

Short Term Hourly Load Forecasting Using Abductive Networks

R. E. Abdel-Aal

Center for Applied Physical Sciences, Research Institute,
King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

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 6th year, the models give an overall mean absolute percentage error (MAPE) of 2.67%. Next-hour models utilizing load data up to the preceding hour give a MAPE of 1.14%, outperforming neural network models for the same utility data. Two methods are described for dealing with the load growth trend. Effects of varying model complexity are investigated and proposals made for further improving forecasting performance.

Index Terms: Abductive networks, Neural networks, Neural network applications, Load forecasting, Forecasting, Modeling, Power systems, Power system planning, Artificial intelligence.

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

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, hydro-thermal coordination, short-term maintenance, fuel allocation, power interchange, transaction evaluation, as well as network analysis functions, such as dispatcher power flow and optimal power flow. 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, are time consuming, 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 which 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 actually 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. Feed-forward 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 over-fitting and ensure 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 results to human experts [11].

To overcome such limitations, we propose using abductive networks [12] as an alternative machine learning approach to electric load forecasting. We have previously used this approach to model and forecast the monthly domestic energy consumption [13], and in forecasting the minimum and maximum daily temperatures [14,15]. Compared to neural networks, 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 structure [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 over-fitting by using a proven regularization criterion based on penalizing model complexity [12], without requiring a dedicated validation data set during training, as is the case with many neural network paradigms.

Following a brief description of 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-hour 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.

II. AIM ABDUCTIVE NETWORKS

AIM (abductory inductive mechanism) [16] 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 group method of data handling (GMDH) algorithm [17], 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 misjudgements. 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. The algorithm selects polynomial relationships and input combinations that minimize the prediction error in each phase. AIM builds networks of various types of polynomial functional elements, based on prediction performance. 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 feed-forward abductive networks of functional elements (nodes) [16], 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.:

$$\text{"White" Output} = W_0 + W_1X_1 + W_2X_2 + W_3X_3 + \dots + W_nX_n \quad (1)$$

where X_1, X_2, \dots, X_n are the inputs to the element and W_0, W_1, \dots, 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,

$$\text{"Double" Output} = W_0 + W_1X_1 + W_2X_2 + W_3X_1^2 + W_4X_2^2 + W_5X_1X_2 + W_6X_1^3 + W_7X_2^3 \quad (2)$$

The database of input-output solved examples is split into a training set and an evaluation set. AIM uses the training set to synthesize the model network layer by layer until no further improvement in performance is possible or a preset limit on the number of layers is reached. Within each layer, every element is computed and its performance scored for all combinations of allowed inputs. The best network structure, element types and coefficients, and connectivity are all determined automatically by minimizing the predicted squared error (PSE) criterion [18], which eliminates the problem of determining when to stop training in neural networks. This criterion selects the most accurate model that does not overfit the training data to strike a balance between the accuracy of the model in representing the training data and its generality which allows it to fit yet unseen future data. The user may optionally control this trade-off between accuracy and generality using the complexity penalty multiplier (CPM) parameter [16]. 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.

III. 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 [19]. 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 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 estimated the mean hourly load 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). This was performed by multiplying the hourly value by the ratio between the mean load for 1989 and that for the relevant year. Values of this normalizing factor are given in the second column from right in Table 1. For the evaluation year, we use the estimated mean because in practice no actual data would be available for that year. With the second approach, no normalization of the hourly load data is necessary, and load growth was represented as an additional model input in the form of the ratio between the mean load for the relevant year and that for the first year in the data set (1985). Values of this normalizing factor are given in the last column of Table 1. This approach considerably reduces the data pre-processing work required prior to model development.

