Improving Electric Load Forecasts Using Network Committees

R. E. Abdel-Aal
Department of Physics,
King Fahd University of Petroleum and Minerals, Dhahran 31261, 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

Accurate daily peak load forecasts are important for secure and profitable operation of modern power utilities, with deregulation and competition demanding ever-increasing accuracies. Machine learning techniques including neural and abductive networks have been used for this purpose. Network committees have been proposed for improving regression and classification accuracy in many disciplines, but is yet to be widely applied to load forecasting. This paper presents a formal approach to apply the technique using historical load and temperature data spanning multiple years, with individual committee members trained on different years. Correlation among data for successive years is investigated and methods to enhance independence between member models for improving committee performance are described. Both neural and abductive networks implementations are presented and compared. An abductive network 3-member committee was developed on data for 3 successive years and evaluated on the fourth year. Compared to a monolithic model trained on the same full 3-year data, the committee reduces the mean absolute percentage error from 2.52% to 2.19%. The corresponding reduction in the mean of the absolute error from 70 MW to 61 MW is statistically significant at the 95% confidence level.

Keywords: Machine learning, Neural networks, Abductive networks, GMDH, Network Committee, Network ensemble, Modeling, Forecasting, Load forecasting, Daily peak load, Power system planning.

1 Introduction

Accurate 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. STLF forecasting covers the daily peak load, total daily energy, and daily load curve as a series of 24 hourly forecasted loads. 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. 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. 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. Neural networks of various architectures and learning paradigms have been widely used for STLF forecasting, e.g. [2-6]. Polynomial or abductive networks [7] based on the self-organizing group method of data handling (GMDH) [8] have also been utilized for this purpose [9,10]. Historical load and weather data often span a number of successive years, and conventionally a single model trained on multiple year data is used. In this case, the problem of dealing with the annual trend due to load growth should be addressed. This paper proposes a modular approach in which a committee (ensemble) is formed of individual networks, each trained on the data for one year. Combining the outputs of such networks can improve the accuracy and reliability of the forecasts beyond those of individual networks and of the single monolithic model developed using the full data for all years.

Network committees have been proposed for improving accuracy and reliability in many classification and regression applications, including medical diagnosis [11], image recognition [12], analysis of seismic data [13], financial forecasting [14], speech recognition [15], and prediction of traffic flow in telecommunication networks [16]. However, the technique is yet to be formally and widely applied to load forecasting. It should be noted that the committee approach is different from modular solutions reported widely in the load forecasting literature where different neural network modules are used to model various aspects of load variations, e.g. weekly, daily, and hourly, and their outputs combined to produce the final forecast [17]. With the committee approach all member modules of the committee tackle the same forecasting problem, albeit from different perspectives arising, for example, from using different training data or adopting different learning algorithms. Early thoughts on network ensembling for load forecasting, though not identified as such at the time, were described by Matsumoto et. al. [18]. The authors compared two approaches for forecasting summer peak load for one year from summer data on the previous four years: using four separate models each trained on the data for one year, and using a single model trained on the collective data for all four years. They described methods for selecting the best among the four models based on the norm distance between the input vector for the forecasting day and those of the corresponding training days. A dedicated neural network was also developed for predicting the best individual module. In the final analysis, however, the authors concluded that the single model developed on the full data provides the best results. In [19], Drezga and Rahman used simple averaging to combine the outputs of two local neural networks trained on different data to improve the accuracy of next-hour load forecasts.

This paper proposes a formal approach to building network committees for forecasting the daily peak load using multiple-year data. The approach is demonstrated using both neural and abductive networks. Statistically significant improvements in forecasting accuracy compared to conventional single-model approaches are reported. Section 2 gives an overview of the network committee technique. Section 3 describes the load and temperature data set used, presents the monolithic and committee approaches adopted for forecasting the daily peak load, and introduces the abductive network modeling technique employed. Section 4 presents results obtained using both neural and abductive network ensembles.

2 Network Committees

In the network committee approach, *n* networks are trained to solve the same problem independently. During prediction the networks operate simultaneously on the input data and their outputs are combined to produce the final committee output, see Fig. 1. The output combination module in Fig. 1 often performs simple functions on the outputs of individual members, such as majority voting for classification and simple/weighted averaging for regression, without involving the input vectors of attributes [20]. Alternatively, a gating network may use the input vectors to determine the optimum weighting factors used with individual member outputs for each case to be classified [21]. In the stacked generalization approach, the combiner takes the form of another higher-level network trained on the outputs of individual members to generate the committee classification output [22].

Simple averaging provides a simple and effective method of combining continuous outputs from individual committee members using the relationship [20]:

$$z_c = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{1}$$

where y_i; i = 1,2,…,n are the outputs from committee members and z_c is the combined committee output, see Fig. 1.

The above relationship assumes that outputs from all members are of equal accuracy. In practice, some outputs may have greater certainty than others, and individual outputs may be weighted to reflect this fact [20]. The committee output is then the weighted sum of the outputs of all members:

$$z_c = \sum_{i=1}^{n} \alpha_i y_i \ , \tag{2}$$

where $\sum_{i=1}^{n} \alpha_i = 1$. (3)

As a static measure, the certainty c_i of the output from member i can be expressed as the inverse of the variance of the error (σ_i^2) by that member over its training set [23]:

$$c_i = \frac{1}{\sigma_i^2}\ , \tag{4}$$

and the weight α_i is then determined by:

$$\alpha_i = \frac{c_i}{\sum_{j=1}^{n} c_j}\ ,$$

(5)

which satisfies the condition on the weights in Eqn. (3).

When member networks are independent, the resulting diversity in the decision making process is expected to boost generalization performance, thus improving the accuracy, robustness, and reliability of predictions. Obviously, combining the outputs of several identical networks produces no gain, and improvement is expected only when members err in different ways so that errors may cancel out [24]. It can be shown [25], that the mean squared error in the averaged committee output contains as a component the covariance of the outputs of individual committee members, therefore individual members should ideally be uncorrelated or even negatively correlated. Krogh and Vedelsby [26] have shown that the committee error can be expressed as two terms, one measuring the average generalization error of individual members and the other measuring the diversity or disagreement among the members. An ideal committee would therefore consist of highly accurate networks that disagree as much as possible. In the 'committee of experts' approach, members are developed using different machine learning techniques that adopt different ways to build decision boundaries for the classification problem at hand, such as neural networks, nearest neighbor classifiers, classification and regression trees (CART), etc. This allows training adequate individual models on the full training dataset available while ensuring a good degree of diversity among them. However, in many situations a committee is restricted to use only one machine learning technology.

