Modeling and Forecasting Electric Daily Peak Loads Using Abductive Networks

R. E. Abdel-Aal
Department of Computer Engineering,
King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

Address for corresponding author:

Dr. R. E. Abdel-Aal
P. O. Box 1759
KFUPM
Dhahran 31261
Saudi Arabia

e-mail: radwan@kfupm.edu.sa
Phone: +966 3 860 4320
Fax: +966 3 860 4281

Abstract

Forecasting the daily peak load is important for secure and profitable operation of modern power utilities. Machine learning techniques including neural networks have been used for this purpose. This paper proposes the alternative modeling approach of abductive networks, which offers simpler and more automated model synthesis. Resulting analytical input-output models automatically select influential inputs, give better insight and explanations, and allow comparison with other empirical models. Developed using peak load and extreme temperature data for five years and evaluated on the 6th year, a model forecasts next-day peak loads with an overall mean absolute percentage error (MAPE) of 2.50%, outperforming neural network models and flat forecasting for the same data. Two methods are described for forecasting daily peak loads up to one week ahead through iterative use of the next-day model or using 7 dedicated models. Effects of varying model complexity are considered, and simplified analytical expressions are derived for the peak load. Proposals are made for further improving the forecasting accuracy.

Keywords: Abductive networks, Neural networks, GMDH, Machine learning, Modeling, Forecasting, Load forecasting, Peak load, STLF, Power system planning.

1. Introduction

Short-term load forecasting (STLF) [1] is important for performing many power utility functions, including generator unit commitment, hydro-thermal coordination, short-term maintenance, fuel allocation, power interchange, transaction evaluation, as well as network analysis functions, security and load flow studies, 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 forecasting covers the daily peak load, total daily energy, and daily load curve as a series of 24 hourly forecasted loads. This paper is concerned with modeling and forecasting daily peak loads with lead times of 1 to 7 days.

Univariate time series techniques such as the Box-Jenkins integrated autoregressive moving average (ARIMA) [2] have been used for peak load forecasting. However, These techniques have limited accuracy because they ignore important weather effects, are time consuming, require extensive user intervention and may become numerically unstable [3]. Multivariate causal models use multiple regression to express the peak load as a function of exogenous inputs including weather and social variables [4]. In addition to the complexity of the modeling process, regression models are often linear devices that attempt to model distinctly nonlinear relationships [5]. Even when a nonlinear relationship is attempted, it is difficult to determine empirically the correct complex relationship that exists between the peak load and the other explanatory inputs.

The availability of large amounts of historical load and weather data at power utilities has encouraged the use of data-based machine learning modeling methods such as neural networks. With such techniques, the user does not need to explicitly specify the model relationship, which enhances automatic knowledge discovery without bias or influence by prior assumptions. 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. Feed-forward neural networks trained with error back-propagation have been widely used for modeling and forecasting the daily peak load, e.g. [6-13]. However, neural networks suffer from a number of limitations, including difficulty in determining optimum network topology and training parameters [14]. There are many choices to be made in determining numerous critical design parameters with little guidance available [5], and designers often resort to trial and error approaches [15] 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. Recently, techniques such as genetic algorithms have been proposed for the automatic optimization of the structure and parameters of neural networks [16]. 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 [17]. The acceptability of, and confidence in, an automated load forecasting tool in an operational environment is related to its transparency and its ability to justify obtained results to human experts [18].

To overcome such limitations, we propose using the alternative machine learning technique of abductive networks [19] for daily peak load forecasting. We have previously used this approach to model and forecast next day's hourly load profile [20], monthly domestic electric energy consumption [21], and the minimum and maximum daily temperatures [22,23]. The approach is based on the self-organizing group method of data handling (GMDH) [24]. The potential for GMDH in load forecasting has been realized about three decades ago [25]. However, the technique was somewhat neglected in the literature due to its heuristic nature and limited set of elementary functions [26], 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, and automatic configuration of the model structure [14]. While relevant input variables are selected automatically during abductive network training, with neural networks this requires additional and separate preprocessing in the form of feature extraction using techniques such as principal components analysis (PCA) [27]. With the abductive model represented as a hierarchy of polynomial expressions, resulting analytical model relationships can provide greater insight into the modeled phenomena, highlight contributions of various inputs, and allow comparison with previously used empirical or statistical models. Information on the significance of the various inputs to the neural network output requires additional effort, e.g. in the form of sensitivity analysis [28]. While many conventional neural network paradigms use a separate validation data set to guard against overfitting, thus reducing the amount of data available for actual training, the method proposed here uses an automatic stopping criterion that penalizes model complexity and operates on the full training set. Analytical model relationships are also easier to export to other software applications compared to neural network models.

This paper uses modern GMDH approaches to model and forecast daily peak loads up to one week ahead, 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 2, the load and temperature data set used is described in Section 3. Next-day peak load forecasters are described in Section 4 and their performance compared with neural networks and flat forecasting methods. In Section 5, two different abductive modeling approaches are presented for forecasting the daily peak load up to 7 days ahead, and the influence of temperature forecasting errors is considered.

2. GMDH and AIM Abductive Networks

AIM (abductory inductive mechanism) [29] 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 ($x_1, x_2, ..., x_m$) 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 ; $i,j = 1,2,...,m$) together with the output y and computing the quadratic regression polynomial [24]:

$$y = A + B x_i + C x_j + D x_i^2 + E x_j^2 + F x_i x_j \qquad (1)$$

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

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

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 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 [30]. 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 [30] 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:

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

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* [30]. 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 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 reduces the learning/development time and effort. The models take the form of layered feed-forward abductive networks of functional elements (nodes) [29], 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.:

$$\text{"White" Output} = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n \tag{4}$$

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

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

3. The Data Set

The 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. It is made available in the public domain by Professor A. M. El-Sharkawi, University of Washington, Seattle, USA [31]. We used the data for 5 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 original data set, were filled-in by interpolating between neighboring values. Table 1 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 obtained from a straight line fit for the mean hourly loads of only the previous four years (1986-1989) having a steady increase in the load. To account for the trend of load growth, 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:

$$f_i = M_{1989} / M_i \quad ; i = 1985, 1986, \ldots, 1990 \qquad (6)$$

These factors are listed in the last column of Table 1. Hourly loads were normalized by multiplying actual values for each year by the corresponding normalization factor. Normalized daily peak load was then determined as the peak of the normalized hourly load profile for the day.