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 ($d-1$). 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 ($d-1$) ($L_1, L_2, L_3, \dots, L_{24}$), the measured minimum (T_{min}) and maximum (T_{max}) air temperatures on day ($d-1$), the forecasted minimum (ET_{min}) and maximum (ET_{max}) 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). T_{min} and T_{max} 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 ($h=1,2,\dots,24$) includes 32 input variables and takes the following form:

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, L3, ..., L24	Tmin, Tmax	ETmin, ETmax	WRK, SUN, SAT, HOLI	L(d,h)

Prior to training and evaluation, all hourly load data for inputs and output were normalized by multiplying by the normalization factors relative to 1989 mean, shown in Table 1. In model evaluation, the effect of normalization was removed by dividing forecasted data for the year 1990 by the estimated normalization factor for that year before comparison with actual load data for that year. To avoid the effect of discontinuities at year boundaries (input loads for day (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 single-element nonlinear 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 values 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 4-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 Parkinson's correlation coefficient as 0.98, and the time series plot shows the mean absolute percentage error (MAPE) as 2.40%. Table 2 summarizes the model structure for all the 24 models, listing the model inputs selected and the number of layers and elements. Models for the first two hours discard temperature and day type information, relying only on the load 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. The second column in Table 3 lists the MAPE values for all hours, giving the overall value for the evaluation year as 2.67%.

Values in the third column of the table 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 1 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. 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 8 August to 3 September 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.

We have investigated the effect of simulated errors in the ideal forecasted extreme temperature values ET_{min} and ET_{max} for the load-forecasting day. As seen from Table 2, the model for hour 12 is an example of 13 load forecasters that use both ET_{min} and ET_{max} , and would therefore be affected most by such errors. There are nine other models, e.g. for hours 3 and 18, that use either ET_{min} or ET_{max} only, and are therefore 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 σ 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 $\sigma = 1^\circ$ F and to 2.60% for $\sigma = 2^\circ$ 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 4 shows the structure and performance of the resulting more complex model with CPM = 0.2 and the simpler model with CPM = 5, in comparison with the default model with 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, 4-layer nonlinear model at CPM = 0.2 to a simple 2-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 train times.

V. NEXT-HOUR LOAD FORECASTERS

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,L3,...,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 extreme temperatures 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 of model inputs limited to 50 for the AIM version used, we had only 3 inputs left to represent temperature and day type information. Temperature was represented using the average temperature T_a on day (d-1) and the forecasted average temperature ET_a for day (d). Again, ET_a was taken as actual T_a 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 ($h=2,3,\dots,24$) takes the following form:

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)

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 5 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 3-input, 1-element as compared to a 7-input 4-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 hour featuring in all models. The middle column in Table 6 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 3 months of the same utility data (working days only) was reported to give a MAPE of 1.41% when evaluated on 22 days [20]. Inputs to the neural network included measured loads and temperatures at the two immediately preceding hours, estimated

temperature at the forecasting hour, and an hour index. Table 7 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 6 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 (d-1), with the load forecasted for hour (i) being fed, among other required inputs, to the next-hour model for hour (i+1). As expected, performance of this type of next-day forecasting is inferior to that given in Table 3, with the overall average MAPE being 4.87% as compared to 2.67%. This is because next-hour models are heavily dependant on recent hourly loads on the forecasting day. With these values being forecasted and not measured, forecasting errors accumulate. As seen from Table 6, next-hour models almost totally ignore temperature and day type information, and therefore should not be expected to form a basis for accurate next-day forecasting which depends heavily on such parameters as indicated in Table 2.

Four of the next-hour models in Table 5 (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 8 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 7 times those for the synthesized wire models.

VI. CONCLUSIONS

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 reported in the literature. 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 applied to other areas including peak load forecasting.

ACKNOWLEDGEMENTS

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

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 ANNPS*, 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 PICA 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 Systems*, 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 Systems*, 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 Systems*, vol. 16, pp. 44-55, Feb.

2001.

[8] A. P. Alves da Silva, U. P. Rodrigues, A. J. Rocha Reis, and L. S. Moulin, NeuroDem - a neural network based short term demand forecaster, Presented at the IEEE Power Tech. Conf., Porto, Portugal, 2001.

[9] W. Charytoniuk and M. S. Chen, "Neural network design for short-term load forecasting," in *Proc. DRPT, 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. Society Winter Meeting*, 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 on Soft Compt. Ind. Appl.*, 2001, pp. 25-27.

[12] G. J. Montgomery and K.C. Drake, "Abductive networks," in *Proc. SPIE Applications of Artificial Neural Networks Conf.*, 1990, pp. 56-64.