Neural networks provide a wide range of available architectures (e.g. multi-layer perceptron (MLP) and radial basis function (RBF)), learning algorithms (e.g. back propagation and simulated annealing), and parameters that can be varied during training (e.g. network topology, neuron transfer functions, initial random weights, learning rate, momentum, and stopping criteria). This allows many possibilities for constructing individual committee members that should be reasonably independent. Enhancing diversity among individual members of neural network committees has been attempted through training on different parts of the dataset [27], on different input features [28], or using different network architectures [29], different learning paradigms [12], or different training parameters [30]. Techniques for automatically enhancing negative correlation between individual committee members during training have been described [31]. Resampling methods, such as bootstrap sampling, have been used to increase independence among training subsets for individual committee members. They form the basis for the bagging (bootstrap aggregating) [32] and boosting [33] ensembling methods which were originally described for classification with decision trees. Both techniques attempt to approximate the ideal averaged model that does not depend on the training set used, through aggregating the outputs of many models trained on different bootstrap subsamples drawn randomly with replacement from the available training set. Each of the subsamples has the same size as the full training set. Bagging simply builds different models using the subsamples generated, and combines their outputs using equal weights. Boosting, however, generates different models sequentially. It modifies the weights of training examples used to build a new model based on the performance of previous models to make the new model concentrate more on training examples that were previously misclassified. Outputs of the different models are combined using weights that depend on model performance. Various approches have been proposed for modifying bagging and boosting for use on regression problems [34]. A modification of the AdaBoost boosting algorithm was used with neural networks to predict drug dissolution profiles [35]. With such modifications, the bagging and boosting ensembling techniques may prove useful in improving the accuracy of load forecasts.

Abductive networks offer several advantages over neural networks [9], including simpler and more automated model synthesis, automatic selection of significant model inputs, automatic stopping criterion that does not require a separate validation data set, and more transparent analytical model relationships. Although neural network committees have been reported for many applications, there appears to be no mention of GMDH-based abductive (or polynomial) network committees in the literature. Due to the self-organizing and self-stopping nature of such networks, the absence of initial random weights, and the little room for user intervention during training, there is less room to introduce diversity in the models that can be synthesized using the same training data. This paper considers both neural and abductive network committees for improving the accuracy of daily peak load forecasts.

3 Data and Methodology

3.1. The Data Set

The data set used for this study 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 [36]. 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. We used the data for 3 years (1987-1989) for model synthesis and that of the following year (1990) for model evaluation. For the evaluation year, 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. The second column from right in the table shows values of the trend inputs for each of the three years (1987 to 1989) used to develop the model trained on collective data for all three years, see Section 3.2. Let the mean hourly load for year i be M_i, the trend input r_i for that year relative to the first training year is given by:

$$r_i = M_i / M_{1987} \quad ; i = 1987, 1988, 1989, 1990 \tag{6}$$

The last column in the table lists three scaling factors, one for each of the training years (1987 to 1989), to be used for implementing the network committee solution, see Section 3.3. The scaling factor for year *i* is given by:

$$s_i = M_{1990} / M_i \quad ; i = 1987, 1988, 1989 \tag{7}$$

3.2. Monolithic Models Using the Collective Data

To compare committee forecasts with those obtained using a single model utilizing all the 3-year data, neural and abductive network models were developed for forecasting the peak load (PL) on the next day (d+1) in terms of data available at the end of day (d) regarding the peak load, measured extreme temperatures, and day type for the seven days preceding the forecasting day, i.e. days d-6, d-5, …, d-1, and d, as well as forecasted extreme temperatures and day type information for the forecasting day (d+1). The models were trained using the full data for the three years (1987-1989) preceding the evaluation year 1990. To account for annual load growth, a model input indicating load growth trend is used. For the various years, the input takes the r_i values (i = 1987, 1988, 1989, 1990), defined in Eqn. (6) and listed in Table 1. In addition to the trend input, the models use 47 load, temperature, and day type variables. For each of the seven preceding days, six variables are used, including the peak load, measured daily maximum (Tmax) and minimum (Tmin) air temperatures, and day type information 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). Data on the forecasting day (d+1) uses only five variables: forecasted minimum and maximum air temperatures (ETmax and ETmin, respectively), in addition to the three day type variables. Due to the absence of next-day forecasts for the maximum and minimum air temperatures in the data set, actual temperature values were used instead, which would be the case with ideal temperature forecasts. Table 2 lists the input and output variables used for training the model. The first seven days of each year were used as preceding days, and therefore the first forecasting day was 8 January of each year. This gave 1075 training records (358 in each of 1987 and 1989, and 359 in 1988 being a leap year). It was possible to leave out only the first seven days in 1987, but we opted for the scheme described above as it gave the same number of training records used by three individual models trained separately on each of the three years (1987 to 1989). Equal number of training records in the two cases allows fair comparison between the committee-based solution employing three yearly models and the monolithic solution using a single 3-year model. The monolithic models are evaluated on 358 records of the evaluation year 1990.

3.3. Network Committee Modeling

For the committee forecasting approach, three separate models for forecasting next-day peak load are developed, each trained on data for one of the years 1987, 1988, and 1989 and the resulting committee is evaluated on data for 1990. Input variables used for training individual models are identical to those listed in Table 2, with the exclusion of the trend input which would not be required with the training data for each model spanning only one year. Therefore committee member models have only 47 inputs. Using the same approach described above for selecting the training data for the monolithic model, individual member models trained on data for the years 1987, 1988, and 1989 use 358, 359, and 358 training records, respectively. The total number of data records used to train the whole committee is identical to that used by the monolithic model. The committee is evaluated on 358 records of the evaluation year 1990. Since individual models are developed using ‘raw’ load data for their respective years, trend adjustments are required during evaluation in order to account for the annual load growth between the respective years used to train the individual models and the evaluation year. Fig. 2 illustrates the scheme adopted when the three models are used in a committee for forecasting the load for the evaluation year 1990. While temperature and day type data in the evaluation records are applied directly (in parallel) to the models, load data must first be normalized to that of the year used to train the model before inputting to the individual models. This is achieved through dividing the evaluation input values for the load by the corresponding scaling factor s_i defined in Eqn. (7) and listed in Table 1. Predicted load outputs from individual models should then be denormalized to the evaluation year through multiplying by the same scaling factor. Denormalized output values can then be combined to form the committee predicted load, which is compared to actual known load values for 1990 for evaluation.

