 | 2 |
| 19 | 1-11,13,14,16-24 | Tmin, Tmax | ETmin, ETmax | WRK, SUN | 2 | 2 |
| 20 | 4,8,20 | Tmax | ETmax | WRK | 3 | 3 |
| 21 | 1,7,16,21 | Tmax | ETmax | WRK | 3 | 3 |
| 22 | 1-13,15-24 | Tmin, Tmax | ETmin, ETmax | WRK, SAT | 2 | 2 |
| 23 | 1-3,5-7,9-12, 14,15-23 | Tmin, Tmax | ETmin, ETmax | WRK | 3 | 3 |
| 24 | 2,8,23 | Tmin | ETmax | SUN | 3 | 3 |

time series. Forecasted temperature and day-type inputs feature in all remaining models. Model complexity and nonlinearity increases as the forecasting hour progresses and the lead time increases. Loads for hours 11 and 12 are significantly influenced by whether it is a Sunday. The second column in Table IV lists the MAPE values for all hours, giving the average for the evaluation year as 2.67%.

Values in the third column of Table IV were obtained with the other method of representing load growth trend as an additional input variable having a different value for each year as given in the last column of Table I and using actual non-normalized load data for both training and evaluation. The additional trend input was selected by all models except those for the first four hours of the day. The results are comparable on average with those for the data normalization method, but exhibit an exceptionally larger error for hour 14. All remaining results in this paper were obtained using the data normalization method for handling the trend. To demonstrate the advantage over flat forecasting, column 4 in Table IV shows the results when the load on the forecasting day was assumed equal to that on the same day of the previous week. Abductive network forecasts are about 2.5 times more accurate. Full-day load curves were forecasted using all 24 models for four days of the evaluation year which represent a working day, a Saturday, a Sunday, and a holiday in the same season over the interval from August 8 to September 3, 1990, and the results are shown in Fig. 4. Forecasting accuracy is best for the working day and poorest for the holiday due to the fewer examples of holiday load patterns encountered during training. However, average MAPE value for the ten holiday days forecasted in the evaluation year is 5.12%. The third generation of the ANNSTLF neural network load forecaster employing a special holiday forecasting technique gives 9.68% for some utilities [22].

We have investigated the effect of simulated errors in the ideal forecasted extreme temperature values ETmin and ETmax for the load-forecasting day. As seen from Table III, the model for hour 12 is an example of 13 forecasters that use both ETmin and ETmax, and would therefore be affected most by such errors.

TABLE IV
HOURLY MAPE VALUES OVER THE EVALUATION YEAR FOR NEXT-DAY LOAD FORECASTING MODELS USING TWO METHODS OF ACCOUNTING FOR LOAD GROWTH TREND AND FOR A NAIVE FORECAST

| Forecasting hour, h | Trend removed by normalizing data (Method 1) | Trend represented as a separate model input (Method 2) | Naive Forecast: Forecasted load = load on the same day of the previous week |
|---|---|---|---|
| 1 | 1.14 | 0.95 | 6.49 |
| 2 | 1.60 | 1.32 | 6.96 |
| 3 | 1.34 | 1.23 | 7.46 |
| 4 | 1.58 | 1.47 | 7.81 |
| 5 | 2.07 | 1.72 | 8.15 |
| 6 | 2.68 | 2.49 | 8.43 |
| 7 | 3.43 | 3.34 | 8.77 |
| 8 | 3.40 | 3.26 | 8.25 |
| 9 | 2.59 | 2.37 | 6.97 |
| 10 | 2.06 | 1.86 | 5.93 |
| 11 | 2.71 | 2.14 | 5.34 |
| 12 | 2.40 | 2.32 | 5.29 |
| 13 | 2.47 | 2.59 | 5.36 |
| 14 | 2.66 | 4.64 | 5.40 |
| 15 | 3.02 | 3.30 | 5.70 |
| 16 | 3.18 | 3.48 | 6.04 |
| 17 | 3.54 | 3.46 | 6.44 |
| 18 | 3.56 | 3.57 | 6.56 |
| 19 | 3.31 | 3.29 | 6.42 |
| 20 | 3.65 | 3.29 | 6.38 |
| 21 | 3.29 | 3.11 | 6.00 |
| 22 | 3.05 | 3.01 | 5.84 |
| 23 | 2.65 | 2.74 | 5.99 |
| 24 | 2.84 | 2.84 | 6.11 |
| Average | 2.67 | 2.66 | 6.59 |