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

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

[15] R. E. Abdel-Aal and M.A. Elhadidy, "Modeling and forecasting the maximum temperature using abductive machine learning," *Weather and Forecasting*, vol. 10, pp.310-325, June 1995.

[16] AbTech Corporation, Charlottesville, VA, USA, *AIM User's Manual*, 1990.

[17] 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.

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

[19] <http://www.ee.washington.edu/class/559/2002spr/>

[20] 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 Systems, vol. 6, pp. 442-449, May 1991.

Table 1. Summary of the 6-year load data showing information on the year-to-year growth and the factors used by the two methods adopted for dealing with the load growth trend.

Year	Total Annual Load, MWH	Mean Hourly Load, MW	Annual Load Growth (year-to-year)	Factor for Normalizing to 1989 Mean (Method 1)	Factor for Normalizing to 1985 Mean (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 2. 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

Table 3. Performance of the next-day load forecasting models over the evaluation year for two methods of accounting for the annual trend of load growth.

Forecasting hour, h	MAPE, %	
	Trend removed by normalizing data	Trend represented as a separate input variable
1	1.14	0.95
2	1.60	1.32
3	1.34	1.23
4	1.58	1.47
5	2.07	1.72
6	2.68	2.49
7	3.43	3.34
8	3.40	3.26
9	2.59	2.37
10	2.06	1.86
11	2.71	2.14
12	2.40	2.32
13	2.47	2.59
14	2.66	4.64
15	3.02	3.30
16	3.18	3.48
17	3.54	3.46
18	3.56	3.57
19	3.31	3.29
20	3.65	3.29
21	3.29	3.11
22	3.05	3.01
23	2.65	2.74
24	2.84	2.84
Average	2.67	2.66

Table 4. Effect of the CPM parameter on the complexity and performance of next-day load forecasting models for hour 12.

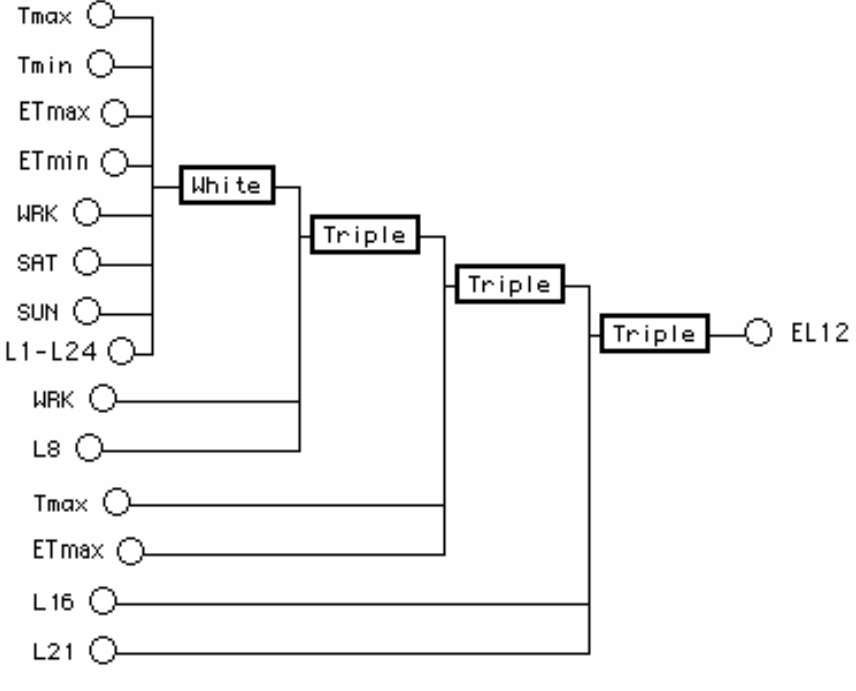
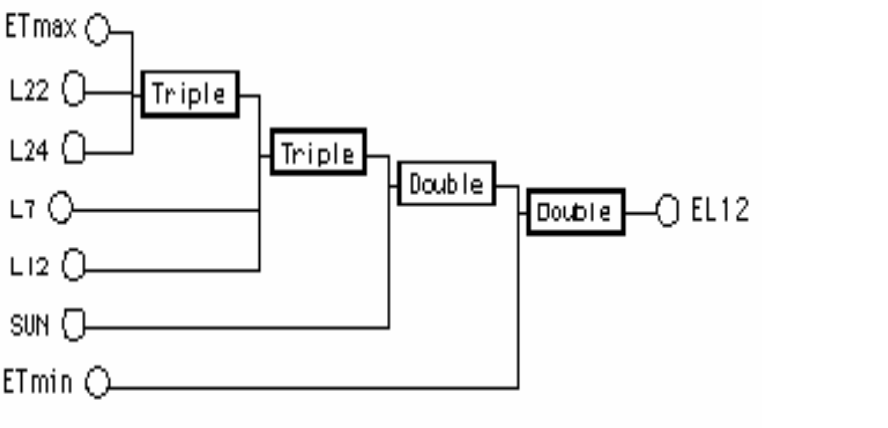
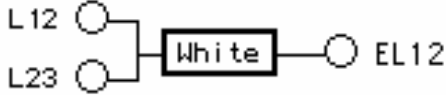
CPM	Model Structure	Relative Training Time	MAPE, %
0.2		1.20	2.21
1		1.00	2.40
5		0.58	3.69