3.4. Multi-Layer Perceptron (MPL), Back Propagation, Neural Networks

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

$$E = \frac{1}{2}\left[\sum_{p}\sum_{k}\left|t_{kp} - O_{kp}\right|^{2}\right] \tag{8}$$

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

$$O_j = f(\sum_{i} W_{ij} O_i), \tag{9}$$

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

$$f(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

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

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}}, \tag{11}$$

where η is a constant that determines the learning rate. To improve convergence characteristics, weight changes are also related to changes introduced in the previous iteration. At the *n*th iteration, the change in W_{ij} for the link from neuron *i* to neuron *j* is given by [37]:

$$\Delta W_{ij}(n) = \varepsilon \delta_j O_i + \alpha \Delta W_{ij}(n-1), \tag{12}$$

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

$$\delta_j = (t_j - O_j)O_j(1 - O_j) \tag{13}$$

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

$$\delta_j = O_j(1 - O_j)\sum_k \delta_k W_{jk} \ , \tag{14}$$

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

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

3.5. GMDH and AIM Abductive Networks

AIM (abductory inductive mechanism) [38] 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. Automation of model synthesis not only lessens the burden on the analyst but also safeguards the model generated against influence 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 data base 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 [8]:

$$y = A + B\,x_i + C\,x_j + D\,x_i^2 + E\,x_j^2 + F\,x_i\,x_j \qquad (15)$$

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 dataset, where:

$$r_k^{\,2} = \sum_{\ell=1}^{n_s} (y_\ell - z_{k\ell})^2 \Bigg/ \sum_{\ell=1}^{n_s} y_\ell^{\,2}; \quad k = 1,2,...,m(m-1)/2 \qquad (16)$$

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 model 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 on 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 [39]. 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 dataset thus eliminating the need for a dedicated selection set. The adaptive learning network (ALN) approach, AIM being an example, uses the predicted squared error (PSE) criterion [39] for selection and stopping to avoid model overfitting, thus solving 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* as:

$$PSE = FSE + CPM\,(2K\big/N)\sigma_p{}^2 \tag{17}$$

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* [39]. 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 considerably reduces the learning/development time and effort. The models take the form of layered feed-forward abductive networks of functional elements (nodes) [38], see Fig. 3. 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. 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.

$$\text{"White" Output} = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + ... + w_nx_n \tag{18}$$

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,

$$\text{"Double" Output} = w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + w_6x_1^3 + w_7x_2^3 \tag{19}$$

4 Results

4.1 3-year Monolithic Models

Both neural and abductive Monolithic models were developed through training on the same collective 3-year data, to allow comparison with the committee approach in both cases. Neural network model were synthesized using the PathFinder neural network software for Windows. 20% of the training data were reserved for cross validation. The 48-6-1 neural model had one hidden layer containing 6 neurons and used a sigmoid transfer function for both the hidden and output layers. Evaluated on the 1990 data, the model gives a mean absolute percentage error (MAPE) of 2.61%. Fig. 4 shows the corresponding AIM abductive network model, which gives a lower MAPE error of 2.52%. The 4-layer, 4-element abductive model uses only 8 inputs selected automatically from the 48 inputs available during training. Compared to the neural network model, this reduction in dimensionality simplifies the resulting model, reduces the amount of historical data required to implement it, and avoids the effects of noise and errors that may be associated with the 40 unused inputs. The model is much more transparent than the corresponding neural model, readily giving tomorrow's peak load forecast as a function of:

- Peak loads for today and on the same day as the forecasting day a week ago.
- Both extreme temperatures forecasted for tomorrow.
- The minimum temperature measured on the on the same day as the forecasting day a week ago.
- Whether tomorrow is a working day or not.
- Whether yesterday was a Saturday or not.
- Annual load growth trend.

The model is described by eight normalizer and one unitizer linear equations, in addition to equations for the three Triple and one Double polynomial functional elements. Fig. 5 shows time series plots of the actual peak loads for the 358 evaluation days during 1990 and the corresponding loads forecasted using both the neural and abductive models. Table 3 shows details of performance comparisons for both models. The table lists the mean and standard deviation (SD) for both the absolute percentage error (APE), %, and the absolute error (AE), MW (in parenthesis), between the actual and forecasted loads, together with the maximum APE value. Shown also are percentages of the evaluation population having APE values ≤ 1%, ≤ 3%, and ≥ 6%. Fig. 5 and Table 3 indicate that the two models have comparable performance, with the abductive model giving slightly better performance inspite of the fact that it uses only one sixth of the inputs. We use the z statistic to test the statistical significance of the difference in performance levels exhibited by the two models. This statistic tests the validity of the null hypothesis that there is no difference between the means of the absolute error (AE) in the two cases, given the statistical variations exhibited, and is expressed as [40]:

$$z = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma^2{}_1}{n_1} + \frac{\sigma^2{}_2}{n_2}}} \tag{20}$$

where μ_i, σ_i, and n_i are the mean and standard deviation of the absolute error (AE), and the sample size for the *i*th error distribution, respectively, and i ∈ 1,2. Here $n_1 = n_2 = 358$. At the 95% confidence level ($\alpha = 0.05$), the null hypothesis is accepted for $-1.96 < z < 1.96$. Applying Eqn. (20) for the AE statistics shown in Table 3, the null hypothesis is accepted, i.e. the difference in the forecasting performance of the neural and abductive models is not statistically significant.

4.2 Neural Network Committees

Table 4 summarizes the results obtained when the three neural models trained on 1987, 1988, and 1989 data, see Section 3.3, were used in the committee arrangement of Fig. 2. Here all three models have identical network structure, training parameters, and initial set of random weights. Each of the 47-6-1 models has 6 neurons in the hidden layer and uses the sigmoid transfer function for both the hidden and output layers. The Table shows the mean and standard deviation of both the APE and AE errors for the individual models, the committee combined output, and the monolithic model trained on collective data for the three years. Two methods were considered for combining the outputs of the committee members: Simple averaging, Eqn. (1), and weighted averaging, Eqn. (2). For the latter case weights were based on the error variance for the individual member models on their training data, Eqns. (4) and (5). The two output combining methods give similar performance, with the simple averaging method being slightly superior. Both committee approaches outperform all three individual models as well as the monolithic 3-year model. Error reduction by both the committee and the monolithic models from that of the 1988 individual model is statistically significant. However, performance improvement by either committee approaches compared to the monolithic model is not statistically significant. This may be attributed to the poor diversity among the individual model members of the committee. With these models using the same learning algorithm, network topology, and training parameters, diversity will be attributed only to the different load and temperature data for different years that were used to train them. It is expected that such data for successive years would also be strongly correlated, which reduces diversity among resulting individual models and degrades the committee advantage. Table 5 shows values for the three pair-wise Pearson correlation coefficient between data for each two of the three training years (1987 to 1989). Data is shown for the peak load and the two extreme temperatures. Shown also is the root-mean-square (RMS) value of the three coefficients, which indicates overall average correlation among data for all three years. Results show that load and temperature data can be highly correlated. For both the peak load and the minimum temperature, correlation is somewhat stronger between adjacent years. Accuracy for the combined committee output is influenced by correlation among prediction errors in the outputs of individual members. Ideally, if such errors are poorly (or even negatively) correlated, then they would at least partially cancel out, thus improving the committee forecasting accuracy. The top row of Table 6 lists values for the pair-wise Pearson correlation coefficient between the prediction errors for each two of the three individual models for years 1987, 1888, 1989, together with the corresponding RMS value.