There are nine other models (e.g., for hours 3 and 18), that use either ETmin or ETmax only, and would be affected by such errors to a lesser degree. The remaining two models for hours 1 and 2 do not use either variables. Simulated Gaussian random errors of zero mean and standard deviation $\sigma$ were added to the ideal forecasted two temperature values in both the training and evaluation datasets for the load forecaster for hour 12. The MAPE of 2.40% for the noiseless case increased to 2.53% for $\sigma = 1°\mathrm{F}$ and to 2.60% for $\sigma = 2°\mathrm{F}$, indicating an acceptable degradation in forecasting accuracy.

The effect of varying the complexity of the resulting forecasting models was investigated for the model for hour 12. Table V shows the structure and performance of the resulting more complex model with $CPM = 0.2$ and simpler model with $CPM = 5$, in comparison with the default model having $CPM = 1$. It is noted that load input L12 features in all three models, which indicates its importance in explaining the modeled output. The level of model complexity varies widely from a 37-input, four-layer nonlinear model at $CPM = 0.2$ to a simple two-input linear model at $CPM = 5$. While further model simplification significantly degrades forecasting accuracy, there are signs that more complex models may improve performance compared to the default model. As indicated in the table, more complex models require longer training times.

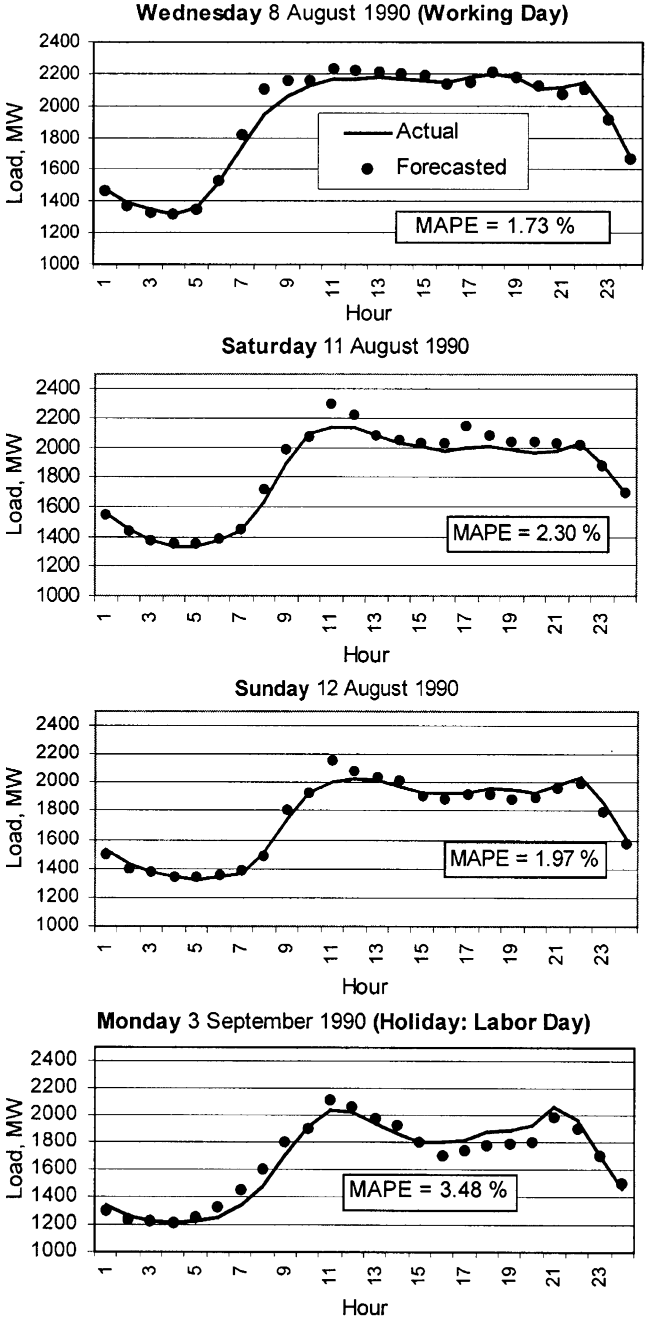


Fig. 4. Performance of the 24 next-day hourly forecasters on four typical days representing a working day, a Saturday, a Sunday, and a public holiday during the summer of the evaluation year.