4. Next-Day Peak Load Forecasting

We have developed an abductive network model that forecasts the peak load on the next day (d+1) in terms of data available on day (d) regarding the peak load (PL), day type, and temperature for the seven days preceding the forecasting day, as well as day type and forecasted temperatures for the forecasting day. The model was trained using data for five years (1985-1989) and evaluated on the year 1990. The set of model inputs consists of 47 variables, including six variables for each of the seven preceding days (d-6, d-5, …, d-1, d) and five variables for the forecasting day (d+1). Day type information was coded as three mutually exclusive binary inputs representing a working day (Monday to Friday) (WRK), a Saturday (SAT), and a Sunday or official holiday (SUN/HOL). Temperature information included the measured daily maximum (Tmax) and minimum (Tmin) air temperatures on days (d-6, d-5, …, d-1, d), and the corresponding forecasted values (ETmax and ETmin) on the forecasting day (d+1). Tmax and Tmin were taken as the maximum and minimum values of the 24 hourly temperatures provided for the day. In the absence of forecasted data for the maximum and minimum air temperatures for the following day, we used actual values instead, which would be the case with ideal temperature forecasts. In Section 5 we investigate the effect of introducing Gaussian noise depicting temperature forecasting errors that would be present in practice. A record in the training data set for the model takes the form shown in Table 2.

Prior to model training and evaluation, all peak load data for inputs and output were normalized to 1989 loads by multiplying by the normalization factors shown in Table 1. For model evaluation, the effect of normalization was removed by dividing forecasted loads for the year 1990 by the estimated normalization factor for that year, and the results compared with the year's actual loads. To allow for data for seven preceding days to be used in the training set, the first forecasting day used for training was 8 January 1985. This has left us with 1819 training records (358 in 1985, 365 in each of 1986, 1987, and 1989, and 366 in 1988, the latter being a leap year). For the evaluation year 1990, it is possible to use preceding days belonging to 1989, and therefore 365 records were made available for evaluation. Training was performed using the default value $CPM = 1$.

Fig. 2 shows the structure of the abductive network model synthesized for forecasting the next-day peak load PL(d+1). This is a 3-layer, 3-element model that uses only 25 of the 47 input variables that are made available for training. Inputs selected are those which are most effective in explaining the output variable. Excluding irrelevant inputs helps simplify the resulting model and speed up its execution, and reduce the effects of possible measurement or forecasting errors that may be associated with such inputs. Commonly used neural network techniques do not automatically select relevant inputs, and resulting models would therefore suffer from such limitations. The model shows a strong coupling between the peak load on the forecasting day (d+1) and inputs pertaining to the same day in the previous week (d-6). All inputs for that day except the SUN/HOL day type input are selected. Peak loads on all preceding seven days are also selected, together with both the maximum and minimum temperatures on the forecasting day and the preceding day. Fig. 2 shows also the performance of the model in the form of scatter and time series plots of the actual and forecasted daily peak loads 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 2.50%. Table 3 summarizes the performance of the model on various day types over the evaluation year. At 2.27%, the MAPE error for working days is the lowest, as such days are well represented in the training set. Ref. [6] gives results on 12 dedicated neural network models that predict the peak load on the working day Thursday using various combinations of input variables and data normalization schemes. The MAPE error obtained varied from 1.96% to 4.80%, with 10 out of the 12 models giving MAPE ≥ 2.37%. In Table 3, the largest MAPE error of 3.45% is for holidays, which form the smallest day type group. The third generation of the ANNSTLF neural network load forecaster, which uses a special technique for forecasting holiday peak loads, gives a MAPE of 9.17% for some utilities [32]. Sundays are as many as Saturdays, but have better forecasting accuracy. Combining Sundays and holidays, which have a somewhat similar load profile, increases the number of representative examples in the training set.

Table 4 compares various aspects of the forecasting performance of the model over the evaluation year with four other approaches using abductive, neural, and flat forecasting techniques on the same data. The table lists the mean and maximum forecasting errors as well as the percentages of evaluation days with the absolute percentage error (APE) being ≤ 1%, ≤ 3%, and ≥ 6%. The alternative abductive network method determines the peak load as the peak of the hourly load profile for the following day forecasted using 24 dedicated abductive network models described in [20]. The neural network model is the default function approximation model automatically synthesized by the NeuroExpert module of the NeuroSolutions 4 software for Windows using the same training data. 20% of the training data were used for cross validation. The model uses a single hidden layer of four neurons and one output neuron. With naive forecast-1, the peak load on the forecasting day is taken as the peak load on the same day in the previous week, i.e.

$$PL(d+1) = PL(d-6) \qquad (7)$$

With naive forecast-2, the peak load on the forecasting day is taken as the peak load on the previous day if both days are of the same day type, e.g. working days:

$$PL(d+1) = PL(d) \qquad (8)$$

If not, and the forecasting day is not a holiday, the peak load is taken as the peak load on the same day in the previous week as given by Equation (7). However, if it is a holiday, the peak load is taken as that of the nearest previous holiday or Sunday. Results in Table 4 show the clear advantage of abductive and neural network methods over flat forecasting. The two abductive network approaches give comparable MAPE values that are superior to all other forecasting approaches, with naive forecast-1 giving the poorest performance. Abductive modeling of the peak load directly gives somewhat lower maximum errors and lower frequency of large errors as compared to obtaining it as the peak of the forecasted hourly load profile. The former approach is also much more convenient as only one model is trained and maintained, as compared to 24 models in the latter approach [20]. Comparable performance of the two approaches suggests that when the next-day hourly load profile is forecasted, then the peak load forecast can be determined as the peak of the forecasted profile, making it unnecessary to build an additional dedicated model for peak load forecasting. However, for longer lead times, forecasting the hourly profile may prove more difficult, and the dedicated approach to forecast the peak load would be necessary in such cases, see Section 5. The neural network model gives poorer MAPE performance compared to the abductive network models. Although it gives a lower maximum percentage error, larger errors are much more frequent. The neural model is a 47-input model that does not readily give any indications on the model relationship or the significant inputs.