To increase diversity among member models beyond that attributed to different training data, the three models were trained using different settings for the network topology and training parameters, e.g. number of hidden neurons, initial random weights (controlled by the seed of the random number generator used to generate the weights), learning rate increments. Table 7 shows changes introduced into the three models and the resulting performance, which should be compared with that listed in Table 4. Only a slight reduction (from 2.50 to 2.45 for the MAPE value) is achieved in the committee performance through changes introduced to make the models more independent. The bottom row of Table 6 lists the pair-wise error correlation coefficients after the changes, showing no reduction in the overall error correlation. The small drop in error may be attributed to improved performance by individual models, e.g. the 1989 model, rather than increased independence amongst the models. It appears that changes introduced in the design and training of the individual neural models are not enough to offset effects of the strong correlation between data for the training years. It should also be noted that the large number of parameters associated with designing and training a neural network makes it more difficulty to select parameter values to achieve a desired effect.

4.3 Abductive Network Committees

Table 8 summarizes the results obtained when the three abductive models trained on 1987, 1988, and 1989 data, see Section 3.3, were used in the committee arrangement of Fig. 2. Here all three models have the same default value of 1 for the CPM parameter that controls model complexity. The two output combining methods give similar performance, with the simple averaging method being slightly superior as was the case with neural committees. Both approaches outperform all three individual models as well as the monolithic 3-year model. Error reduction by both the committee and the monolithic models from that of the 1987 individual model is statistically significant. As with neural models, performance improvement by either committee approaches compared to the monolithic model is not statistically significant. The top row of Table 9 lists values for the pair-wise Pearson correlation coefficient between the prediction errors for each two of the three individual models for years 1987, 1888, 1989, together with the corresponding RMS value. To increase diversity among member models beyond that contributed to different training data, the three models were trained using different settings for the CPM parameter. Table 10 lists the CPM values for the three models together with the resulting performance, which should be compared with that given in Table 8. An appreciable reduction (from 2.36 to 2.19 for the MAPE value) is achieved in the committee performance through the use of different complexity for the individual member models to make them more independent. The bottom row of Table 9 lists the pair-wise error correlation coefficients after the changes, showing some reduction in the overall error correlation which is indicative of more independent models. Significance tests performed on the data in Table 10 show that performance improvement by the abductive network committee with members having different CPM values is statistically significant compared to the monolithic model, with a reduction from 2.52 to 2.19 in the MAPE error. With CPM being the main training parameter set by the user, the search for more independent abductive models has been much easier compared to the case of neural networks. Table 11 shows a more detailed performance comparison between the monolithic 3-year model and the best committee model, both using abductive networks. Gains by the committee approach include: one third of a percent point reduction in the MAPE error, 9 MW reduction in the mean absolute error, about 4 percent points reduction in the maximum absolute percentage error, and 6 percent points increase in the forecasting days having a percentage error within ± 3%.

5 Conclusions

We have presented a formal approach for applying the network committee technique to improve the accuracy of forecasting the next-day peak load using multiple-year historical load and temperature data. The method takes into account the trend due to annual load growth. Back propagation neural networks and GMDH-based abductive networks were considered as modeling tools. In both cases, a committee whose three members were trained on 3 individual successive years improved forecasting performance compared to individual models as well as a monolithic model trained on the full 3-year data. However, improvements achieved over the monolithic model were statistically significant only using abductive networks with individual members having different levels of model complexity to enhance independence. The strong correlation among load and temperature data for successive years tends to discourage diversity among resulting models, thus reducing the committee advantage. To overcome this problem, independence among the member models was enhanced through the use of different network structures and training parameters. Such attempts were not very successful with neural models, and the large number of parameters that can be adjusted made the task difficult. With abductive networks, simply using different values for the CPM parameter that controls model complexity has proved effective. Future work will consider other methods for enhancing diversity and improving performance of neural network committees through using different network architectures or learning paradigms. Expert committees will also be considered where committee members employ different machine learning techniques.

Acknowledgement

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

References

[1] G. Gross, F.D. Galiana, Short-term load forecasting, Proc. IEEE 75 (1987) 1558-1573.

[2] A.-U. Asar, J.R. Mcdonald, A specification of neural network applications in the load forecasting problem, IEEE Trans. Control Systems Technology 2 (1994) 135–141.

[3] Y.–Y. Hsu, C.–C. Yang, Design of artificial neural networks for short-term load forecasting. II. Multilayer feedforward networks for peak load and valley load forecasting, IEE Proc. C 138 (1991) 414 –418.

[4] T. Onoda, 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, Yokohama, Japan, 1993, 284-289.

[5] M.A. Aboul-Magd, E.E.-D.E.-S. Ahmed, 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, Halifax, Canada, 2001, 105–113.

[6] Y. Morioka, K. Sakurai, , A. Yokoyama, and , Y. Sekine, 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, Yokohama, Japan, 1993, 60–65.

[7] G.J. Montgomery, K.C. Drake, Abductive networks, Proceedings of the SPIE conference on the Applications of Artificial Neural Networks, Orlando, FL, USA, 1990, 56-64.

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

[9] A.P. Alves Da Silva, U.P. Rodrigues, A.J. Rocha Reis, L.S. Moulin, NeuroDem - a neural network based short term demand forecaster, IEEE Power Tech Conference, Porto, Portugal, 2001.

[10] R. E. Abdel-Aal, Short term hourly load forecasting using abductive networks, IEEE Trans. Power systems, In Press.

[11] Z.-H. Zhou, Y. Jiang, Y.-B. Yang, S.-F. Chen, Lung cancer cell identification based on artificial neural network ensembles. Artificial Intelligence in Medicine 24 (2002) 25-36.

