4373

Development of a Generalized Neural Network Model to Detect Faults in Building Energy Performance—Part II

Marcus R.B. Breekweg **Peter Gruber, Ph.D.** **Osman Ahmed, Ph.D., P.E.**
Member ASHRAE

ABSTRACT

Part I of this paper discussed the theoretical considerations of creating a nonlinear black box model. In Part II, the constraints on the nonlinear model imposed by the application are discussed, followed by presentation of the model structure, training method, input selection, and input transformation. The test results of applying the proposed model with the selected features to five test buildings are discussed next. One of the test buildings (Zachry Engineering Center) selected for this study was also used in a previous study as a part of energy prediction competition (Haberl and Thamilseran 1996). The proposed model is also compared with other black box models (linear and nonlinear). The conclusion of this paper is that the prediction accuracy depends heavily on the building to which the method is applied. The more regular the operation of a building is, the better the model approximates the actual data. In this respect, the Zachry Engineering Center is an ideal candidate for modeling, while some of the other buildings exhibit more irregular behavior.

MODEL SELECTION

Application Constraints

The man hours put into designing a building-specific model cannot be justified in practice. The goal of this study is to design a model that can be used on any building without changes to the model's design. A number of constraints can be drawn up to ensure minimum changes to the model design.

- **Fixed design.** Over- or underfitting cannot be compensated by changing the structure of the model but must be avoided in the algorithm.
- **Reproducibility.** The output of the model must be independent of initial settings. In other words, if the same day is tested a second time with the same inputs and training set, the output must be the same.
- **Automated input selection.** The input selection and transformation should be automated as much as possible.
- **Data update.** Since the building state is not static, the database from which the model estimations are made tend to become outdated. As a result, the model must adapt to new data in the database. It is necessary to ensure that the quality of the database is good before model estimations are made.
- **Robustness.** The model must be robust with respect to incomplete data.

Choice of Structure and Training

The following decisions must be made in order to fulfill the requested goal:

- Model structure
- Global/local training range
- Data point selection
    - Clustering
    - CBR method

Model Structure

As stated in Part I, the selected model structure is static. The main reason for not investigating dynamic structures is their complexity compared to static structures. Since the model must perform with any building data, the model should be as simple as possible.

Marcus R.B. Breekweg is a client partner at Cambridge Technology Partners, Amsterdam, the Netherlands. **Peter Gruber** is a senior scientist at Siemens L&S Division, Zug, Switzerland. **Osman Ahmed** is a senior principal engineer at Siemens Building Technologies, Buffalo Grove, Illinois.

The static model structure used in the proposed model as discussed in Part I is the general regression neural network (GRNN) or conditional mean as a normalized radial basis function (RBF) network. The following considerations were made to choose a normalized RBF over a multi-layer perceptron (MLP) structure:

- The advantage of RBFs compared to the distributed architecture of MLPs is that local units can adapt to local patterns in the data without having unwanted side effects in other regions. In a distributed architecture such as a MLP, adapting the network to fit a local pattern in the data can cause unwanted side effects in other parts of the input space. This is true especially since data updates may only pertain to a certain area of the input space (e.g., a chiller replacement would probably only affect the cooling season). MLPs could cause such changes in other parts of the inputs space (e.g., heating season).
- Since the model must perform well with any given data set, the MLP could result in overfitting with one data set and underfitting with another, due to the fixed number of neurons in the hidden layer. Although it is possible to use an algorithm that adds neurons depending on the data set, it is easier to select a normalized RBF architecture that automatically adapts the number of activation functions to the number of training vectors.

Based on these two considerations, a normalized RBF network is preferred over an MLP. The next consideration is the reason for choosing a GRNN as the normalized RBF.

- Compared to GRNN, other normalized RBF networks (and the MLP as well) have more free parameters to be optimized. This increases the chance of achieving local minima during optimization, thereby possibly compromising the reproducibility. With GRNN, only one parameter (the smoothing parameter) has to be optimized.

Global/Local Training Range

Local training has been selected for the proposed model based on two considerations.

1. Local training can be faster than global training since not all training data are used during computation. If there were 10,000 training vectors in the whole set, a GRNN trained globally with a validation set of 2000 vectors would take 20 million distance calculations. Using local training with a subset can reduce calculations drastically.
2. With GRNN, only one smoothing vector is calculated for all kernels. Using GRNN with local training still has the benefit of optimizing only one free parameter but also the benefit of having a different smoothing parameter per subspace. This enables a better adaptation to local patterns.

Data Point Selection

The next decision concerns using a clustering algorithm or a case based reasoning (CBR) method for data point selection for local training.

Clustering Algorithms

In Part I, clustering was proposed for local training. Certain constraints, however, apply to the clustering algorithm for benchmarking.

a. The algorithm must be robust. The algorithm must always result in the same clusters regardless of its initial settings or the presentation order of the data points. This constraint is directly related to the reproducibility constraint.
b. The number of clusters should be defined by a conditional distance or probability density rather than by a prior chosen number. For different buildings, different clusters can be distinguished in the data sets, as well as a different number of clusters (e.g., for an office building, weekends and workdays may need distinct clusters; whereas for a hotel or a hospital, weekend loads may not be distinguishable from general workday [weekday] loads).
c. The clustering should be fast since the data sets used for the modeling must be easy to update. Robust clustering methods require a lot of computation time, since all points in the data set must be compared to one another.

Knowledge-based clustering

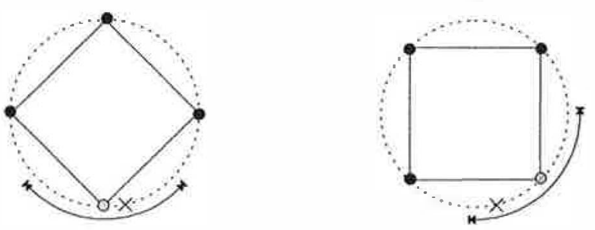
It is not easy to find clustering criteria relevant to any data set. Clustering based on knowledge of building operations for every building is too expensive and commercially not justifiable.

K-mean clustering

It is already obvious from the name of this technique that it does not comply with the second constraint. As mentioned in Part I, this method assumes a fixed number of clusters. Part I also mentions that the k-mean algorithm is influenced by the initial assignment of the centroids and, therefore, does not comply with the first constraint either. However, the latter problem could be solved with a pseudo-random selection of the initial assignment. However, this does not guarantee an optimal clustering and does not solve the fixed cluster problem.

Kohonen's self-organizing feature map (SOFM) algorithm

With Kohonen's self-organizing feature maps (SOFM), the first constraint poses a problem. The forming of the clusters does not depend on the initial settings but is influenced by the presentation order. This problem may be eliminated by offering the data points in a pseudo, or nonrandom, way to the Kohonen algorithm. To compare simulations, a constant seed is often used. Another problem is that SOFM suffers from boundary effects. This boundary effect is due to the fact that the output nodes at the boundary of the neuron-grid need too much time to adapt to the extremes of the input set.



A *B*

Figure 1 *Both clustering methods in A and B have the same clustering error and, therefore, have a correct representation. If the X is a test input, the vectors belonging to the winning cluster (light grey) are comparable vectors. The vectors belonging to the winning cluster are located in the lowest quarter of the circle in A, but in B the vectors belonging to the winning cluster are located in the lower right quarter. The test vector X will be compared with a different subset for A and B.*

Figure 1 shows the problem of random presentation for clustering vectors divided evenly over a circle with a two-by-two Kohonen SOFM.