Table 5. Summary of the abductive network models for the 24 next-hour load forecasters.

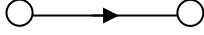
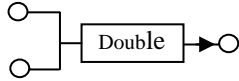
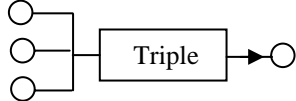
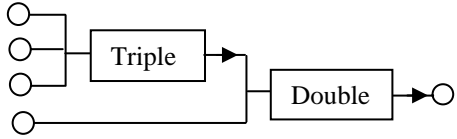
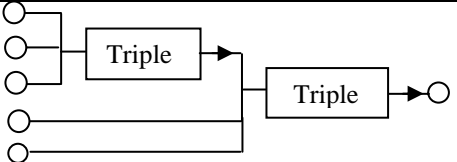
Day (d) Forecasting Hour	Model Input(s)			Load Time Lags Selected	Model Structure
	Day (d-1) Load at Hours:	Day (d) Load at Hours:	Other		
2		1		1	
3		2		1	
4		3		1	
8		7		1	
7		6,5		1,2	
1	24,20,3			1,5,22	
5	10	4	WRK	1,19	
6	22	5,1		1,5,8	
9		8,6,4		1,3,5	
10	18	9,8		1,2,16	
11	16	10,9		1,2,19	
12		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 6. Performance of the next-hour load forecasting models over the evaluation year (column 2), and of the next-day load forecasts obtained using such models 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

Table 7. Summary of the MAPE error histograms for all forecasting hours over the evaluation year for both next-day and next-hour hourly forecasting models.

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 8. Comparison of the performance of the four next-hour ‘wire’ models in Table 5 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