[12] P.S. Parmpero, A.C.P.L.F. De Carvalho, Recognition of vehicles silhouette using combination of classifiers, Proceedings of the 1998 IEEE International Joint Conference on Neural Networks, Anchorage, AK USA, 1998, 1723 – 1726.

[13] Y. Shimshoni, N. Intrator, Classification of seismic signals by integrating ensembles of neural networks, IEEE Transactions on Signal Processing 46 (1998) 1194 – 1201.

[14] M.H.L.B. Abdullah , V. Ganapathy, Neural network ensemble for financial trend prediction, Proceedings TENCON 2000, Kuala Lumpur, Malaysia, 2000, 157–161.

[15] C.A. Antoniou, T.J. Reynolds, Modular neural networks exploit multiple front-ends to improve speech recognition systems, Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, Brighton, UK, 2000, 205–208

[16] X. Yao, M. Fischer, G. Brown, Neural network ensembles and their application to traffic flow prediction in telecommunications networks, Proceedings of the International Joint Conference on Neural Networks, Washington, DC, USA, 2001, 693–698.

[17] A. Khotanzad, R.-C. Hwang, A. Abaye, D. Maratukulam, An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities, IEEE Transactions on Power Systems 10 (1995) 1716–1722.

[18] T. Matsumoto, S. Kitamura, Y. Ueki, T. Matsui, Short-term load forecasting by artificial neural networks using individual and collective data of preceding years, Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems, Yokohama, Japan, 1993, 245–250.

[19] I. Drezga, S. Rahman, Short-term load forecasting with local ANN predictors, IEEE Transactions on Power Systems 14 (1999) 844 – 850.

[20] D. Jimenez, Dynamically weighted ensemble neural networks for classification, IEEE World Congress on Computational Intelligence, Anchorage, AK, USA, 1998, 753-756.

[21] M. Su, M. Basu, Gating improves neural network performance, IEEE International Joint Conference on Neural Networks, Washington, DC, USA, 2001, 2159–2164.

[22] D.H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241-260.

[23] J.-J. Guo, P.B. Luh, Market clearing price prediction using a committee machine with adaptive weighting coefficients, IEEE Power Engineering Society Winter Meeting, New York, USA, 2002, 77–82.

[24] A. Swann, N. Allinson, Fast committee learning: Preliminary results, Electronics Letters 34 (1998) 1408-1410.

[25] S.-J. Kim, B.-T. Zhang, Combining locally trained neural networks by introducing a reject class, IEEE International Joint Conference on Neural Networks, Washington, DC USA, 1999, 4043-4047.

[26] J. Krogh, A. Vedelsby, Neural network ensembles, cross validation, and active learning, in:D.S. Touretzky, M.C. Mozer, M.E. Hasselmo, eds., Advances in Neural Information Processing Systems 8, (MIT Press, Cambridge, Mass) 231-238.

[27] I.T. Podolak, S.-L. Lee, A. Bielecki, E. Majkut, A hybrid neural system for phonematic transformation, 15th International Conference on Pattern Recognition, Barcelona, Spain, 2000, 957-960.

[28] V. Radevski, Y. Bennani, Reliability control in committee classifier environment, IEEE-INNS-ENNS International Joint Conference on Neural Networks, Como, Italy, 2000, 561-565.

[29] P.S. Prampero, A.C.P.L. De Carvalho, Classifier combination for vehicle silhouettes recognition, Seventh International Conference on Image Processing and its Applications, Manchester, UK, 1999, 67-71.

[30] D. Edelman, P. Davy, Y.L. Chung, Using neural network prediction to arbitrage the Australian All-Ordinaries Index, Third International Conference on Knowledge-Based Intelligent Information Engineering Systems, Adelaide, SA, Australia, 1999, 166–169.

[31] X. Yao, M. Fischer, G. Brown, Neural network ensembles and their application to traffic flow prediction in telecommunications networks, International Joint Conference on Neural Networks, Washington, DC, USA, 2001, 693-698.

[32] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123-140.

[33] H. Drucker and C. Cortes, Boosting decision trees, in: S. Touretzky, M. C. Mozer, M. E. Hasselmo, eds., Advances in Neural Information Processing Systems 8 (MIT Press, MIT Press, Cambridge, MA, 1996) 479-485.

[34] S. Borra and A.D. Ciaccio, Improving nonparametric regression methods by bagging and boosting, Computational Statistics & Data Analysis 38 (2002) 407–420.

[35] W.Y. Goh, C.P. Lim, K.K. Peh, Predicting drug dissolution profiles with an ensemble of boosted neural networks: A time series approach, IEEE Transactions on Neural Networks 14 (2003) 459-463.

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

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

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

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

[40] W. Mendenhall and R.J. Beaver, Introduction to Probability and Statistics, (Duxbury Press, Belmont, CA, 1994).

Table 1. Summary of the 6-year load data showing information on the year-to-year growth, normalization factors used for dealing with the load growth trend, and scaling factors used for implementing the network committee forecasters.

Year, i		Total Annual Load, (MWH)	Mean Hourly Load, M_i (MW)	Annual Load Growth (year-to-year)	Trend Inputs, r_i $r_i = \frac{M_i}{M_{1987}}$	Scaling Factors, s_i $s_i = \frac{M_{1990\ (Estimated)}}{M_i}$
1985		16,310,645	1862	1		
1986		16,017,335	1828	0.982		
1987		16,510,405	1885	1.031	1.000	1.162
1988		17,563,434	2000	1.061	1.061	1.095
1989		18,434,815	2104	1.052	1.116	1.041
1990	Actual	19,357,130	2210	1.050		
	Estimated	19,184,400	2190	1.041	1.162	
Average Load Growth 1986-1990 (Actual)				1.035		

Table 2. Layout of a training record for the monolithic model utilizing all 3-year data using 48 inputs. Committee member models trained on individual years use the same layout with the trend input omitted.

Inputs			Output
Annual Trend Input	Data for each day, j, of seven preceding days: j = d-6, d-5, d-4, d-3, d-2, d-1, d	Data for forecasting day (d+1)	Peak load for day (d+1)
r_i	PL(j), Tmax(j), Tmin(j), WRK(j), SAT(j), SUN/HOL(j)	ETmax(d+1), ETmin(d+1), WRK(d+1), SAT(d+1), SUN/HOL(d+1)	PL(d+1)

Table 3. Performance comparison over the evaluation year for the neural and abductive next-day peak load forecasting monolthic models developed using all 3-year data.