The effect of varying model complexity from the default model having *CPM* = 1 was investigated. Complexity varied widely from a 4-layer nonlinear model with 37 different inputs for *CPM* = 0.2 to the simple 1-element, 3-input linear model shown in Fig. 3 for *CPM* = 5. The latter overly simplified model significantly degraded forecasting accuracy, increasing the MAPE value from 2.50% to 3.87%, while performance of the more complex model was comparable with that the default model. It is noted that load inputs PL(d) and PL(d-6) are used by all three models, indicating the importance of such inputs in explaining the peak load on day (d+1). Symbolic substituting for the equations given in Fig. 3 for the functional elements of the *CPM* = 5 model and then dividing by the normalization factor of 0.961 leads to the following linear relationship for the peak load:

$$PL(d+1) = \text{٠,}2453\ PL(d-6) + \text{٠,٧٨}30\ PL(d) + 176\ SUN/HOL(d) + 1.68, \tag{9}$$

which does not use temperature. It gives a rough estimate of tomorrow’s peak load during the evaluation year using only today’s peak load, the peak load six days ago, and a Boolean indication of whether today was a Sunday/Holiday (1) or not (0). The forecasted load contains a small portion of the load on day (*d-6*) and a larger portion of the most recent load on day (*d*). If that day was a Sunday or a holiday, 176 MW is added to account for the low load on the previous day. Expressions such as Equation (9) give some insight into the forecasting process, and it would be interesting to compare them with empirical models and manual methods employed by operators.

5. Next-Week Peak load forecasting

We have considered two approaches for forecasting the daily peak loads over the week ahead. In the first approach, the next-day (d+1) peak load model described in Section 4 is used iteratively at the end of day (d) to forecast the peak loads for days d+1, d+2, …, up to day d+7. The peak load forecasted in the ith iteration of the model is used as the PL(d+i) input in next iterations. As shown in Section 4, 1819 records are used for training the (d+1) model. Day (d+h); h = 1, 2, …, 7 is forecasted using h iterations, and (366-h) evaluation records are available in the year 1990. Therefore, the number of records used for evaluating the forecasts for one to seven days in advance are 365, 364, …, and 359, respectively. With this approach, forecasts for the extreme temperatures ETmax and ETmin are required for all days up to and including the forecasting day.

In the second approach, seven dedicated models are developed, each forecasting the peak load on a day of the week ahead. A record in the training data set for the model that forecasts the peak load on day (d+h) takes the form shown in Table 5. The model that forecasts day (d+2) uses 9 January 1985 as the first forecasting day, which gives 1818 training records. In general, the model for day (d+h) uses (1820-h) training records, and the number of records used to train the seven models (h = 1,2,…,7) is 1819, 1818, …, and 1813, respectively. For the evaluation year 1990, use of an increasing number of preceding days belonging to 1989 allows having the full set of 365 evaluation records with each of the seven forecasting models. With this approach, forecasts for the extreme temperatures ETmax and ETmin are required only for the forecasting day. It is noted that both approaches become the same for the case of h = 1, i.e. forecasting tomorrow's peak load. Table 6 summarizes the structures of the seven peak load forecasting models for days (d+h); h = 1,2,…,7. Selected model inputs are marked with a dot, and the numbers of model inputs and layers are listed. In general, model complexity increases initially with the increase in lead time h and then decreases. For small lead times, forecasting is relatively easy, and simple models are synthesized. As the lead time increases, model complexity increases to match the increased difficulty in forecasting. However, with further increase in lead time, increase in model complexity does not prove effective in improving forecasting accuracy, and simpler models are synthesized again. The model for h = 4 has the largest number of inputs and layers. The peak load on the same day as the forecasting day a week ago (i.e. day d+h-7) is selected by all models, as well as all temperature and peak load data on day d, which is the last day on which data is available. All models also use both forecasted temperatures (ETmax and ETmin) on the forecasting day (d+h), which indicates the importance of accurate temperature forecasts. WRK is the only day type input selected on the forecasting day by all models, indicating the importance of whether the day is a working day or not.

Performance of the two approaches is compared in Table 7. The left hand side of the table shows performance with ideal temperature forecasts for all days ahead, where ETmax and ETmin for a given day were taken as the actual values measured on that day. The table shows a small gradual increase in forecasting error with the increase in lead time, with the MAPE value for day 7 being around 3.10%. The corresponding value quoted for the third generation of the ANNSTLF neural network load forecaster is 3.52% for some utilities [32]. It is noted that both forecasting approaches in Table 7 give nearly identical performance, with the same average MAPE of 2.96% over the full forecasting week.

We have also considered the effect of simulating forecasting errors in ETmax and ETmin that exist in practice. Simulated Gaussian random errors of zero mean and standard deviation σ were added to the ideal forecasted two temperature values in both the training and evaluation data sets. As the lead time increases, it is expected that the temperature forecasting error increases. Given a linear increase in error variance with lead time and an error variance of 2° F on the first day ahead, the temperature forecasting error variance on day h is assumed to take the form:

$$\sigma^2_h = 2 + (h-1), \qquad h = 1,2,\ldots,7 \tag{10}$$

This gives error standard deviations of 1.41° F and 2.83° F for the first and 7th days ahead, respectively. Corresponding values of the mean absolute error (MAE) computed for 365 samples of simulated Gaussian noise having zero mean and the above two values for σ are 1.05° F and 2.19° F, respectively. The right hand side of Table 7 shows performance of peak load forecasting using the two forecasting approaches with erroneous temperature forecasts for all days ahead. In general, temperature forecasting errors lead to an acceptable degradation in forecasting accuracy, with the average MAPE error over the forecasting week increasing by only 0.36 and 0.38 percentage points for the first and second approaches, respectively. The iterated approach is influenced by temperature forecast errors on all days up to and including the forecasting day, while the other approach is affected by temperature errors on the forecasting day only. Therefore, it is expected that the iterated approach would be degraded more by such errors compared to the other approach. However, as seen from the h = 1 column of Table 6, Tmax and Tmin are not necessarily selected as model inputs on all previous days. Out of the seven previous days, both extreme temperatures are selected on only two days (d and d-6), none are used on two days (d-2 and d-4), and only Tmin is used on the remaining three days. This reduces the actual degradation in the performance of the iterated approach as indicated by the two figures quoted above for the increase in the average MAPE value with temperature errors. Results in Table 7 show that the iterated approach using a single (d+1) model has similar overall performance to that of the multiple model approach. The former has the added advantages of the simplicity and convenience of developing and maintaining only one model instead of seven models.

6. Conclusions

We have demonstrated the use of abductive network machine learning as an alternative tool for forecasting daily peak loads up to one week ahead. Compared to neural networks, the approach simplifies model development, automatically selects effective inputs, gives better insight into the load function, and allows comparison with empirical models. Simplified next-day models confirm the significance of peak loads on the previous day and on the same day of the previous week. Performance compares favorably with that of neural network models, and holiday forecasts are superior to those quoted for dedicated forecasters in commercial use. Iterative use of a single next-day model gives adequate forecasting for up to 7 days ahead. Future work will attempt to further improve the forecasting accuracy by developing dedicated models based on seasons and day types.