## V. NEXT-HOUR LOAD FORECASTING

We have developed 24 models for forecasting the load at the next hour (h) during day (d) using the full hourly load data on day (d − 1) (L1, L2, ..., L24) and all available hourly load data on day (d) up to, and including, the preceding hour (h − 1) (NL1, NL2, ..., NL(h − 1)) as well as temperature and day-type information. Contrary to the case of next-day hourly forecasters, the number of load inputs here is not fixed, but varies from 24 for hour 1 to 47 for hour 24. With the number

TABLE V
EFFECT OF THE CPM PARAMETER ON THE COMPLEXITY AND PERFORMANCE OF NEXT-DAY LOAD FORECASTING MODELS FOR HOUR 12

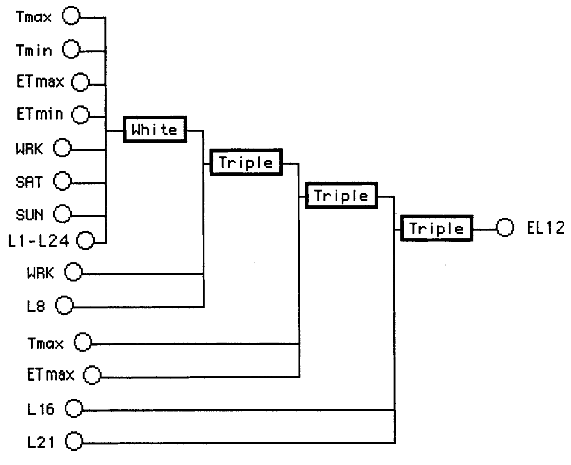
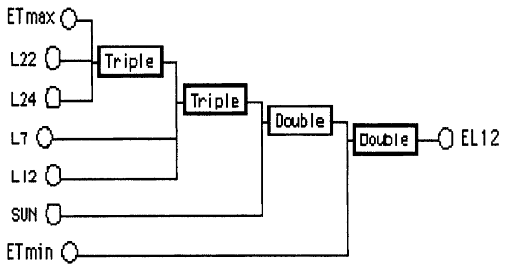
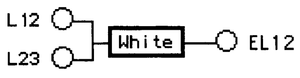
| CPM | Model Structure | Relative Training Time | MAPE, % |
|---|---|---|---|
| 0.2 |  Tmax<br>Tmin<br>ETmax<br>ETmin<br>WRK<br>SAT<br>SUN<br>L1-L24<br>White<br>Triple<br>Triple<br>Triple<br>EL12<br>WRK<br>L8<br>Tmax<br>ETmax<br>L16<br>L21 | 1.20 | 2.21 |
| 1 |  ETmax<br>L22<br>L24<br>Triple<br>Triple<br>Double<br>Double<br>EL12<br>L7<br>L12<br>SUN<br>ETmin | 1.00 | 2.40 |
| 5 |  | 0.58 | 3.69 |