The second constraint poses an even bigger problem. As already mentioned, with SOFM, the number of neurons (= clusters) must be preselected. An adaptive algorithm, which adds or reduces the number of clusters depending on the underlying distribution, is not incorporated in a SOFM.

Self-creating and organizing neural networks (SCONNs)

The SCONN neither has the boundary problems related to SOFM nor does it need a prior decision on the number of clusters. This cluster method could be a suitable cluster algorithm for the application of benchmarking because it complies with all three constraints related to clustering. But because this algorithm was not found until the end of the project, it was not considered.

There are several other clustering methods, but none of the reviewed cluster techniques comply with all three constraints. Furthermore, there are application constraints of corrupted data. Although the proposed model was not tested for this constraint, it should still be taken into consideration during model selection. A missing element from an input vector would require total reclustering since the clustering must be based on one less input dimension. As a result, clustering is not used in the proposed model.

CBR Method

There are three reasons for using case based reasoning (CBR) for local training.

1. CBR finds the same training subset every time the same test vector is used.
2. It is easy to add new training vectors since a newly calculated approximation is made for every new test vector. A new training vector will, therefore, be included automatically if it lies close enough to the test vector.
3. If a test vector is corrupted and an element in the vector is missing, the distance criterion can easily be based on one less dimension.

CBR Choices

In Part I, the following four CBR methods are presented:

- Knowledge-based decisions
- Fixed radius
- Fixed number of vectors
- Adaptive radius with a preset minimum number of training vectors

The decisions for selecting the relevant training vectors for all four methods are based on a rule of thumb or are deterministic. For the first method, the question is on which information to base the selection. The radius and the number of vectors still have to be chosen for both the second and third methods, respectively. For the last method, the relationship between the density estimation and the adaptive radius must be defined.

With a fixed radius, the radius selected may not lead to a training set that is big enough for a good estimation. On the other hand, a fixed number of vectors may compromise the ability of the model to adapt to local patterns.

For the proposed model, a combination between the first and the last possibility is used. The first selection will be knowledge-based, after which the number of vectors in the subset is further reduced using an adaptive radius method.

Knowledge-based decisions

Since weather conditions and the occupancy level are mostly dependent on the time of year and the time of day, the data set is reduced by assuming the following:

- Most buildings are occupied during office hours. The load at night and during the day probably have different dependencies. Consequently, only training vectors featuring a similar hour time-stamp are used.
- The load dependencies in summer and in winter are most likely different because different systems are often in operation (heating in the winter and cooling in the summer). As a result, training vectors that are not more than 60 days apart from the test vector in the date stamp are selected.

Adaptive radius with a preset minimum number of training vectors

Reduction of this subset of training vectors is performed using a radius dependent on the Parzen density estimation, as suggested in Part I. The relationship between the adaptive radius and the density estimation is determined by trial and

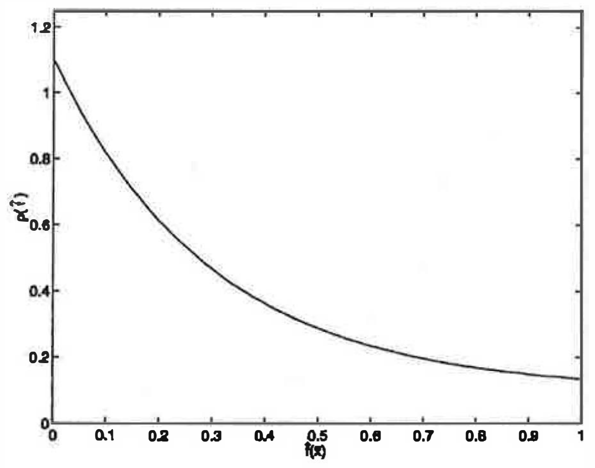


Figure 2 *The relationship between $f(\bar{x})$ and ρ as found with experiments.*

error during experiments. Thus, this method equates to a mere rule of thumb. The following relationship is used:

$$\rho(\hat{f}) = \rho_{min} + e^{\left(\frac{f(\bar{x})}{c}\right)} \tag{1}$$

where

ρ	= selection radius.
$f(\bar{x})$	= probability estimation of test vector, calculated with the Parzen density estimation. The $f(\bar{x})$ can be estimated from any suitable function expression as indicated in Equations 9, 10, or 11 in Part I.
ρ_{min}	= constant minimum radius (ρ_{min} is set to 0.1 for all tests).
c	= constant (c is set to 0.3 for all tests).

The case based reasoning (CBR) method is based in part on knowledge and the adaptive radius method. Hereafter, CBR will be used to refer to this selection method.

The proposed model is locally trained with a training subset that is obtained with the CBR method. The general regression neural network (GRNN), as a normalized radial basis function (RBF) structure, is used for the calculation of the approximation.

INPUT SELECTION

It is advantageous to make a fully automated input selection from a building database to model the energy usage of a building. For the selection to be automated, it must first be clear what kind of time-series are present in the database and where in the building the measurements are taken. Secondly, information is needed to indicate the relevance of a particular time-series to the whole building energy usage.

A practical problem arises with the first condition. There are several BEMS with their own database structure. The signal information and the location are described in a descriptive field as a string of characters instead of by special predefined codes. Therefore, an automated interface to select the requested inputs is often problematic or even impossible.

The second condition is not easy to comply with either. The purpose of the modeling is to find the underlying relationship between the measured signals obtained from a building and its energy usage. It is possible to use all available inputs in a neural network that updates the input weights during training (e.g., a feed-forward back-propagation network). The weights for nonrelevant inputs should tend to move toward zero. In reality, these weights do not become exactly zero because a relationship always exists between two finite time-series. With fairly short time-series, a cross-correlation may be quite big, even between two physically independent variables.

With networks such as GRNN, the inputs are not weighted and other methods have to be used to find the relevant inputs. There are some linear multiple-regression techniques that can discriminate between the inputs. A Bayesian framework (MacKay 1994) is used for this purpose. It is, however, not proven that either method finds the relevant inputs for every possible problem. Extensive trial-and-error runs for every building in order to find the best combination of input variables represent another option but may take very long and are therefore commercially unattractive.

The design of an automated input selection procedure would probably be very difficult. Since building an automated selection procedure is not the main objective of this study, designing such a procedure is left to further study. In the proposed model, the building operator (or user) chooses the inputs because a building operator is able to interpret the text in the description field of a database and guess which variables are relevant and which are not.

A selection must be made from the following possible inputs:

- Date-stamp: year, month, and day
- Day-type information: working or nonworking day
- Outside air temperature
- Relative or absolute outside air humidity
- Solar radiation
- Time-stamp: depends on the sample frequency (e.g., daily, hourly, or ones per 10 minutes)
- Wind speed

Certain building data sets include all inputs or even multiple inputs of the same class. The building operator could be guided by a setup program that asks for the point-name representing a given input class (e.g., most relevant outside air temperature).

In the "Input Transformation" section, a more complete list of possible input variables is given. Many building databases do not include all of these variables, and some variables are available only for specific zones within the building instead of for the whole building.