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

[1] Gross G, Galiana FD. Short-term load forecasting. Proc. IEEE 1987;75:1558-73.

[2] El Desouky AA, Elkateb MM. Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA. IEE Proc. C 2000;147:213 –17.

[3] Alfuhaid AS, El-Sayed MA, Mahmoud MS. Cascaded artificial networks for short-term load forecasting. IEEE Trans. Power Systems 1997;12:1524-29.

[4] Haida T, Muto S. Regression based peak load forecasting using a transformation technique. IEEE Trans. Power Systems 1994;9:1788-94.

[5] Hippert HS, Pedreira CE, Souza RC. Neural networks for short-term load forecasting: A review and Evaluation. IEEE Trans. Power Systems 2001;16:44–55.

[6] Asar A-U, Mcdonald JR. A specification of neural network applications in the load forecasting problem. IEEE Trans. Control Systems Technology 1994;2:135–41.

[7] Hsu Y–Y, Yang C–C. Design of artificial neural networks for short-term load forecasting. II. Multilayer feedforward networks for peak load and valley load forecasting. IEE Proc. C 1991;138:414 –8.

[8] Onoda T. Next day's peak load forecasting using an artificial neural network. Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems, 1993, pp. 284-9.

[9] Aboul-Magd MA, Ahmed, EE-DE-S. An artificial neural network model for electrical daily peak load forecasting with an adjustment for holidays. Proceedings of the Large Engineering Systems Conference on Power Engineering, 2001, pp.105–13.

[10] Morioka Y, Sakurai K, Yokoyama, A, Sekine, Y. Next day peak load forecasting using a multilayer neural network with an additional learning. Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems, 1993, pp. 60–5.

[11] Dillon TS, Sestito S. and Leung S. Short term load forecasting using an adaptive neural network'. Journal of Electrical Power and Energy Systems. 13(4);186-192, 1991.

[12] Dillon TS and Sestito S. Short term load forecasting using neural networks. In T.S. Dillon and D. Niebur, eds. Neural Networks Applications in Power Systems. Leicester, UK, CRL Publishing Ltd, 1996, pp. 387-404.

[13] Senjyu T, Takara H, Uezato K, Funabashi T. One-hour-ahead load forecasting using neural network. IEEE Transactions on Power Systems 17(1);113–118, 2002.

[14] Alves Da Silva AP, Rodrigues UP, Rocha Reis AJ, Moulin LS. NeuroDem - a neural network based short term demand forecaster. IEEE Power Tech Conference, Porto, Portugal, 2001.

[15] Charytoniuk W, Chen MS. Neural network design for short-term load forecasting. Proceedings of the International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, 2000, pp. 554–61.

[16] Lam HK, Ling SH, Leung FHF, Tam PKS. Tuning of the structure and parameters of neural network using an improved genetic algorithm. The 27th Annual Conference of the IEEE Industrial Electronics Society, 2001, pp. 25-30.

[17] Matsui T, Iizaka T, Fukuyama Y. Peak load forecasting using analyzable structured neural network. Proceedings of the IEEE Power Engineering Society Winter Meeting, 2001, pp. 405–10.

[18] Lewis III HW. Intelligent hybrid load forecasting system for an electric power company. Proceedings of the Mountain Workshop on Soft Computing in Industrial Applications, 2001, pp. 25-7.

[19] Montgomery GJ, Drake KC. Abductive networks. Proceedings of the SPIE conference on the Applications of Artificial Neural Networks, 1990, pp. 56-64.

[20] Abdel-Aal RE. Short term hourly load forecasting using abductive networks. IEEE Trans. Power Systems 2004;19:164-73.

[21] Abdel-Aal RE, Al-Garni AZ, Al-Nassar YN. Modelling and forecasting monthly electric energy consumption in eastern Saudi Arabian using abductive networks. Energy - The International Journal 1997;22:911-21.

[22] Abdel-Aal RE, Elhadidy MA. A machine-learning approach to modelling and forecasting the minimum temperature at Dhahran, Saudi Arabia. Energy - The International Journal 1994;19:739-49.

[23] Abdel-Aal RE, Elhadidy MA. Modeling and forecasting the maximum temperature using abductive machine learning. Weather and Forecasting 1995;10:310-25.

[24] Farlow SJ. The GMDH algorithm. In: Farlow, S. J. (Ed.), Self-Organizing Methods in Modeling: GMDH Type Algorithms. Marcel-Dekker, New York, 1984, pp. 1-24.

[25] Dillon TS, Morsztyn K, Phula K. Short term load forecasting using adaptive pattern recognition and self organizing techniques. 5th Power systems Computer conference (PSCC-5), Cambridge, September 1975, Paper 2.4/3, pp. 1-15.

[26] Tenorio MF, Lee W-T. Self-organizing network for optimum supervised learning. IEEE Trans. Neural Networks 1990;1:100-10.

[27] Saini LM, Soni MK. Artificial neural network-based peak load forecasting using conjugate gradient methods. IEEE Trans. on Power Systems 2002;17:907–12.

[28] Frost F, Karri V. Determining the influence of input parameters on BP neural network output error using sensitivity analysis. Proceedings of the Third International Conference on Computational Intelligence and Multimedia Applications, 1999, pp. 45–49.

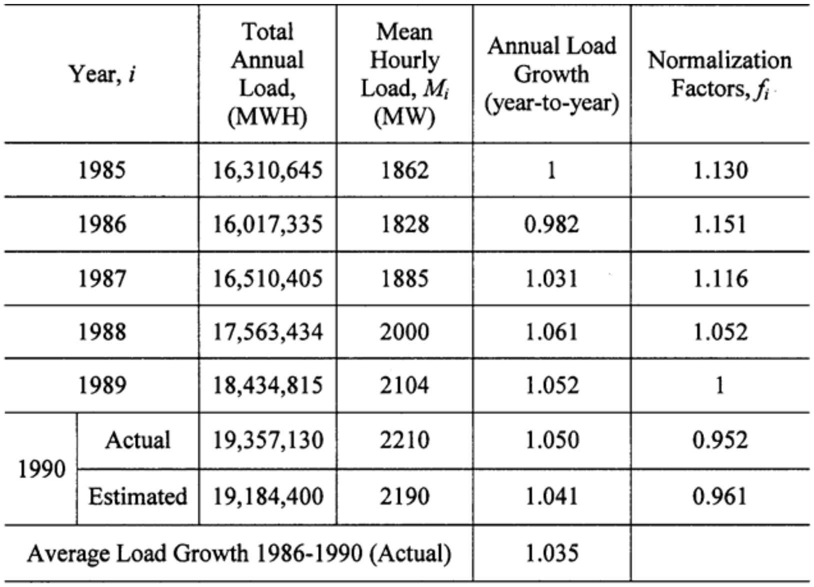
[29] AbTech Corporation, 1990. AIM User's Manual, Charlottesville, VA.