TABLE VI
TRAINING RECORD LAYOUT FOR NEXT-HOUR FORECASTING MODEL FOR HOUR (h)

| Inputs | | | | | Output |
|---|---|---|---|---|---|
| 24 hourly loads for day (d-1) | (h-1) available hourly loads on day (d) | Average Temperature for day (d-1) | Forecasted Average Temperature for day (d) | Day type code for day (d) | Load for hour (h) on day (d) |
| L1, L2, ..., L24 | NL1, NL2, ...NL(h-1) | Ta | ETa | WRK | L(d,h) |

of model inputs limited to 50 for the AIM version used, we had only three inputs left to represent temperature and day-type information. Temperature was represented using the average temperature Ta on day $(\mathrm{d}-1)$ and the forecasted average temperature ETa for day (d). Again, ETa was taken as actual Ta for day (d). Day type for the forecasting day (d) was represented by a single binary input (WRK) that is 1 for a working day and 0 otherwise. A record in the training dataset for the model for hour h $(\mathrm{h}=2,3,\ldots,24)$ takes the form of Table VI. Training was performed on 1821 records (1985–1989) with the default value $CPM = 1$ for the complexity penalty multiplier and 364 evaluation records in 1990. All load data were normalized to account for load growth as described in Section IV above. Table VII summarizes the model structure for all the 24 hourly models, listing the model inputs selected and the corresponding time lags in the load time series and showing a sketch of the model structure. Compared to models for next-day hourly loads, next-hour models are much simpler, reflecting the relative ease of forecasting with previous load data as recent as the previous hour being available. For example, the model for hour 12 is a three-input, one-element as compared to a seven-input four-layer model for the corresponding next-day hourly model. Dependence on previous day loads is reduced, with half of the 24 models totally ignoring them in favor of the more recent loads on the forecasting day. The exogenous temperature variable is used by only one model, and the day type by two models. No use is made of the forecasted average temperature ETa, and, therefore, results are not affected by any forecasting errors in ETa in practice. The models are dominated by the load time series, with the time lag of 1 h featured in all models. The middle column in Table VIII lists the MAPE values for all hours, giving the overall value for the evaluation year as 1.14%, indicating the effectiveness of such models for very short-term load forecasting. A neural network trained on three months of the same utility data (working days only) was reported to give a MAPE of 1.41% when evaluated on 22 days [23]. Inputs to the

TABLE VII
ABDUCTIVE NETWORK MODELS FOR THE 24 NEXT-HOUR LOAD FORECASTERS

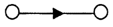
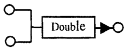
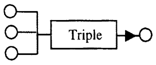
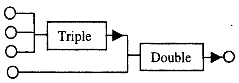
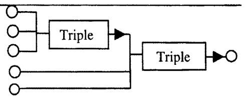
| Day (d) Forecasting Hour | Model Input(s) | | | Load Time Lags Selected | Model Structure |
|---|---|---|---|---|---|
| | Day (d-1) Load at Hours: | Day (d) Load at Hours: | Other | | |
| 2 | | 1 | | 1 |  |
| 3 | | 2 | | 1 | |
| 4 | | 3 | | 1 | |
| 8 | | 7 | | 1 | |
| 7 | | 6,5 | | 1,2 |  |
| 1 | 24,20,3 | | | 1,5,22 |  |
| 5 | 10 | 4 | WRK | 1,19 | |
| 6 | 22 | 5,1 | | 1,5,8 | |
| 9 | | 8,6,4 | | 1,3,5 | |
| 10 | 18 | 9,8 | | 1,2,16 | |
| 11 | 16 | 10,9 | | 1,2,19 | |
| 12 | | 11,9,8 | | 1,3,4 | |
| 13 | | 12,10,7 | | 1,3,6 | |
| 14 | | 13,10,8 | | 1,4,6 | |
| 15 | | 14,11,8 | | 1,4,7 | |
| 17 | 18,11 | 16 | | 1,13,20 | |
| 19 | | 18,16,9 | | 1,3,10 | |
| 20 | 20,18 | 19 | | 1,24,26 | |
| 21 | 22,20 | 20 | | 1,23,25 | |
| 22 | 23,20 | 21 | | 1,23,26 | |
| 23 | 23,21 | 22 | | 1,24,26 | |
| 18 | | 17,15,9 | Ta | 1,3,9 |  |
| 24 | 19,9 | 23,22 | WRK | 1,2,29,39 |  |