It is also important to keep the number of inputs to a minimum to avoid dimensionality problems as mentioned in Part I. In this study, only the date- and time-stamps, the day-type information, and the outside air temperature are used.

Input Transformation

To make use of the information hidden in the measured data, input transformations must be made. The transformation depends on the type of variable that has to be preprocessed.

Representation of Time and Date Variables

- Date-stamp: the date information is converted into the day-number of the year (e.g., January 1 = 1, May 5 = 125, or 126 if it is a leap year).
- Time-stamp: this is converted into hours if necessary (e.g., 07:40 h = 7.67, 18:12 h = 18.20).
- Day type (working or nonworking day [1 or 0]): this variable should give an indication of the occupancy level in the building. In a previous study (Breekweg 1995), it was found that a binary occupancy indicator can cause discontinuities in the model's output at the point of a day-type change. To avoid this discontinuity, the occupancy indicator can be made continuous.

It is also possible to incorporate holiday influences into the occupancy indicator instead of having an additional variable. This reduces the input dimension as well.

The conditional mean is used on the binary signal. As a result, the influence of a holiday season is incorporated and the signal is no longer discrete (see Figure 3).

Three occupancy indicators are used. The preceding occupancy of 24 hours, the current occupancy, and the occupancy in 24 hours. The influence of past and future occupancy levels can then be optimized within the model by weighting.

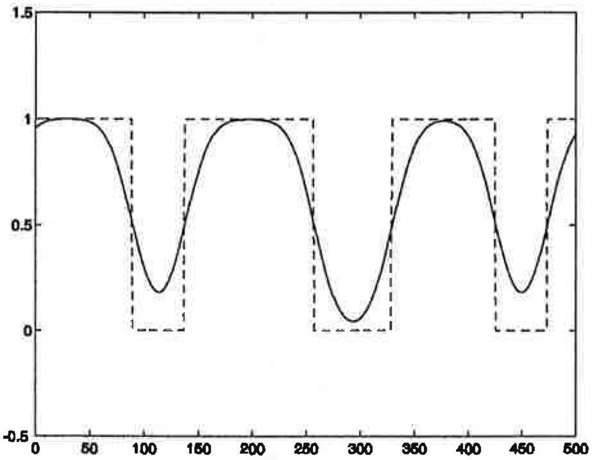


Figure 3 *The continuous output of the conditional mean with a binary input.*

Representing Environmental Variables

In this study, a static nonlinear model is used. This means that the environmental variables have to either be filtered or delayed to include the dynamics of the building in the model.

With physical insight about the system at hand, this information should be used to form new variables by transforming the raw measurements. The filter time constants should be based on physical considerations if possible.

Slow-changing environmental variables are expected to have a relatively large influence on the building structure since the building's thermal capacity is fairly massive. Since temperature and humidity are relatively slow changing variables, larger time windows are chosen for those signals (24-hour moving average windows).

Using the current outside air temperature is not expected to have an instant influence on the inside air conditions or the energy usage. It is possible to calculate a time lag from a cross-correlation between the outside air temperature and the building thermal load. Using the temperature corresponding to the lag-time gives a better relationship to the energy usage than using the current temperature.

Since the capacity of the structure is relatively large, fast-changing signals are expected to have relatively little impact on the building structure. The long-term effects of fast-changing signals, such as solar radiation and wind, are eventually lumped into the outside air temperature. Conversely, the inside air temperature can also be directly influenced by the solar radiation. Therefore, short time constants are used for this kind of environmental variable.

As previously mentioned, the only environmental variable used in this study is the outside air temperature. This variable is transformed into three input variables described as follows:

1. Outside air temperature with a lag calculated from the cross-correlation with the building energy usage.
2. Twenty-four-hour moving window average temperature before the observed value as indicated in item 1.
3. Twenty-four-hour moving window average temperature before the calculated temperature as indicated in item 2.

Input Scaling

After the input transformation, the inputs must be scaled. This is necessary so that the data are in the range where the activation functions are not in saturation. Additionally, all the input variables should be approximately equal in size, which makes them of equal importance. For networks that don't update input weights (e.g., GRNN), prior knowledge of the importance of the input variables could be integrated in the model by weighting the scaled input according to importance. Generally this type of input weighting is not used. For back propagation (BP) models, this is not necessary because this algorithm incorporates input weight updating.

Each of the time series is normalized by subtracting the mean value of the time series and dividing it by twice its standard deviation.

$$X' = \frac{X - E_x}{2 \cdot \sigma_x} \tag{2}$$

where

X' = the normalized time series,

X = the original time series,

E_x = mean of the original time series,

σ_x = standard deviation of the original time series.

The reason for using this normalization is that the outliers were not included in the normalization process. If normalization occurs so that all data points lie in the range [-1,1] or [0,1], it is based on the outliers. If outliers are subsequently omitted, the normalization of the different data series is no longer the same (see Figure 4). The reason for using twice the standard deviation is that approximately 95% of the data points will lie in the range [-1,1]. In this range, the sigmoid function and the radial basis function have a high gradient.

MODEL PROPOSAL

The proposed structure of the system is now as follows (see Figure 5).

Input selection. Only four different raw inputs are selected.

1. Outside air temperature
2. Time-stamp
3. Date-stamp
4. Binary signal: working/nonworking indicator

Input transformation. The following input transformations are performed:

1. Outside air temperature is converted into three signals:
   - T1 = outside air temperature with a lag of h hours. Lag h is calculated from the cross-correlation between the outside air temperature and the energy usage.
   - T2 = average temperature over the 24 hours before h.
   - T3 = average temperature over the 24 hours before T2.
2. Time-stamp is converted into hours.
3. Date-stamp is converted into the day number of the year.
4. Binary indicator is converted into three signals:
   - O1 = smooth signal obtained by using the conditional mean with a smoothing parameter of 18 hours (see Figure 3).