[30] Barron AR. Predicted squared error: A criterion for automatic model selection. In: Farlow, S. J. (Ed.), Self-Organizing Methods in Modeling: GMDH Type Algorithms. Marcel-Dekker, New York, 1984, pp. 87-103.

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

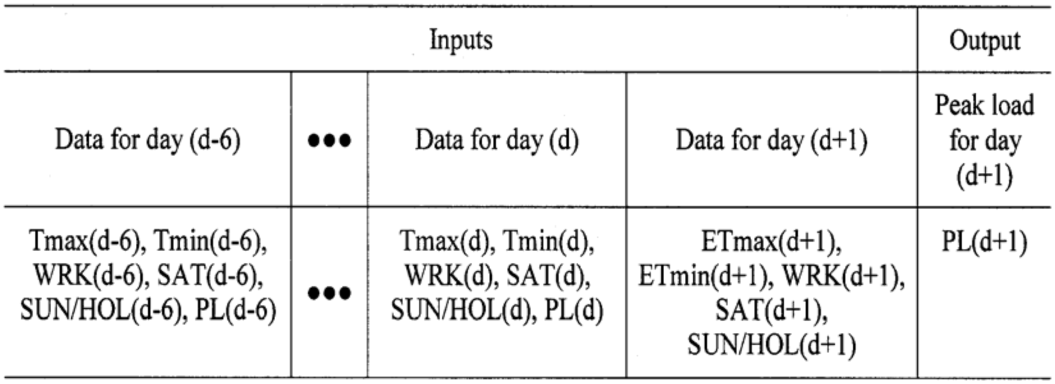
[32] Khotanzad A, Afkhami-Rohani R, Maratukulam D. ANNSTLF-Artificial neural network short-term load forecaster- generation three. IEEE Trans. Power Systems 1998;13:1413–22.

Table 1. Summary of the 6-year load data showing year-to-year growth and the normalization factors used for handling trend.



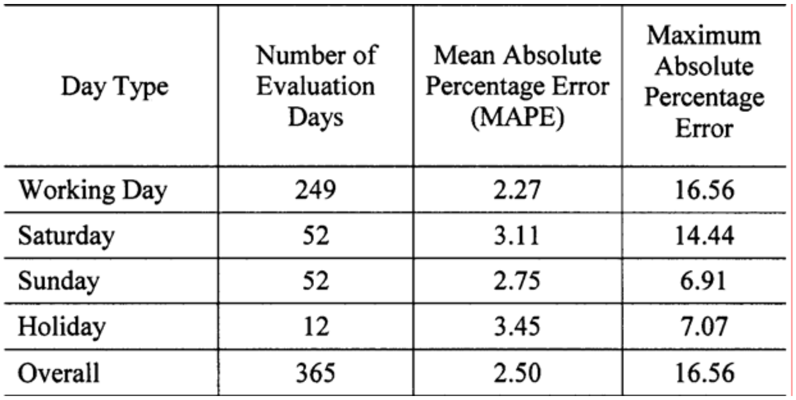
Year, *i*		Total Annual Load, (MWH)	Mean Hourly Load, M_i (MW)	Annual Load Growth (year-to-year)	Normalization Factors, f_i
1985		16,310,645	1862	1	1.130
1986		16,017,335	1828	0.982	1.151
1987		16,510,405	1885	1.031	1.116
1988		17,563,434	2000	1.061	1.052
1989		18,434,815	2104	1.052	1
1990	Actual	19,357,130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

Table 2. Layout of the training records for the peak load forecasting model for the next day (d+1).



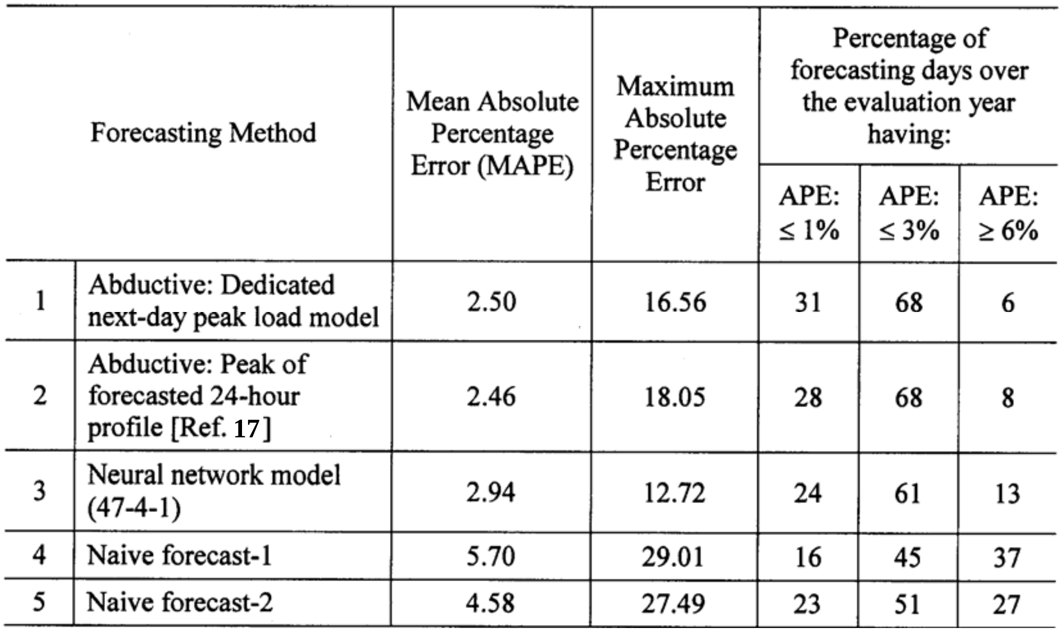
Inputs				Output
Data for day (d-6)	•••	Data for day (d)	Data for day (d+1)	Peak load for day (d+1)
Tmax(d-6), Tmin(d-6), WRK(d-6), SAT(d-6), SUN/HOL(d-6), PL(d-6)	•••	Tmax(d), Tmin(d), WRK(d), SAT(d), SUN/HOL(d), PL(d)	ETmax(d+1), ETmin(d+1), WRK(d+1), SAT(d+1), SUN/HOL(d+1)	PL(d+1)

Table 3. Performance of the next-day peak load forecasting model on holidays and various days of the week over the evaluation year.