Forecasting Method	Error Statistics for: APE, % and (AE), MW		Maximum APE, %	Correlation Coefficient , R, between Actual and Predicted	Percentage of forecasting days having:		
	Mean	SD			APE: ≤ 1%	APE: ≤ 3%	APE: ≥ 6%
Neural model (48-6-1)	2.61 (69.6)	2.08 (55.9)	12.52	0.986	24	65	6
Abductive model (8-1-1-1-1), Fig. 4	2.52 (70)	2.10 (61.4)	14.20	0.986	28	68	7

Table 4. Forecasting error statistics for the APE, %, and the (AE), MW, for the individual neural models (using identical network structures and training conditions), the committee models using two approaches for combining individual model outputs, and the monolithic model using all 3-year data.

Individual Member Models: Identical Network Structures and Training Conditions			Network Committee				Monolithic Model on 3-Year Data	
			Simple Averaging		Weighted Averaging			
Member Model	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1987	2.72 (73.2)	2.13 (61.2)	2.50 (67.2)	1.97 (57.1)	2.53 (68.0)	1.97 (57.7)	2.61 (69.6)	2.08 (55.9)
1988	2.91 (80.0)	2.31 (73.1)						
1989	2.76 (74.6)	2.24 (63.8)						

Table 5. Pair-wise Pearson correlation coefficients between data for each two of the three training years (1987 to 1989) for the peak load and the two extreme temperatures.

Data Parameter	1987-1988	1987-1989	1988-1989	RMS
Peak Load	٠,٨٣٧	٠,٧٦٥	٠,٨٢6	٠,٨١٠
Maximum Temperature	٠,٧٤٨	٠,٧٤٨	٠,٧٤٠	٠,٧٤6
Minimum Temperature	٠,٧٣٨	٠,٦٨٣	٠,٧١٥	٠,٧١3

Table 6. Pair-wise Pearson correlation coefficients between prediction errors over the evaluation year by the three neural member models using two approaches for training them.

Individual Members	1987- 1988	1987- 1989	1988- 1989	RMS
Identical Network Structures and Training Conditions (Table 4)	٠,674	٠,٧34	٠,564	٠,661
Different Network Structures and Training Conditions (Table 7)	٠,644	٠,٧61	٠,612	٠,675

Table 7. Forecasting error statistics for the APE, %, and the (AE), MW, for the individual neural models (using different network structures and training conditions), the committee model using simple averaging to combine individual model outputs, and the monolithic model using all 3-year data.

Individual Member Models (Different Network and Training Conditions)				Network Committee (Simple Averaging)		Monolithic Model on 3-Year Data	
Member Model	Changes from Table 4	Mean	SD	Mean	SD	Mean	SD
1987	4 hidden neurons, Different initial weights, Different learning rate increments	2.72 (73.3)	2.06 (59.3)	2.45 (65.6)	1.94 (56.1)	2.61 (69.6)	2.08 (55.9)
1988	No Change (6 hidden neurons)	2.91 (80.0)	2.31 (73.1)				
1989	8 hidden neurons, different initial weights	2.66 (71.8)	2.10 (59.2)				

Table 8. Forecasting error statistics for the APE, %, and the (AE), MW, for the individual abductive models (using the same CPM value), the committee models using two approaches for combining individual model outputs, and the monolithic model using all 3-year data.

Individual Member Models: Identical Training Conditions (CPM = 1)			Network Committee				Monolithic Model Using 3-Year Data	
			Simple Averaging		Weighted Averaging			
Member Model	Mean	SD	Mean	SD	Mean	SD	Mean	SD
1987	2.83 (79.6)	2.41 (74.6)	2.36 (65.2)	1.99 (60.0)	2.37 (65.7)	2.03 (61.6)	2.52 (70)	2.10 (61.4)
1988	2.48 (70.3)	2.28 (76.7)						
1989	2.76 (73.2)	2.36 (60.9)						

Table 9. Pair-wise Pearson correlation coefficients between prediction errors over the evaluation year by the three abductive member models using two approaches for training them.

Individual Members	1987-1988	1987-1989	1988-1989	RMS
Same CPM value (Table 8)	0.798	0.527	0.611	0.655
Different CPM values (Table 10)	0.588	0.577	0.566	0.577

Table 10. Forecasting error statistics for the APE, %, and the (AE), MW, for the individual abductive models (using different CPM values), the committee model using simple averaging to combine individual model outputs, and the monolithic model using all 3-year data.

Individual Member Models (Different Training Conditions)				Network Committee (Simple Averaging)		Monolithic Model Using 3-Year Data	
Member Model	Changes from Table 8	Mean	SD	Mean	SD	Mean	SD
1987	No change (CPM = 1)	2.83 (79.6)	2.41 (74.6)	2.19 (61.0)	1.87 (58.0)	2.52 (70)	2.10 (61.4)
1988	CPM = 0.5	2.49 (70.9)	2.22 (80.4)				
1989	CPM = 0.2	2.26 (62.2)	1.95 (58.5)				

Table 11. Performance improvements over the evaluation year obtained with the best abductive committee model in comparison with the corresponding monolithic model using 3-year data.

Forecasting Method	Error Statistics for: APE, % and (AE), MW		Maximum APE, %	Correlation Coefficient, R, Between Actual and Predicted	Percentage of forecasting days having:		
	Mean	SD			APE: ≤ 1%	APE: ≤ 3%	APE: ≥ 6%
Abductive Monolithic Model Using 3-Year Data	2.52 (70)	2.10 (61.4)	14.20	0.986	28	68	7
Abductive Committee with Different CPM Values (Table 10)	2.19 (61.0)	1.87 (58.0)	10.02	0.988	31	74	6