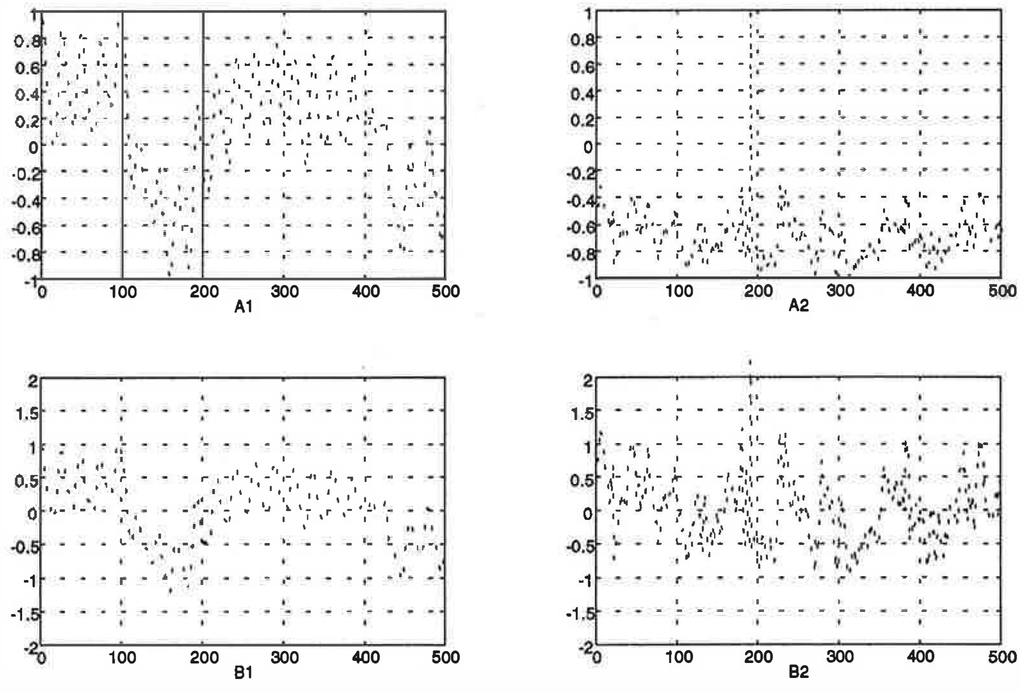


Figure 4 *Graphs A1 and A2 are two signals (e.g., two different inputs) that are scaled with respect to the maximum and minimum value of the signal. Graphs B1 and B2 show the same signals normalized with Equation 2. More than 95% of the data points of A2 lie in the range [-1, -0.2], whereas for A1, the range is [-1, 1]. The normalization of the two signals in B1 and B2 is not made with respect to the outliers, and 95% of the data points lie in approximately the same range ([-1, 1]).*

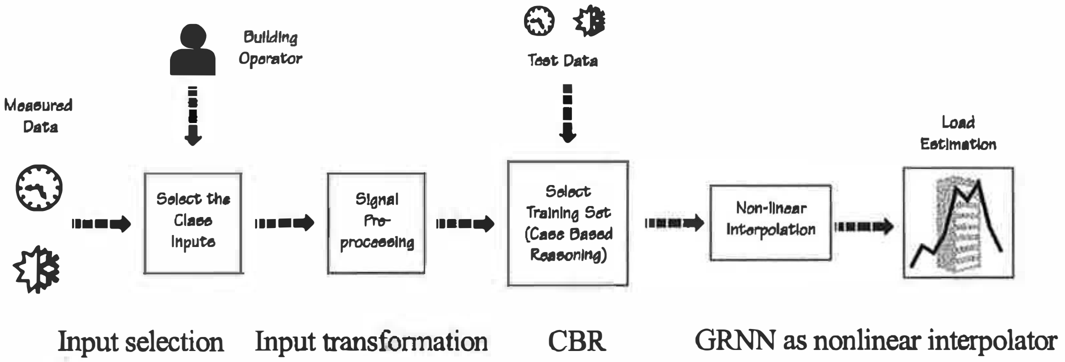


Figure 5 *Overview of the different phases in the modeling system.*

- O2 = O1 shifted 24 hours back in time.
- O3 = O1 shifted 24 hours forward in time.

This adds up to eight input vectors. For daily sampled data, T1 is the actual temperature of the day and T2 the temperature of yesterday. T3 is the average temperature of the day before yesterday. The binary indicator is directly shifted back and forth by one day without using conditional mean.

Selection of relevant training sets (case based reasoning). For each case and test vector, the training set is first limited to the training vectors that comply with the following conditions:

a. The day number of the training vector data set must be within 60 days of the test vector data set.
b. The hour of the training vector data set must be within three hours of the test vector data set.

A second reduction of training vectors is performed using the Parzen density estimation on the other six inputs and calculates the radius according to Equation 1.

NONLINEAR INTERPOLATION

GRNN (or conditional mean) is used to compute the energy usage estimation. The GRNN returns the estimation by performing a nonlinear interpolation. The smoothing parameter is not calculated using the holdout method, as this method is very time consuming for larger training sets. Instead, the smoothing parameter is calculated by taking the nearest five data points closest to the test vector from the subset of selected training vectors and by using these five data points as a validation set to calculate the optimum smoothing parameter. The smoothing parameter is calculated by minimizing the SSE of the GRNN on the five data points taken from the subset. A built-in optimization function in a commercial software (Beale and Demuth 1994) was used for this purpose. This function finds the minimum within a data range based on golden section search and parabolic interpolation. After the smoothing parameter is calculated, the five nearest data points are again included in the subset and the load estimation is computed. The calculation is performed once. The holdout method is used once to check both performance of estimation and calculation time. Table 6 shows the difference in estimation accuracy and calculation time between the hold-out method and the method using a validation set composed of the five training vectors closest to the test vector.

PERFORMANCE CRITERIA

Three performance measures are used to compare the results of different tests.

Mean bias error (MBE):

$$MBE = \frac{\sum_{i=1}^{n}(y_{pred,i} - y_{data,i})}{n \cdot \bar{y}_{data}} \times 100\% \quad (3)$$

Coefficient of variation (CV):

$$CV = \frac{\sqrt{\frac{\sum_{i=1}^{n}(y_{pred,i} - y_{data,i})^2}{n}}}{\bar{y}_{data}} \times 100\% \quad (4)$$

where

$y_{pred,i}$ = predicted dependent variable value
$y_{data,i}$ = data value of the independent variable
$\bar{y}_{data}$ = mean value of the dependent-variable test set
n = number of records of data in the test set

Robust CV (RCV):

$$RCV = \frac{\sqrt{\frac{\sum_{i=1}^{n}(y_{pred,i} - y_{data,i})^2_{90\%best}}{n}}}{y_{data,95\%} - y_{data,5\%}} \times 100\% \quad (5)$$

The third criterion was suggested by MacKay (1994). This performance measure neglects the outliers. In this measure, the 10% of worst predictions are omitted and normalization occurs by dividing the data set by the distance between the 5th and 95th percentile of the entire data set. MacKay added this performance measure because the CV measure implicitly assumes that the residual error from the prediction is Gaussian. MacKay's RCV measure corresponds roughly to a model that assumes that the residual is Gaussian, except for glitches that occur 10% of the time, which are uniformly distributed and are much larger than the normal residual level. For example, these glitches can be caused by sensor failures or data corruption.

RESULTS

This section discusses the results with simulated and real data. A total of four buildings and one set of simulated data are tested. Three of the four building data sets and the simulated data set contain hourly time-series. The fourth building data set consists of daily time-series.

All data sets are tested with the proposed model and with five to six other models. Two of the models, a GRNN and a MLP BP, are implemented in a commercial software package (Ward 1995). These models are referred to as NS GRNN and NS BP hereinafter in this paper. Three other models follow the proposed model except for the nonlinear interpolation/approximation module. This module is replaced with linear interpolation, MLP nonlinear interpolation, and the average of the selected data points, respectively.

The Zachry Engineering Center data and the test building 3 data are also compared with the model designed in a previous study (Breekweg 1995). This model is referred to as the Combi Net because of its hybrid training method. This model consists of two MLPs that are trained with a subset of the training data (local training). The outputs of these two MLPs are then combined in a third MLP trained globally.

Unfortunately, it is impossible to give an exact definition of the algorithms implemented in the software package (Ward 1995). Information of that sort is not provided by the software vendor.

The next section begins with an overview of the extra models for comparison, and the selected inputs per method are given. Then the results on the simulated data are presented. Finally, the results on the real building data are presented.