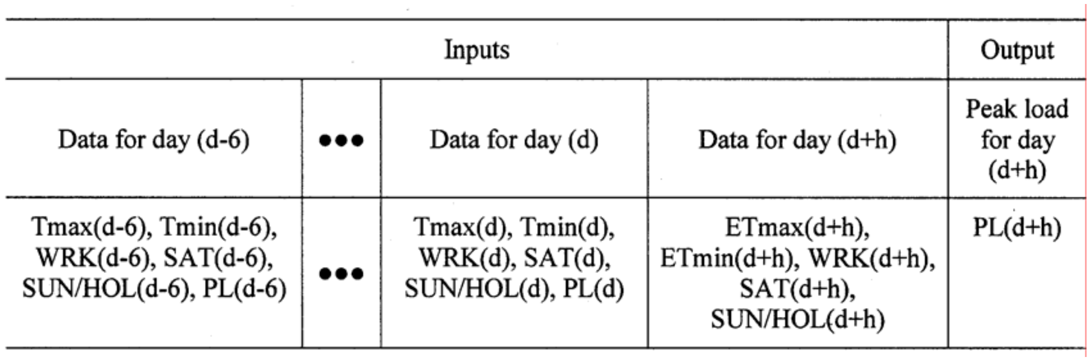
Day Type	Number of Evaluation Days	Mean Absolute Percentage Error (MAPE)	Maximum Absolute Percentage Error
Working Day	249	2.27	16.56
Saturday	52	3.11	14.44
Sunday	52	2.75	6.91
Holiday	12	3.45	7.07
Overall	365	2.50	16.56

Table 4. Comparison of the performance of the peak load forecaster with other abductive, neural, and flat forecasting methods



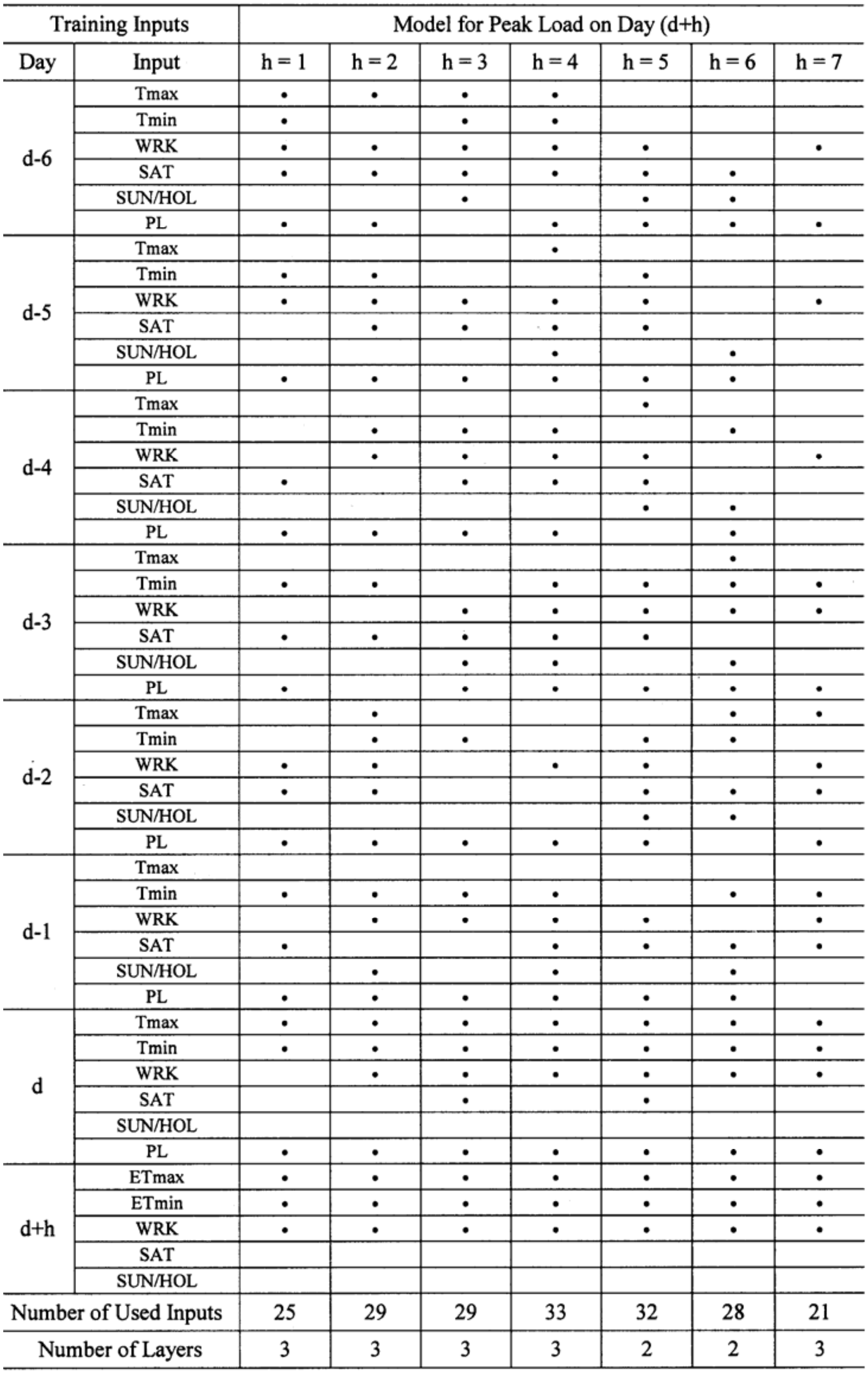
Forecasting Method		Mean Absolute Percentage Error (MAPE)	Maximum Absolute Percentage Error	Percentage of forecasting days over the evaluation year having:		
				APE: ≤ 1%	APE: ≤ 3%	APE: ≥ 6%
1	Abductive: Dedicated next-day peak load model	2.50	16.56	31	68	6
2	Abductive: Peak of forecasted 24-hour profile [Ref. 17]	2.46	18.05	28	68	8
3	Neural network model (47-4-1)	2.94	12.72	24	61	13
4	Naive forecast-1	5.70	29.01	16	45	37
5	Naive forecast-2	4.58	27.49	23	51	27

Table 5. Layout of the training records for the peak load forecasting model for day (d+h); h = 1, 2, …, 7.