TABLE VIII
PERFORMANCE OF THE NEXT-HOUR LOAD FORECASTERS (COLUMN 2) AND NEXT-DAY LOAD FORECASTS USING SUCH FORECASTERS ITERATIVELY AT THE END OF THE PRECEDING DAY (COLUMN 3)

| Forecasting hour, h | MAPE, % | |
|---|---|---|
| | Next-hour forecasting | Next-day forecasting (Iterated application) |
| 1 | 1.14 | 1.14 |
| 2 | 1.01 | 1.87 |
| 3 | 0.93 | 2.53 |
| 4 | 0.88 | 3.21 |
| 5 | 1.08 | 4.16 |
| 6 | 1.27 | 5.95 |
| 7 | 2.08 | 8.39 |
| 8 | 1.55 | 8.13 |
| 9 | 1.28 | 5.98 |
| 10 | 0.82 | 4.56 |
| 11 | 0.94 | 4.16 |
| 12 | 0.70 | 4.08 |
| 13 | 0.80 | 4.17 |
| 14 | 0.69 | 4.52 |
| 15 | 0.70 | 4.91 |
| 16 | 0.77 | 5.33 |
| 17 | 1.27 | 5.73 |
| 18 | 1.31 | 6.23 |
| 19 | 1.48 | 6.48 |
| 20 | 1.25 | 6.13 |
| 21 | 1.59 | 6.07 |
| 22 | 1.23 | 5.17 |
| 23 | 1.20 | 4.20 |
| 24 | 1.29 | 3.90 |
| Average | 1.14 | 4.87 |

neural network included measured loads and temperatures at the two immediately preceding hours, estimated temperature at the forecasting hour, and an hour index. Table IX summarizes the MAPE error histograms for all forecasting hours over the evaluation year for both next-day and next-hour models.

The third column of Table VIII lists the MAPE values for next-day (day (d)) forecasting obtained by repetitive use of the next-hour models for all hours up to, and including, the forecasting hour (h). In practice, this would be performed at the end of day $(\mathrm{d}-1)$, with the load forecasted for hour (i) being fed, among other required inputs, to the next-hour model for hour $(\mathrm{i}+1)$. As expected, performance of this type of next-day forecasting is inferior to that given in Table IV, with the overall average MAPE being 4.87% compared with 2.67%. This is because next-hour models are heavily dependent on recent hourly loads on the forecasting day. With these values being forecasted rather than measured, forecasting errors accumulate. As seen from Table VII, next-hour models almost totally ignore temperature and day-type information and, therefore, should not be expected to form as a basis for accurate next-day forecasting which depends heavily on such parameters as indicated in Table III.

Four of the next-hour models in Table VII (hours 2, 3, 4, and 8) take the simple 1-input "wire" form, in which the functional element is a direct connection from the normalizer unit of the input to the unitizer unit generating the output. In all four cases, the model input is the load at the immediately preceding hour, indicating some form of load persistence. However, due to the equations of both the normalizer and unitizer units, the input-output model relationship is not necessarily that of simple persistence where the forecasted load is equal to that of the preceding hour. Table X lists the overall model equations derived for the four wire models and compares their forecasting performance with that of simple persistence. MAPE values for persistence can be as high as seven times those for the synthesized abductive wire models.

## VI. COMPARISON WITH NEURAL NETWORK MODELS

The default abductive network models obtained with $CPM = 1$ for forecasting the load at hour 12 were compared with the corresponding back propagation neural network models using the same training and evaluation data. The networks were the default function approximation models synthesized by the NeuroExpert module of the NeuroSolutions 4 software for Windows. Twenty-percent of the training data

TABLE IX
MAPE ERROR STATISTICS FOR NEXT-DAY AND NEXT-HOUR FORECASTS

| MAPE Error | Percentage population of forecasting hours over evaluation year | |
|---|---|---|
| | Next-day models | Next-hour models |
| ≤ 1% | 29% | 57% |
| ≤ 3% | 68% | 95% |
| ≥ 6% | 9% | 0.25% |