Training Methods and Input Variables

Each data set is tested with six models (Zachry Engineering Center and test building 3 with seven models). These models can be divided according to their training methods, global training and local training methods, as discussed in Part I. The global training methods are performed with an application from commercially available software (Ward 1995). The results from these methods are used for comparison. The local training methods are based on the CBR approach.

The following models with global training methods are used.

1. **NS GRNN Global.** The built-in genetic algorithm (part of NET-PERFECT function in the commercial neural network software (Ward 1995) is used to find the best set of smoothing parameters (GRNN with the NET-PERFECT option). The genetic algorithm works by selective breeding of a population of potential smoothing parameters. The genetic algorithm is seeking to breed the set of smoothing parameters that minimizes the mean squared error of the validation set. The larger the breeding pool size, the greater the potential of it producing better smoothing parameters. However, each breeding set of smoothing parameters is tested with the validation at each cycle, so larger breeding pools take longer training. The breeding pool size is set to 20 for all data sets. The algorithm finds a general smoothing parameter plus individual smoothing parameters for each input. These input smoothing parameters are the weights for the inputs as noted in Equation 9 in Part I. The larger the factor for a given input, the more important that input is to the model (with respect to the validation set). The validation set is obtained by extracting 10% of the training vectors randomly. The calculation is performed once.
2. **NS BP Global.** The back-propagation (BP) algorithm is also used with the NET-PERFECT option. For back-propagation (BP) networks, NET-PERFECT uses a validation set (selected from the training set) for cross-validation to compute the optimum point to save the network as discussed in Part I. This prevents the network from overfitting the data. Furthermore, the option TurboProp (another special built-in function of the neural network software) is used, which does not require a preset learning rate or momentum. All tests are performed with a three-layer neural network with 25 hidden neurons. This relatively high number of hidden neurons is chosen to ensure that the network has enough freedom to avoid underfitting. Overfitting is avoided with the NET-PERFECT option. The calculation is performed five times; the best performance on the validation set is chosen for the eventual estimation. The validation set is obtained by extracting 10% of the training vectors randomly.

The following models with local training methods are used.

3. **CBR GRNN Local.** See the "CBR Method" and "CBR Choices" sections of this paper.
4. **CBR BPLM Local.** CBR BPLM Local has an MLP neural network structure that uses BP with the Levenberg-Marquardt (BPLM) optimization to compute the energy usage estimation. This nonlinear interpolation algorithm is taken from a commercial software (Beale and Demuth 1994). A three-layer, back-propagation, feed-forward neural network is used. The hidden layer

has three hidden neurons with sigmoid activation functions. The output neuron has a linear amplifier as an activation function. The network is randomly initialized. The Levenberg-Marquardt optimization was used as part of the back-propagation algorithm. No attempt is made to optimize the neural network to prevent the algorithm from getting stuck in local minima. No measures are taken to see if there is over- or underfitting. The calculation is performed five times, and the best performance on the validation set is chosen for the eventual estimation. The validation set is obtained by extracting 10% of the training vectors randomly.

5. **CBR Linear Local.** Here the energy usage estimation is a linear estimator of the loads corresponding to the selected subset. The solution to such a problem can be directly calculated or derived by training a linear network with the Widrow-Hoff learning rule (Hassoun 1995; Beale and Demuth 1994). Since training a linear network takes a lot more calculation time, the problem is solved directly. It might, however, occur that the equation is underdetermined. In that case a zero error solution is returned. However, this solution is not unique. The solution was obtained using a packaged software solver (Beale and Demuth 1994). The solution yields smallest weights and biases. The linear interpolation might, therefore, be a poor generalization. An approximation of the ideal solution to an underdetermined problem can still be found by training a linear network. This is, however, not implemented to minimize calculation time. The calculation is performed once.

6. **CBR Average Local.** This estimation is calculated by taking the average of the energy targets corresponding to the selected data points.

7. **Combi Net BPLM Global** (hybrid trained model: two locally trained MLPs combined in a third globally trained MLP). The data sets of the real building and test building 3 are also tested with an extra network. This network was designed in a previous study (Breekweg 1995; Breekweg and Gruber 1996). The calculation is performed five times; the best performance on the validation set is chosen for the eventual load estimation.

The target output for all tests is the building energy usage. As mentioned in Part I, it is not easy to automatically select the relevant inputs. It will be left up to the building operator to select the appropriate inputs. For all tests, the following inputs were selected for the hourly respectively daily time-series.

Inputs for hourly sampled data		Inputs for daily sampled data	
1	Year	1	Year
2	Month	2	Month
3	Day	3	Day
4	Hour		
5	Binary workday/ nonworkday indicator	4	Binary workday/ nonworkday indicator
6	*Temp (h)*	5	*Temp (d)*

where *Temp* (h) = temperature at time h.

For the hourly data, the following transformed input sets are used.

Transformed inputs for global training (NS GRNN and NS BP) with hourly sampled data		Transformed inputs for local training (CBR) with hourly sampled data	
1	$-\cos(2\pi\,[h + 8532 \text{ hours}] / 8760)$	1	Day of the year (1 to 366)
2	$-\sin(2\pi\,[h + 8532 \text{ hours}] / 8760)$		
3	$\cos(2\pi\,[h / 24])$	2	Hour of the day (0 to 23)
4	$\sin(2\pi\,[h / 24])$		
5	$L_o(h - 24)$	3	$L_o(h - 24)$
6	$L_o(h)$	4	$L_o(h)$
7	$L_o(h+24)$	5	$L_o(h + 24)$
8	$Temp\,(h - lag)$	6	$Temp\,(h - lag)$
9	$\frac{1}{24}\sum_{i = lag+1}^{lag+24} Temp(h-i)$	7	$\frac{1}{24}\sum_{i = lag+1}^{lag+24} Temp(h-i)$
10	$\frac{1}{24}\sum_{i = lag+25}^{lag+48} Temp(h-i)$	8	$\frac{1}{24}\sum_{i = lag+25}^{lag+48} Temp(h-i)$

where h = hour of the year (1 to 8760), $L_o(h)$ = level of occupancy at time h, and $Temp(h)$ = temperature at time h.

There is a difference in time representation between the global and the local approach. This is done because it is not possible to create a correct time representation in one dimension in the global approach. For the local approach, this problem does not occur since a new training set is selected for every test vector. By taking the day and hour differences between the selected training vectors and the test vector, a correct one-dimensional representation of time is available. The advantage is a direct dimension reduction.

With global training, a time difference cannot be calculated because there is no reference time. One could directly use the time in hours, but that would cause a discontinuity problem going from 2300 hours to 0000 hours. Although only one hour has passed, these two hours are at opposite extremes of the time scale. To avoid such problems with global training, time is represented by the cosine and sine of the day angle instead.

For test building 3 and the Zachry Engineering Center, an extra global network is trained. The inputs used for this network are based on the previous study (Breekweg 1995). The influences of the sun, wind, and humidity are omitted for comparison with the other models. The following table shows the inputs for this network.