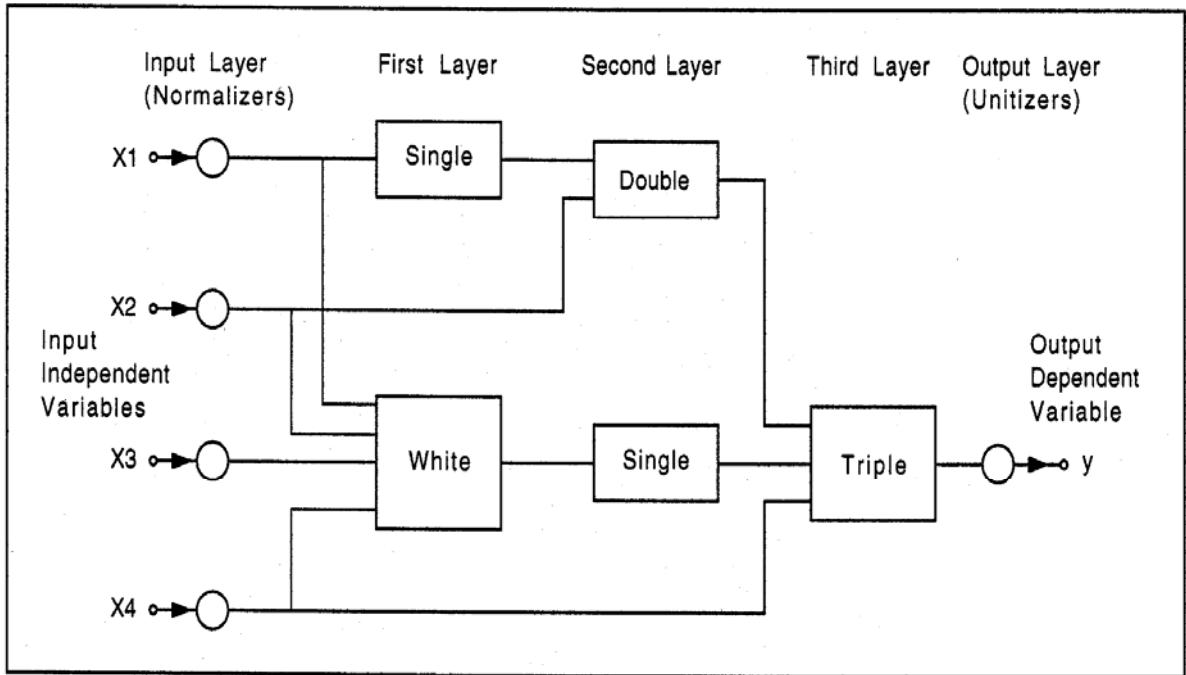
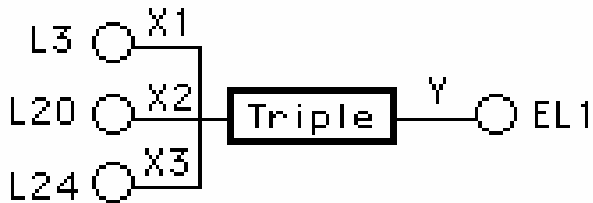


Fig. 1. A typical AIM abductive network model showing various types of functional elements.

- Normalizer Equations:

$$\begin{aligned}
 X1 &= -4.52 + 0.00303 L3 \\
 X2 &= -4.66 + 0.00295 L20 \\
 X3 &= -5.61 + 0.00315 L24
 \end{aligned}$$



- Triple Equation:

$$Y = 0.125 X1 + 0.868 X3 - 0.115 X1 X2 + 0.0506 X1 X3 + 0.0582 X2 X3$$

- Unitizer Equation:

$$EL1 = 1600 + 312 Y$$

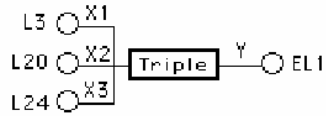
Fig. 2. Structure and equations for the next-day load forecasting model for hour 1.

- 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:

$$EL1 = 1600 + 312 Y$$

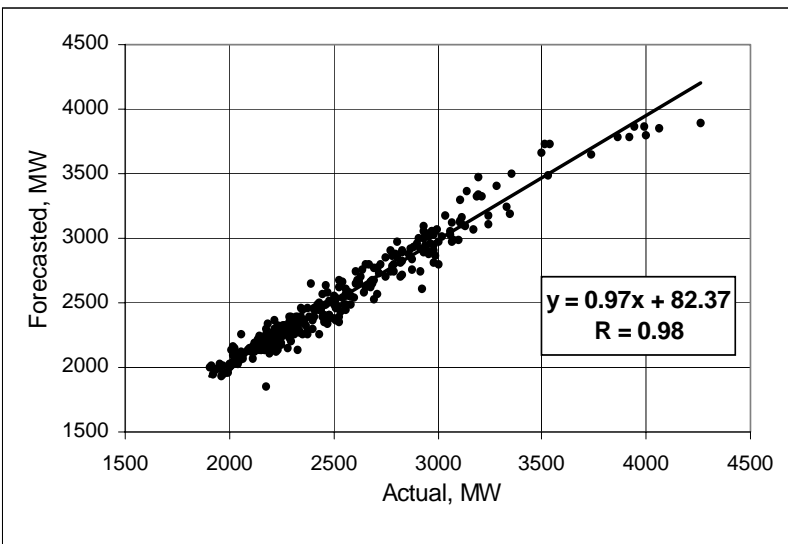
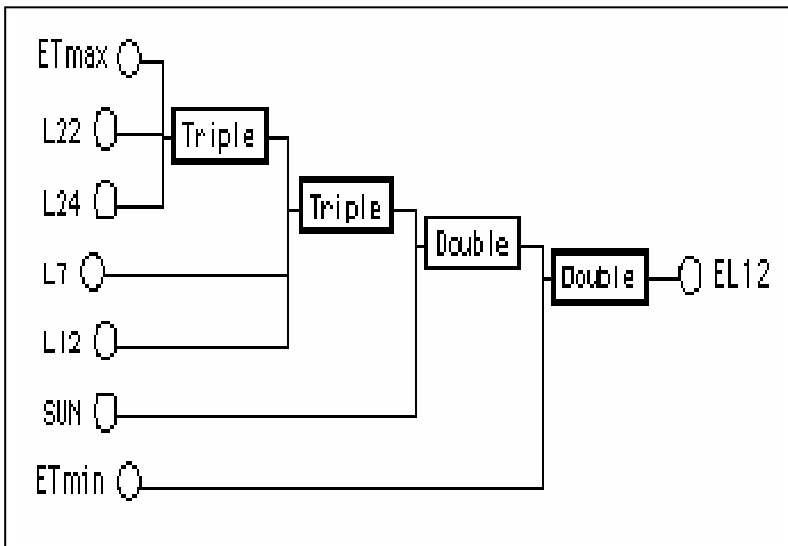
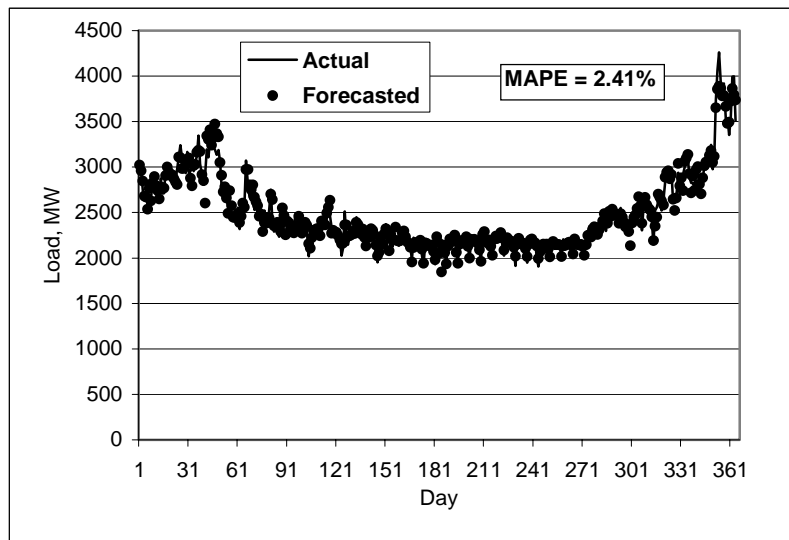


Fig. 3. Structure and performance of the next-day load forecasting model for hour 12 over the evaluation year.

MAPE Error	% population of forecasting hours over evaluation year	
	Next-day models	Next-hour models
$\leq 1\%$	29%	57%
$\leq 3\%$	68%	95%
$\geq 6\%$	9%	0.25%