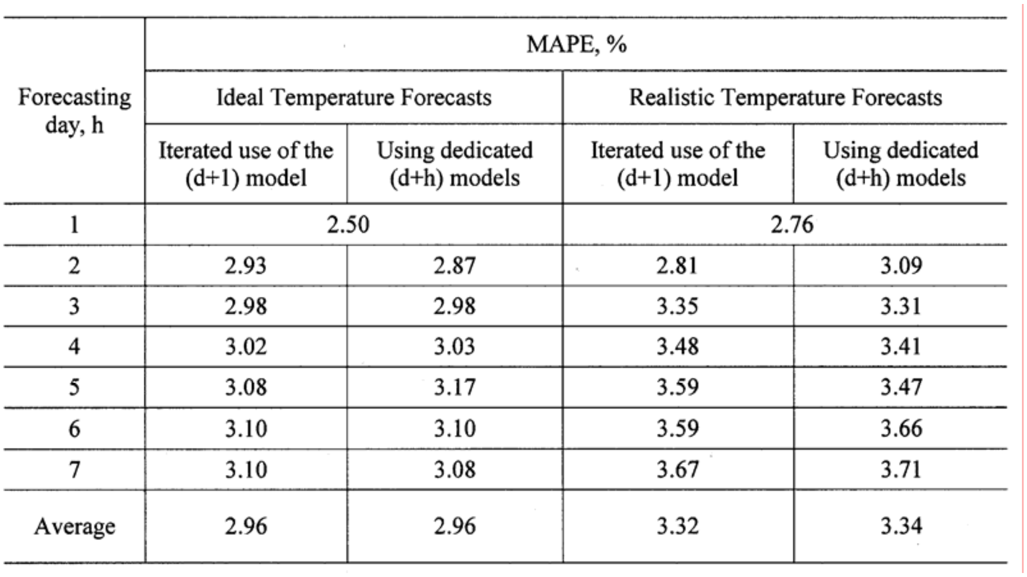
Inputs				Output
Data for day (d-6)	●●●	Data for day (d)	Data for day (d+h)	Peak load for day (d+h)
Tmax(d-6), Tmin(d-6), WRK(d-6), SAT(d-6), SUN/HOL(d-6), PL(d-6)	●●●	Tmax(d), Tmin(d), WRK(d), SAT(d), SUN/HOL(d), PL(d)	ETmax(d+h), ETmin(d+h), WRK(d+h), SAT(d+h), SUN/HOL(d+h)	PL(d+h)

Table 6. Summary of the abductive network models for the seven next-week daily peak load forecasters



Training Inputs		Model for Peak Load on Day (d+h)						
Day	Input	h = 1	h = 2	h = 3	h = 4	h = 5	h = 6	h = 7
d-6	Tmax	•	•	•	•			
	Tmin	•		•	•			
	WRK	•	•	•	•	•		•
	SAT	•	•	•	•	•	•	
	SUN/HOL			•		•	•	
	PL	•	•		•	•	•	•
d-5	Tmax				•			
	Tmin	•	•			•		
	WRK	•	•	•	•	•		•
	SAT		•	•	•	•		
	SUN/HOL				•		•	
	PL	•	•	•	•	•	•	
d-4	Tmax					•		
	Tmin		•	•	•		•	
	WRK		•	•	•	•		•
	SAT	•		•	•	•		
	SUN/HOL					•	•	
	PL	•	•	•	•		•	
d-3	Tmax						•	
	Tmin	•	•		•	•	•	•
	WRK			•	•	•	•	•
	SAT	•	•	•	•	•		
	SUN/HOL			•	•		•	
	PL	•		•	•	•	•	•
d-2	Tmax		•				•	•
	Tmin		•	•		•	•	
	WRK	•	•		•	•		•
	SAT	•	•			•	•	•
	SUN/HOL					•	•	
	PL	•	•	•	•	•		•
d-1	Tmax							
	Tmin	•	•	•	•		•	•
	WRK		•	•	•	•		•
	SAT	•			•	•	•	•
	SUN/HOL		•		•		•	
	PL	•	•	•	•	•	•	
d	Tmax	•	•	•	•	•	•	•
	Tmin	•	•	•	•	•	•	•
	WRK		•	•	•	•	•	•
	SAT			•		•		
	SUN/HOL							
	PL	•	•	•	•	•	•	•
d+h	ETmax	•	•	•	•	•	•	•
	ETmin	•	•	•	•	•	•	•
	WRK	•	•	•	•	•	•	•
	SAT							
	SUN/HOL							
Number of Used Inputs		25	29	29	33	32	28	21
Number of Layers		3	3	3	3	2	2	3

Table 7. Performance summary of the two approaches for next week peak load forecasting using both ideal and noisy temperature forecasts.



Forecasting day, h	MAPE, %			
	Ideal Temperature Forecasts		Realistic Temperature Forecasts	
	Iterated use of the (d+1) model	Using dedicated (d+h) models	Iterated use of the (d+1) model	Using dedicated (d+h) models
1	2.50		2.76	
2	2.93	2.87	2.81	3.09
3	2.98	2.98	3.35	3.31
4	3.02	3.03	3.48	3.41
5	3.08	3.17	3.59	3.47
6	3.10	3.10	3.59	3.66
7	3.10	3.08	3.67	3.71
Average	2.96	2.96	3.32	3.34

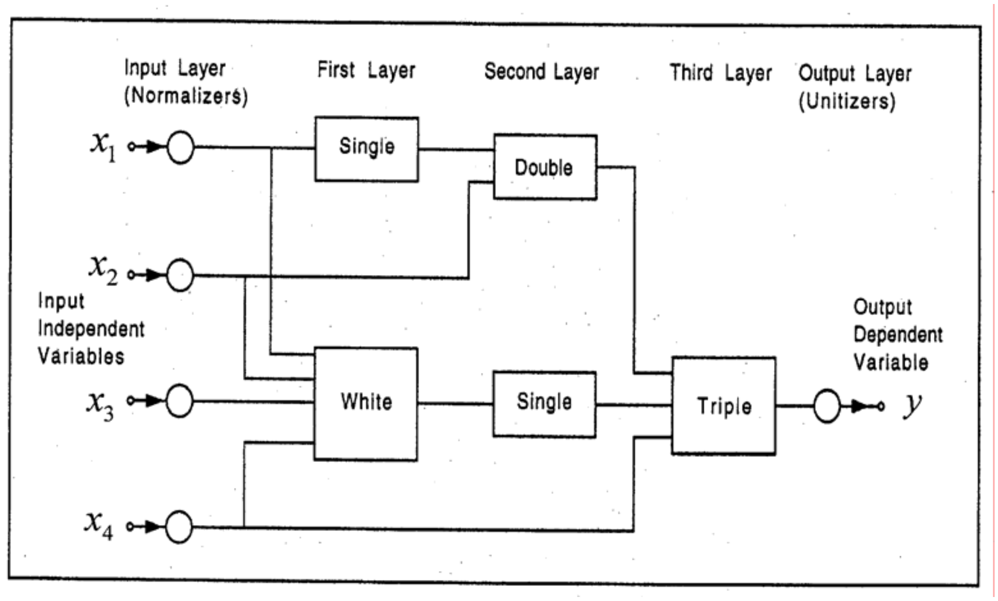


Fig. 1. A typical AIM abductive network model showing various types of functional elements.