Transformed inputs for global Combi Network (Breekweg 1995) with hourly sampled data	
1	$-\cos(2\pi\,[h + 8532 \text{ hours}] / 8760)$
2	$-\sin(2\pi\,[h + 8532 \text{ hours}] / 8760)$
3	$\cos(2\pi\,[h / 24])$
4	$\sin(2\pi\,[h / 24])$
5	$\cos(2\pi\,[h / 12])$
6	$\sin(2\pi\,[h / 12])$
7	$L_o(h-24)$
8	$L_o(h)$
9	$L_o(h+24)$
10	$Temp\,(h - lag)$
11	$\frac{1}{24}\sum_{i=lag+1}^{lag+24} Temp(h-i)$
12	$\frac{1}{24}\sum_{i=lag+25}^{lag+48} Temp(h-i)$

where h = hour of the year (1 to 8760), $L_o(h)$ = level of occupancy at time h, and $Temp(h)$ = temperature at time h.

The occupancy indicators are created as described before. The temperature is cross-correlated to find a lag constant. Furthermore, two signals are created from this delayed input. The first derived input is a 24-hour moving average. The second signal is the same moving average but with a time delay of another 24 hours. For the daily data set (test building 5), the following transformed inputs are used.

Transformed inputs for global training (NS GRNN and NS BP) with daily sampled data		Transformed inputs for local training (CBR) with daily sampled data	
1	$-\cos(2\pi\,[d + 355.5 \text{ days}] / 365)$	1	Day of the year (1 to 365)
2	$-\sin(2\pi\,[d + 355.5 \text{ days}] / 365)$		
3	$L_o(d-1)$	2	$L_o(d-1)$
4	$L_o(d)$	3	$L_o(d)$
5	$L_o(d+1)$	4	$L_o(d+1)$
6	$Temp(d)$	5	$Temp(d)$
7	$Temp(d-1)$	6	$Temp(d-1)$
8	$\frac{1}{7}\sum_{i=1}^{7} Temp(d-i)$	7	$\frac{1}{7}\sum_{i=1}^{7} Temp(d-i)$

where d = day of the year (1 to 365), $L_o(d)$ = level of occupancy at day d, and $Temp(d)$ = average temperature at day d.

The temperature variable, $Temp(d)$, does not have a lag since the lag in buildings is typically in the order of hours. Such lags are smaller than the sample frequency and are therefore negligible.

The inputs and the output are scaled according to Equation 2 for all the methods. No prior weighting is applied at the first trials.

Since scaling influences the selection of the training subset with CBR, prior weighting of the inputs could have a significant contribution to the accuracy of the CBR models. In this study, no algorithm is designed to find such weights. However, for NS BP, the software package includes a module to evaluate a contribution factor of each input that is a rough measure of the importance of that input in estimating the load. Although one cannot compare contribution factors of different models or functions, an extra trial is performed with each data set to check if there is a significant performance change when these contribution factors are used as prior weights to the CBR inputs. This is only performed with the CBR GRNN method.

Since the inputs of the CBR models and the NS BP for global training with hourly sampled data are not the same, the following conversion is used to come up with prior weights.

$$u_i = \begin{cases} \dfrac{v_1 + v_2}{\sum v_j} & i = 1 \\ \dfrac{v_3 + v_4}{\sum v_j} & i = 2 \\ \dfrac{v_{i+2}}{\sum v_j} & i > 2 \end{cases} \tag{6}$$

where

v_j = the jth NS model's input contribution factor (note that the NS models have 10 inputs whereas the CBR model has eight inputs)

u_i = the ith CBR input contribution factor

Once the conversion of contribution factors from NS models to CBR is completed, the next step is the normalization of CBR contribution factors as expressed in Equation 7:

$$w_i = \frac{u_i}{\max\limits_{1 \le j \le N} |u_j|} \tag{7}$$

where w_i is the ith normalized CBR input contribution factor.

The validation data set for the NS models is randomly selected from the CBR training set. The remaining data points are designated as the NS training set. The validation set is approximately 10% of the CBR training set.

Simulated Data (Test Building 1)

The purpose of this simulated data set is to test the different methods on regular data. The simulated data are generated with a very simple model that resembles the behavior to the

TABLE 1
Information on the Simulated Data Set

Sample frequency	Hourly
Size training set (NS)	7635 data points
Size validation set (NS)	850 data points
Size training set (CBR)	8485 data points
Size test set	1259 data points
Mean of load in training set	775.5 (tons)
Standard deviation of load in training set	1290.44 (tons)
Ratio Std./Mean	0.60

data set of test building 2 but without any noise. More details about the model can be found in Breekweg (1996). Table 1 gives some information on the data set. The training set covers a whole year of data. The test set follows directly after the training set. For the NS models, the training set is divided into two subsets, one smaller training set and a validation set. The unit of mean load and standard deviation is in tons of refrigeration (1 ton = 3.52 kW). The same units of loads are used for all test buildings.

Table 2 shows the performance of the different models. For the CBR methods, no training and testing time is available since training takes place for every test vector separately. The proposed CBR model with GRNN has the lowest CV and RCV values. The total time for training and testing is far higher for the NS models and the CBR BPLM model. For the BP models (indicated with †), this is the total training time of five trials. The genetic algorithm that is used to find the best smoothing parameter for the NS GRNN model (designated with *) is causing the long training time for this model. It is interesting to see that even the CBR method with linear interpolation performs better than the NS models. Apparently the selection of the training subset with CBR performs well enough to get a good approximation even with linear interpolation.

Table 3 shows the improvement in performance when contribution factors found with the NS BP model are applied as prior input weights for the CBR GRNN model.

The weighting shows a small performance improvement, indicating that weighting could result in a better performance (Table 4). It must be noted again, however, that the weights used are probably not the optimal weights since they were derived from a different model. The contribution factors of the NS models are apparently improving the performance on this

TABLE 2
Results of the Benchmark for Simulated Data Using Six Different Models

	MBE	CV	RCV	Training Time	Testing Time	Total Time
NS GRNN Global*	-1.35	16.45	8.11	2 h, 18 m, 16.0 s	1 m, 35.4 s	2 h, 19 m, 51.4 s
NS BP Global†	-2.69	22.96	10.18	2 h, 27 m, 4.8 s	1 m, 47.1 s	2 h, 28 m, 51.9 s
CBR GRNN Local	1.10	11.03	5.55	X	X	36 m, 6.2 s
CBR BPLM Local†	1.98	15.48	6.84	X	X	1 h, 57 m, 12.1 s
CBR Linear Local	1.77	14.60	7.04	X	X	14 m, 2.9 s
CBR Average Local	-0.29	18.78	11.51	X	X	13 m, 51.9 s

* The long training time of this method is caused by the genetic algorithm.
† The training time of these methods is the total training time over five trials.