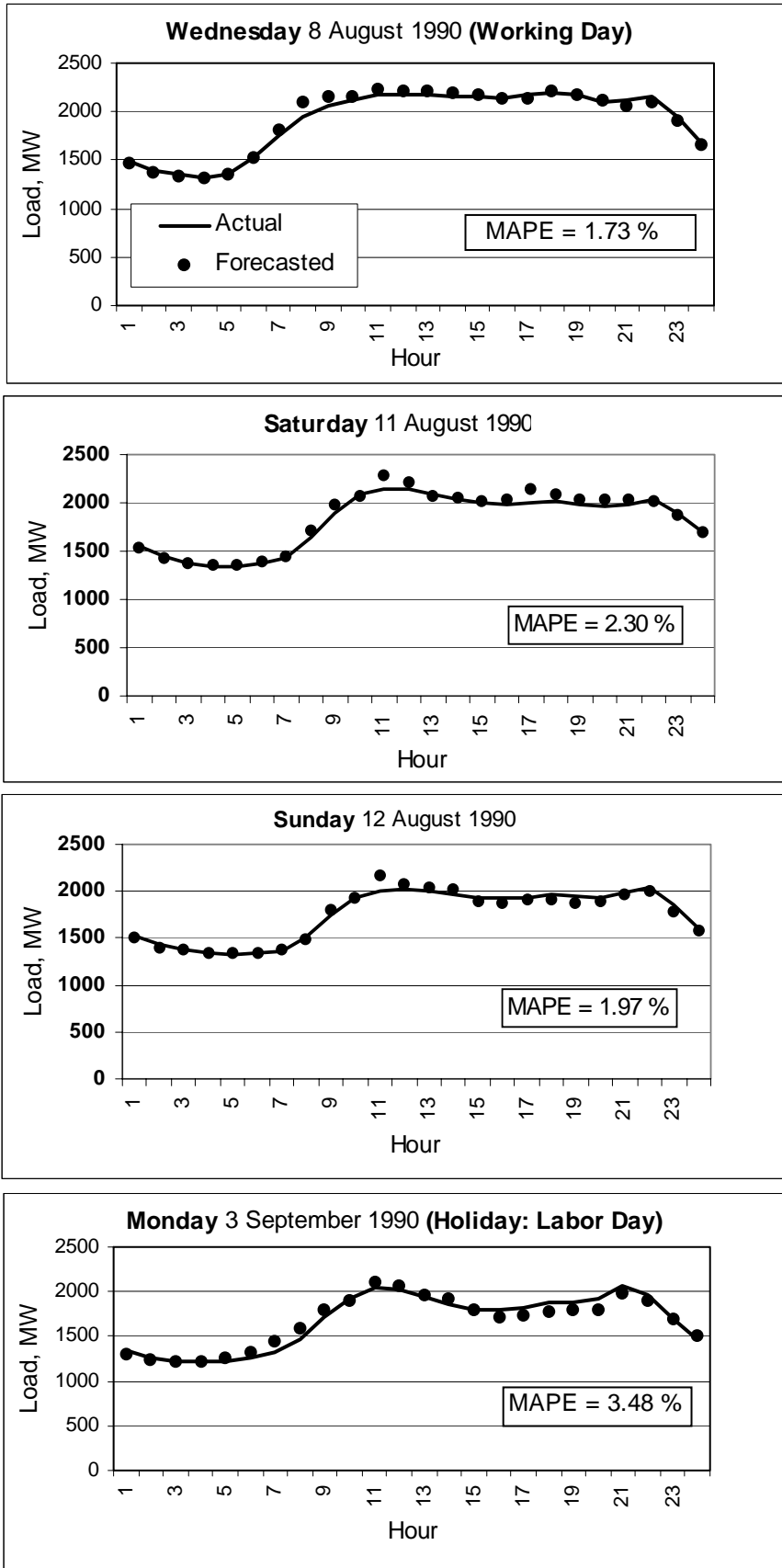


Fig. 4. Performance of the 24 next-day hourly forecasters on four typical days representing a working day, a Saturday, a Sunday, and a public holiday during the summer of the evaluation year.

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)

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)

Forecasting hour, h	MAPE, %	
	Trend removed by normalizing data (Method 1)	Trend represented as a separate model input (Method 2)
1	1.14	0.95
2	1.60	1.32
3	1.34	1.23
4	1.58	1.47
5	2.07	1.72
6	2.68	2.49
7	3.43	3.34
8	3.40	3.26
9	2.59	2.37
10	2.06	1.86
11	2.71	2.14
12	2.40	2.32
13	2.47	2.59
14	2.66	4.64
15	3.02	3.30
16	3.18	3.48
17	3.54	3.46
18	3.56	3.57
19	3.31	3.29
20	3.65	3.29
21	3.29	3.11
22	3.05	3.01
23	2.65	2.74
24	2.84	2.84
Average	2.67	2.66