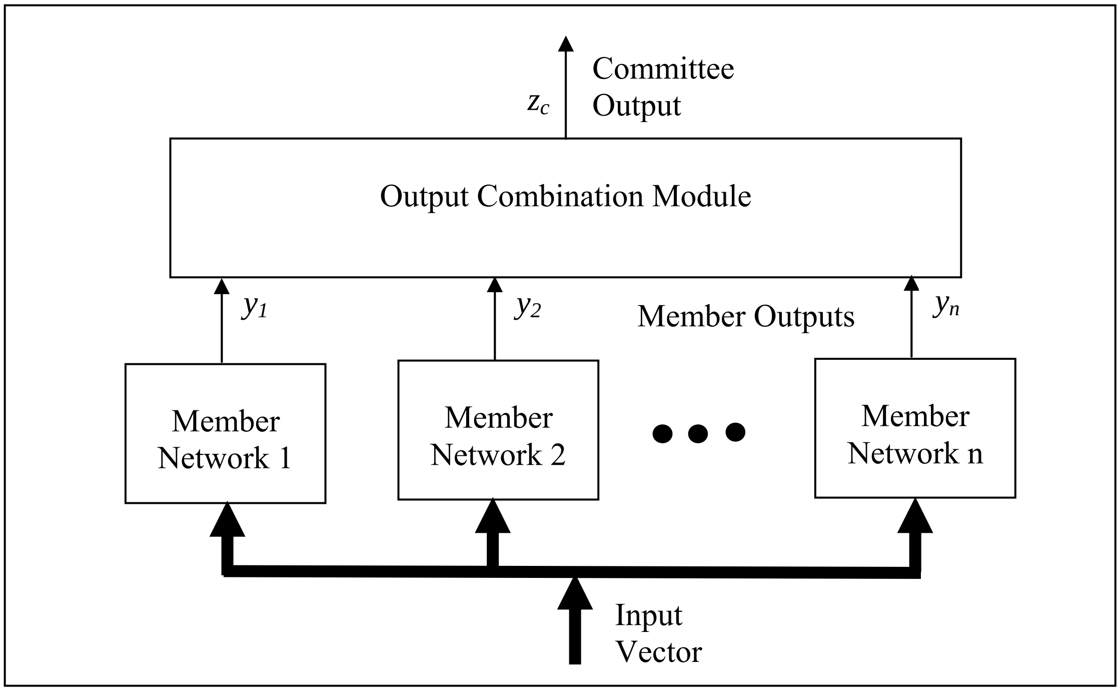

Fig. 1. A schematic diagram for a network committee.

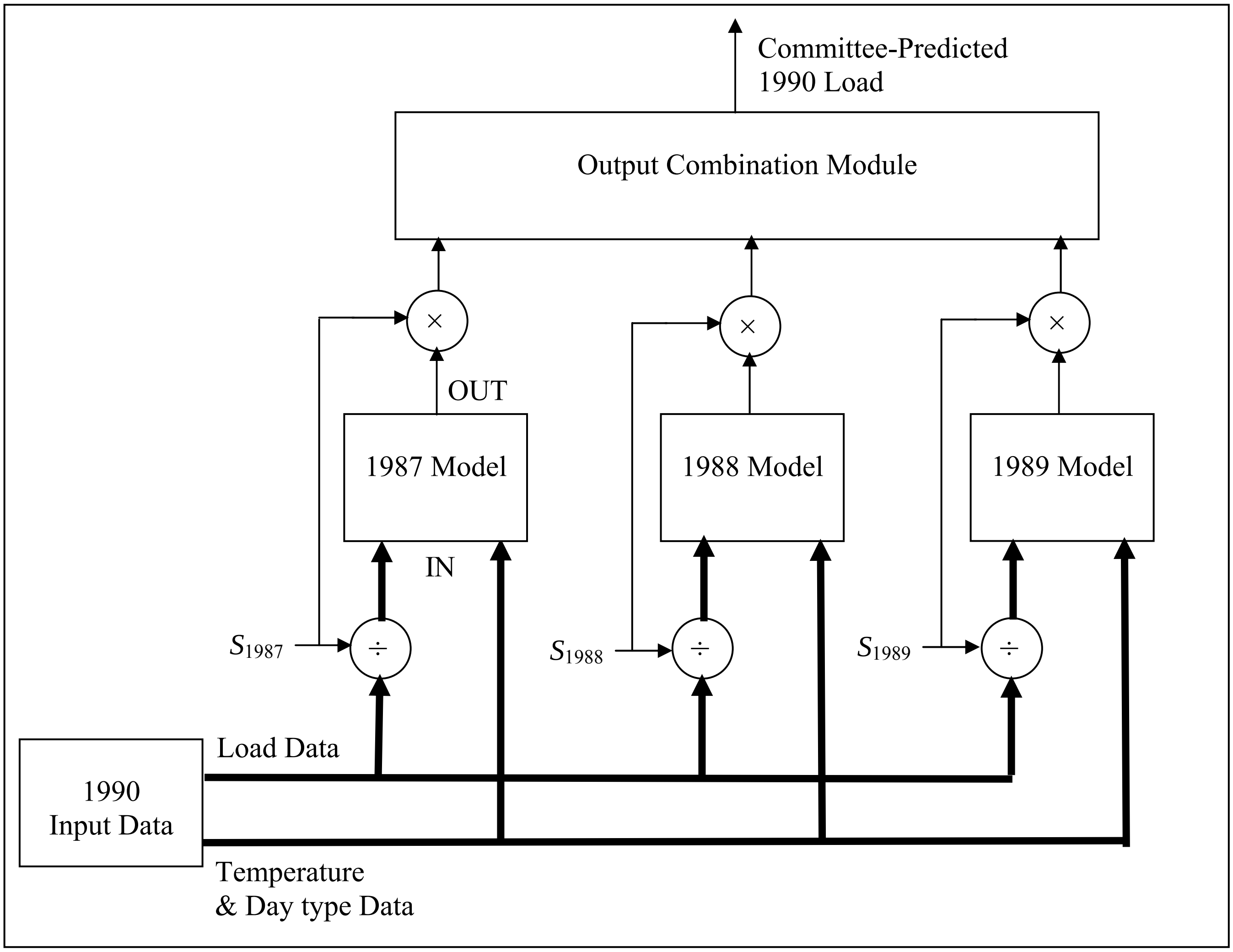

Fig. 2. Committee configuration for using the three individual member models trained on 1987, 1988, and 1989 data to forecast the 1990 evaluation year peak loads.