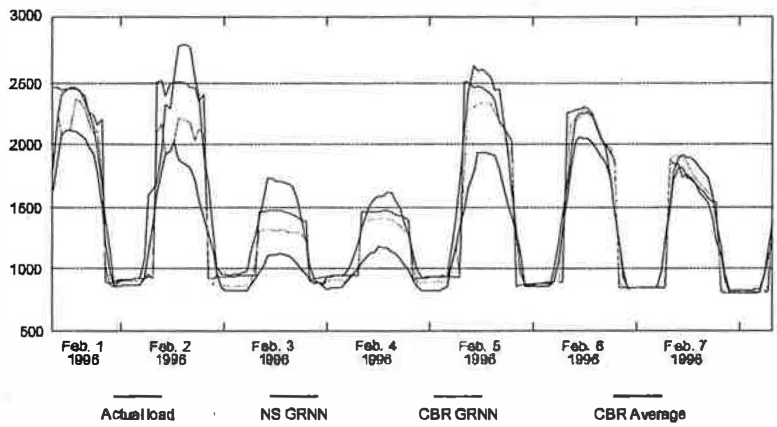


Figure 6 *Graphical representation of a section of the test set of the simulated data set. The actual (or simulated) data and the corresponding estimations of three of the models are shown.*

TABLE 3
Results of the Benchmark for Simulated Data With and Without Prior Input Weights

	MBE	CV	RCV	Total Time
CBR GRNN Local $w_i = w_j$	1.10	11.03	5.55	36 m, 6.2 s
CBR GRNN Local $w_i \neq w_j$	0.56	10.10	5.13	35 m, 56.3 s

data set. These contributing factors give a reasonable indication for the importance of the inputs, independent of the model or function used.

Although the RCV values are quite good, they are still higher than expected for simulated data. The CV values are relatively poor given that the goal of the accuracy is set to a 5.0% CV value. The lowest CV value is more than double the expected value of 5%. Taking into consideration that this data set is a simulated data set without noise, this suggests a poor performance for all models. The relatively poor performance is probably caused by two factors.

- First, the function used for the simulated model is not differentiable. The discontinuities at 7:00 in the morning and in the evening are causing extra approximation errors, reducing the overall performance. This can also explain the large difference between the CV and the RCV values, since extreme errors are omitted with the RCV measure.
- The second reason for the relatively poor performance is that the selected inputs and input translation do not fully match the variables in the underlying function. There is, for example, no influence of the occupancy levels of the previous and next days on the simulated output; however, these variables are used as inputs.

Test Building 2

This building has approximately 65 floors and consists mainly of offices and a data processing center. The system used in this building is not conventional. The building is run solely with chillers and has five chillers. Two chillers with a capacity of 1500 tons provide both cooling and heating. For heating, the chiller condensation water is used. The other chillers are used for cooling only. One of these chillers is exclusively for the data processing center, while another one is only used as a summer chiller. Although not conventional, the heating by the chiller's heat losses may make sense if the heating demand is low and cost savings are obtained by not having heating equipment. The occupancy profiles are characterized by a weekday schedule of 08:00 to 18:00.

This data set has the same input variables as the simulated data set. The training, validation, and test set have the same inputs. The output (or load) is different and has an unknown function. The training set covers a whole year of data. The test set follows directly after the training set. For the NS models, the training set is divided into two subsets, one smaller training set and a validation set.

Since the input data are the same as for the previous set, the CBR methods will select the same training subset as in the previous case. The only difference is the underlying function. Table 5 shows the results.

The result is significantly worse than with the simulated data. The real data from building 2 are clearly harder to model. This can be caused by several factors. First, this data set has noise that cannot be estimated, thus causing the accuracy to be less. Furthermore, it could be that some external influence is not incorporated in the model. In this case, it is known that the solar radiation has a significant influence on the building. This variable is not included as an input. This could be done at a later stage. A third factor that can cause this poor performance is that the training set and the test set do not have the same input dependency because of changes in building usage or system changes.

TABLE 4
Information on Test Building 2 Data

Sample frequency	Hourly
Size training set (NS)	7635 data points
Size validation set (NS)	850 data points
Size training set (CBR)	8485 data points
Size test set	1259 data points
Mean of load in training set	1587.46 (tons)
Standard deviation of load in training set	1129.63 (tons)
Ratio Std./Mean	0.71

TABLE 5
Results of the Benchmark for Test Building 2 Using Six Different Models

	MBE	CV	RCV	Training Time	Testing Time	Total Time
NS GRNN Global*	1.65	20.65	14.72	2 h, 21 m, 59.8 s	1 m, 42.2 s	2 h, 23 m, 42.2 s
NS BP Global†	1.03	18.53	12.67	2 h, 25 m, 42.6 s	2 m, 0.1 s	2 h, 27 m, 42.7 s
CBR GRNN Local	2.93	23.31	14.82	X	X	35 m, 1.8 s
CBR BPLM Local†	1.57	27.04	16.57	X	X	1 h, 55 m, 43.8 s
CBR Linear Local	7.53	26.53	16.53	X	X	13 m, 44.9 s
CBR Average Local	1.21	19.44	14.06	X	X	13 m, 36.3 s

* The long training time of this method is caused by the genetic algorithm.
† The training time of these methods is the total training time over five trials.

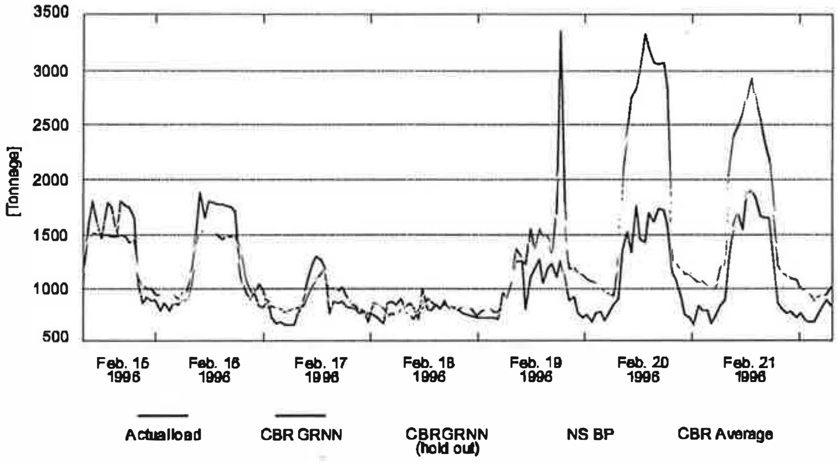


Figure 7 *Graphical representation of a section of the test set of test building 2. The actual data and the corresponding estimations of three models are shown.*

With this data set, both NS model methods perform better than the first three CBR methods. Furthermore, the CBR methods with (non)linear interpolation are performing worse than the CBR method with an average as the estimation phase.

Two extra tests are performed with this data set. The first extra test was also performed on the previous data set. That is, the contribution factors obtained with the NS BP model are used to weight the inputs of the CBR GRNN model. The second test was performed by choosing an optimal smoothing parameter for GRNN using a hold-out method. The hold-out method is explained in the next paragraph. Table 6 gives the results of these two extra tests compared to the CBR GRNN performance.

With the hold-out method, the smoothing parameter is found by taking out one vector from the training set at a time and calculating the error of the GRNN on the vector taken out, given a certain smoothing parameter. Next the vector is put back in the training set, a new vector is taken out, and a new error calculation is performed. This is continued until the last vector of the training set is put back in. The smoothing parameter is computed to minimize the SSE over all the training vectors. This is very time consuming. Therefore, the hold-out method is only performed on the nearest five training vectors. Compared to the method used in this study, these five vectors are not taken out at the same time but in sequential order.

The prior weighting does not improve the model nor does the hold-out method. Even though the hold-out method is only performed on the nearest five vectors, calculation time is very long.