TABLE X
PERFORMANCE COMPARISON BETWEEN THE FOUR NEXT-HOUR "WIRE" MODELS IN TABLE VII WITH SIMPLE LOAD PERSISTENCE OVER THE EVALUATION YEAR

| Forecasting Hour | Abductive 'Wire' Model | | Simple Persistence | |
|---|---|---|---|---|
| | Equation | MAPE, % | Equation | MAPE, % |
| 2 | L(h) =1.020 L(h-1) – 105.86 | 1.01 | L(h) = L (h-1) | 4.90 |
| 3 | L(h) =1.034 L(h-1) – 77.68 | 0.93 | | 1.95 |
| 4 | L(h) =1.045 L(h-1) – 62.97 | 0.88 | | 1.17 |
| 8 | L(h) =1.087 L(h-1) – 92.70 | 1.55 | | 10.66 |

TABLE XI
COMPARISON BETWEEN ABDUCTIVE AND NEURAL MODELS FOR HOUR 12

| Forecast | Technique | Model Structure | Absolute Percentage Forecasting Error | |
|---|---|---|---|---|
| | | | Mean (MAPE), % | Maximum, % |
| Next-day | Abductive | 7-1-1-1-1-1 (Fig. 3) | 2.40 | 15.10 |
| | Neural | 32-6-1 | 2.68 | 17.80 |
| Next-hour | Abductive | 3-1-1 (Table V) | 0.70 | 3.45 |
| | Neural | 38-5-1 | 0.90 | 5.49 |

were used for cross validation. The networks used a single hidden layer with neurons having a hyperbolic tangent transfer function, and a linear function for the output neuron. Table XI compares the structure and performance of the neural and abductive network models obtained for both next-day and next-hour forecasting. The table indicates that the abductive models fare better for both the mean and maximum forecasting errors, particularly for next hour forecasting. It may be possible to further improve the performance of both techniques through fine-tuning these default models. Neural networks for load forecasting usually use the sigmoid transfer function for the output neurons [7]. Making this change reduced the maximum forecasting error by the next-day neural model in Table XI to 14%, but degraded the overall performance of the next-hour model. Structure comparison shows the simpler and more transparent nature of the abductive models. For example, the next-hour neural model is a 38-input model that does not readily give any indication of the most relevant inputs or the pertinent model relationship. On the other hand, the abductive model (Table VII) shows that the load at hour 12 is adequately determined by only three previous hourly loads and gives a manageable analytical relationship.

## VII. CONCLUSION

We have demonstrated the use of abductive network machine learning as an alternative tool for next-day and next-hour electric load forecasting. Compared to neural networks, the approach simplifies model development, automatically selects effective inputs, gives better insight into the load function, and allows comparison with previously used analytical models. While next-day models utilized exogenous inputs such as temperature and day-type variables, next-hour models developed were largely influenced by the load time series. Forecasting performance compares favorably with that of neural network models. Future work will attempt to further improve the forecasting accuracy through the inclusion of hourly temperature data and the development of dedicated seasonal models. The technique will be extended to other applications including peak load forecasting.

## REFERENCES

[1] G. Gross and F. D. Galiana, "Short-term load forecasting," *Proc. IEEE*, vol. 75, pp. 1558–1573, Dec. 1987.

[2] M. C. Brace, V. Bui-Nguyen, and J. Schmidt, "Another look at forecast accuracy of neural networks," in *Proc. IEEE 2nd Int. Forum Applicat. Neural Networks To Power Syst.*, 1993, pp. 389–394.

[3] W. R. Christianse, "Short term load forecasting using general exponential smoothing," *IEEE Trans. Power App. Syst.*, vol. PAS-90, pp. 900–911, Mar./Apr. 1971.

[4] S. Vemuri, D. Hill, and R. Balasubramanian, "Load forecasting using stochastic models," in *Proc. 8th Power Ind. Comput. Applicat. Conf.*, 1973, pp. 31–37.

[5] A. S. AlFuhaid, M. A. El-Sayed, and M. S. Mahmoud, "Cascaded artificial networks for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 12, pp. 1524–1529, Nov. 1997.

[6] A. D. Papalexopoulos and T. C. Hesterberg, "A regression-based approach to short-term system load forecasting," *IEEE Trans. Power Syst.*, vol. 5, pp. 1535–1547, Nov. 1990.

[7] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Trans. Power Syst.*, vol. 16, pp. 44–55, Feb. 2001.