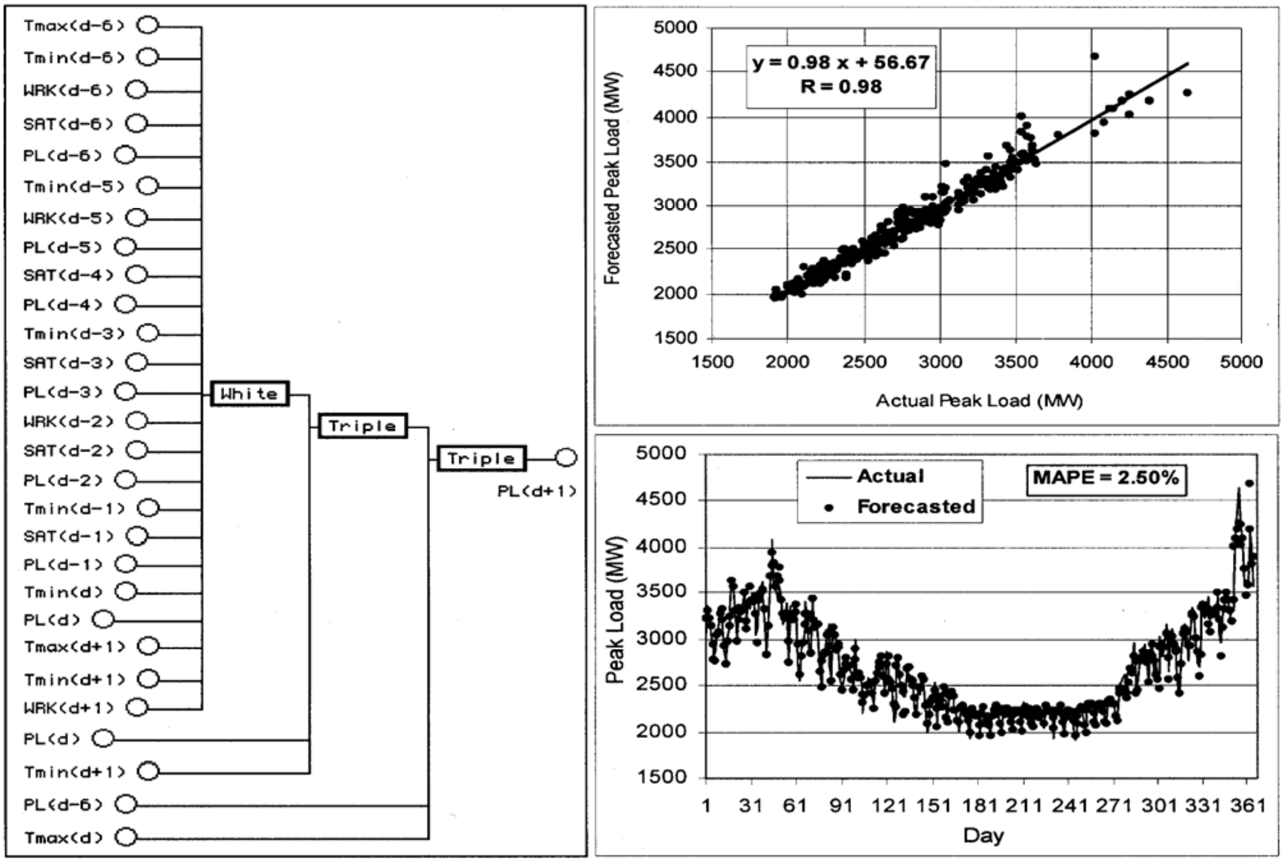
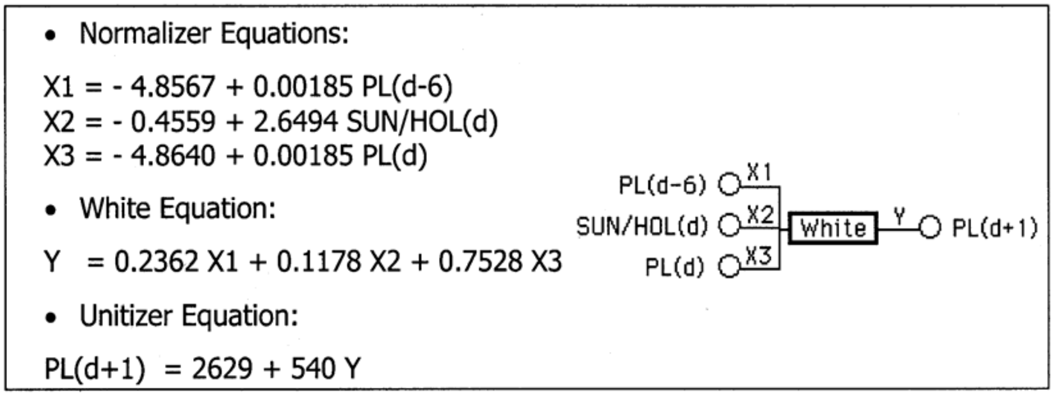


Fig. 2. Structure and performance over the evaluation year for the default next-day peak load forecasting model with *CPM* = 1.



- Normalizer Equations:

$$X1 = -4.8567 + 0.00185\ PL(d-6)$$
$$X2 = -0.4559 + 2.6494\ SUN/HOL(d)$$
$$X3 = -4.8640 + 0.00185\ PL(d)$$

- White Equation:

$$Y = 0.2362\ X1 + 0.1178\ X2 + 0.7528\ X3$$

- Unitizer Equation:

$$PL(d+1) = 2629 + 540\ Y$$

Fig. 3. Structure and equations for the simplified next-day peak load forecasting model using *CPM* = 5.