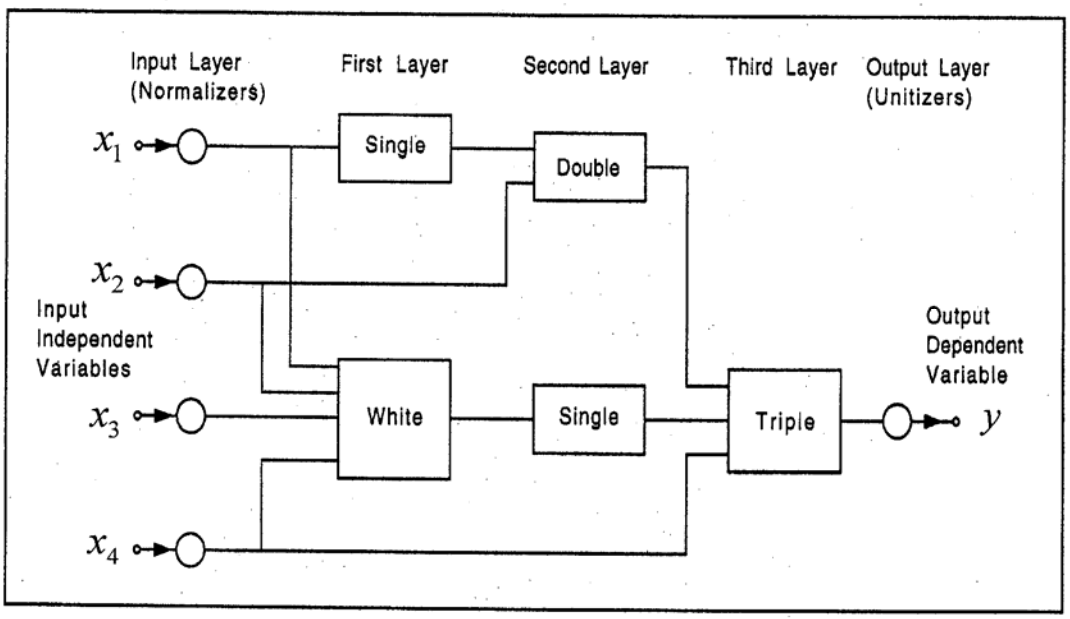

Fig. 3. AIM abductive network showing various types of functional elements.

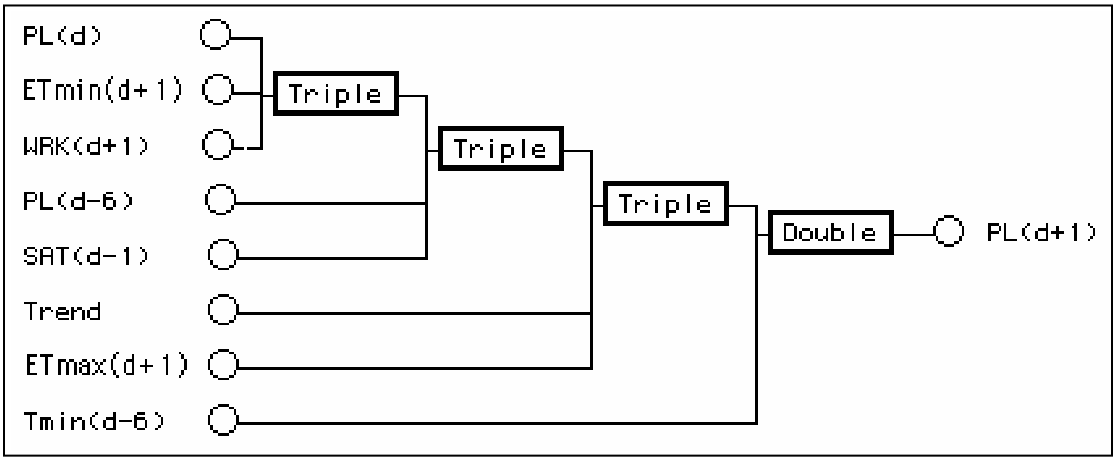

Fig. 4. Structure of the monolithic abductive network model for the peak load trained on the collective data for all three years (1987-1989). The model automatically selects only eight of the available 48 inputs.

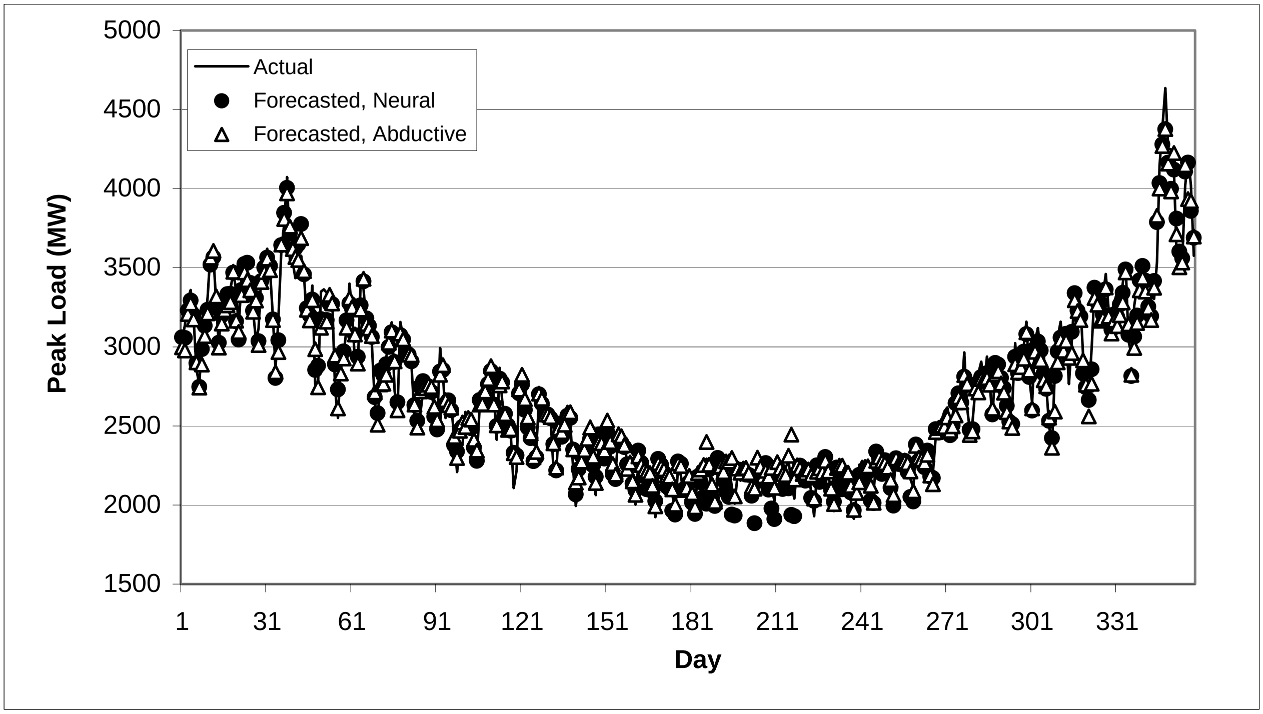

Fig. 5. Time series plots of actual peak loads for the 358 evaluation days in 1990 and forecasts from both neural and abductive network monolithic models trained on all data for the previous three years (1987-1989).