[8] A. P. A.A. P. Alves da Silva, U. P. Rodrigues, A. J. R.A. J. Rocha Reis, and L. S. Moulin, "NeuroDem—a neural network based short term demand forecaster," in *Proc. IEEE Power Tech. Conf.*, Porto, Portugal, 2001.

[9] W. Charytoniuk and M. S. Chen, "Neural network design for short-term load forecasting," in *Proc. 1st Int. Conf. Elect. Utility Deregulation, Restructuring, Power Technol.*, 2000, pp. 554–561.

[10] T. Matsui, T. Iizaka, and Y. Fukuyama, "Peak load forecasting using analyzable structured neural network," in *Proc. IEEE Power Eng. Soc. Winter Meet.*, vol. 2, 2001, pp. 405–410.

[11] H. W. Lewis III, "Intelligent hybrid load forecasting system for an electric power company," in *Proc. Mountain Workshop Soft Comput. Ind. Appl.*, 2001, pp. 25–27.

[12] G. J. Montgomery and K. C. Drake, "Abductive networks," in *Proc. Int. Soc. Opt. Eng. App. Artificial Neural Networks Conf.*, 1990, pp. 56–64.

[13] R. E. Abdel-Aal, A. Z. Al-Garni, and Y. N. Al-Nassar, "Modeling and forecasting monthly electric energy consumption in eastern Saudi Arabian using abductive networks," *Energy—The Int. J.*, vol. 22, pp. 911–921, Sept. 1997.

[14] R. E. Abdel-Aal and M. A. Elhadidy, "A machine-learning approach to modeling and forecasting the minimum temperature at Dhahran, Saudi Arabia," *Energy—The Int. J.*, vol. 19, pp. 739–749, July 1994.

[15] ——, "Modeling and forecasting the maximum temperature using abductive machine learning," *Weather and Forecasting*, vol. 10, pp. 310–325, June 1995.

[16] S. J. Farlow, "The GMDH algorithm," in *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, S. J. Farlow, Ed. New York: Marcel-Dekker, 1984, pp. 1–24.

[17] T. S. Dillon, K. Morsztyn, and K. Phula, "Short term load forecasting using adaptive pattern recognition and self organizing techniques," in *Proc. 5th Power Syst. Comput. Conf.*, Cambridge, Sept. 1975.

[18] M. F. Tenorio and W.-T. Lee, "Self-organizing network for optimum supervised learning," *IEEE Trans. Neural Networks*, vol. 1, pp. 100–110, Mar. 1990.

[19] AbTech Corporation, AIM User's Manual, Charlottesville, VA, 1990.

[20] A. R. Barron, "Predicted squared error: a criterion for automatic model selection," in *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, S. J. Farlow, Ed. New York: Marcel-Dekker, 1984, pp. 87–103.

[21] . [Online]http://www.ee.washington.edu/class/559/2002spr/

[22] A. Khotanzad, R. Afkhami-Rohani, and D. Maratukulam, "ANNSTLF-Artificial neural network short-term load forecaster-generation three," *IEEE Trans. Power Syst.*, vol. 13, pp. 1413–1422, Nov. 1998.

[23] D. C. Park, M. A. El-Sharkawi, R. J. Marks II, L. E. Atlas, and M. J. Damborg, "Electric load forecasting using an artificial neural network," *IEEE Trans. Power Syst.*, vol. 6, pp. 442–449, May 1991.

**Radwan E. Abdel-Aal** (M'84) received the B.Sc. degree in electrical engineering from Cairo University, Cairo, Egypt, in 1972, the M.Sc. degree in aviation electronics from Cranfield Institute of Technology, Bedford, U.K., in 1974, and the Ph.D. degree from Strathclyde University, Glasgow, U.K., in 1983.

Currently, he is a Research Scientist I with the Research Institute of King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, where he has been since 1985. From 1974 to 1978, he was a Design Engineer with Microwave Associates Ltd., Dunstable, U.K. He was a Research Assistant and Research Fellow with Strathclyde University from 1979 to 1985. His current research interests include nuclear physics instrumentation, machine learning, and multidisciplinary data mining applications.