Figure 7 clearly illustrates that the CBR GRNN models have a large estimation error for February 20 and 21. The CBR average model performs better. After checking the training subset for these days, an explanation can be given. The training subsets for these days include several load values that are indeed in the range from 2750 to 3500 tons. These values are weighted high since their Euclidean distance to the test vector was small compared to the vectors that had values in the range 1250 to 2000 tons. The CBR average model can still perform better since all selected load values are weighted equally. In the previous year (training set), unexpected higher loads occurred under the same input conditions as in 1995. Since this is only a small part of the total training set, the global models (NS) were not influenced as much by these relatively high load values. The local models with interpolation suffer considerably from such outliers since these outliers can be a large percentage of the selected subset.

TABLE 6
Results of the Benchmark for Test Building 2 With and Without Prior Weighting of the Inputs

	MBE	CV	RCV	Total Time
CBR GRNN Local $w_i = w_j$	2.93	23.31	14.82	35 m, 1.8 s
CBR GRNN Local $w_i \neq w_j$	2.47	24.27	15.07	36 m, 3.1 s
CBR GRNN Local $w_i = w_j$ hold-out	2.81	23.44	14.88	2 h, 44 m, 43.6 s

Test Building 3

Test building 3 is a 100-story commercial office building located in downtown Chicago. Retail/ commercial offices occupy the building's lower 44 floors while the upper floors have a total of 700 apartments. The total floor area of retail space is about a million square feet. The building's central plant serving retail space cooling needs consists of a total of seven chillers with a total capacity of 6000 tons. Although the building has a modern BAS, the building operator primarily selects the chiller operation.

The whole data set only covers four months. Therefore, the training set and the test set are selected from the same period. Several full weeks are selected as the test set. Test data are never located at the end or beginning of a time series to avoid extrapolation. The rest of the data form the training set. For the NS models, the training set is again divided into two subsets, one smaller training set and a validation set.

This data set is clearly smaller than the previous sets. The total set includes a total of 3000 data points. Unfortunately, approximately 1000 of these data points seem to be corrupted, or at least questionable, based on graphical analysis. Therefore, these data points are omitted from both the training and test set.

The results are shown in Table 8. The first thing that attracts attention is the very large MBE and CV values for CBR with linear interpolation. This is caused by the fact that the equation $\bar{x} = \bar{b}/A$ is underdetermined in those circumstances, at least to the working precision of the computer.

The RCV value is not influenced by these errors, since 10% of the worst estimations are omitted. The CBR GRNN method performs well compared to the other models with respect to the RCV value. Both NS models, especially the BP method, are clearly worse than the CBR methods. The extra model designed in a previous study (Breekweg and Gruber 19996) performs similarly to the NS GRNN model.

The results of the extra test with weighted inputs are given in Table 9. Here the contribution factors found with the NS models seem to give a good general indication of the importance of the individual inputs to the load independent of the model.

Figure 8 clearly shows that the CBR linear estimation (the line that goes out of the scale) has some very poor estimations. These are probably caused by the singularity of the matrix A in the equation $\bar{x} = \bar{b}/A$. Apart from these outliers, the linear model seems to follow the load, although not very accurately. The NS GRNN model has problems estimating the load on Tuesdays, Wednesdays, and Thursdays. These dates are August 7, 8, and 9 in Figure 8. There was no apparent reason for such performance on these specific days. The other days are estimated with about the same accuracy as the CBR GRNN models. The actual load of the test building seems to be inconsistently high beginning the night of Friday, August 10, and then extending to Saturday, August 11, 1995. This estimation error should not be considered as a model error. Instead, this is a good example of the purpose of benchmarking. It can be concluded, from the difference between the benchmark signal (e.g., CBR GRNN) and the actual load, that the load is high considering the given input variables. The operator should try to find an explanation for this high load. There is a similar load during the next night. Moreover, here is a peak load. It seems to be rather odd to have a peak load on a Saturday or Sunday night. Starting on Sunday evening (August 12, 1995) all models are estimating a far higher load than the load that occurred in reality. All models

TABLE 7
Information on Test Building 3 Data

Sample frequency	Hourly
Size training set (NS)	898 data points
Size validation set (NS)	100 data points
Size training set (CBR)	998 data points
Size test set	982 data points
Mean of load in training set	689.49 (tons)
Standard deviation of load in training set	353.59 (tons)
Ratio Std./Mean	0.51

TABLE 8
Results of the Benchmark for Test Building 3 Using Seven Different Models

	MBE	CV	RCV	Training Time	Testing Time	Total Time
NS GRNN Global*	-2.25	35.05	17.87	46 m, 23.7 s	18.4 s	46 m, 42.1 s
NS BP Global†	1.64	43.34	30.36	37 m, 12.3 s	16.1 s	37 m, 28.4 s
CBR GRNN Local	2.44	24.07	9.27	X	X	12 m, 20.5 s
CBR BPLM Local†	-0.75	21.45	12.68	X	X	53 m, 38.9 s
CBR Linear Local	-165.13	3952.97	15.35	X	X	2 m, 54.2 s
CBR Average Local	-0.45	26.42	14.37	X	X	2 m, 50.1 s
NS GRNN Global†	1.09	35.07	16.74	26 m, 36.8 s	12.9 s	26 m, 49.7 s

* The long training time of this method is caused by the genetic algorithm.
† The training time of these methods is the total training time over five trials.

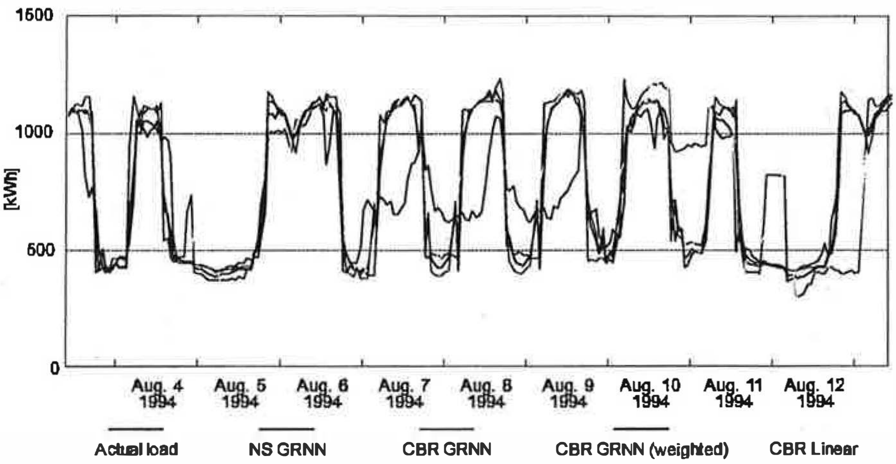


Figure 8 *Graphical representation of a section of the test set of the test building 3 data set. The actual data and the corresponding estimations of four of the models are shown.*

TABLE 9
Results of the Benchmark for Test Building 3 With and Without Prior Weighting the Inputs

	MBE	CV	RCV	Total Time
CBR GRNN Local $w_i = w_j$	2.44	24.07	9.27	12 m, 20.5 s
CBR GRNN Local $w_i \neq w_j$	1.66	22.11	8.30	12 m, 53.1 s

estimated pre-cooling starting on Sunday to get the building at the right temperature on Monday morning. A week earlier, precooling did occur, but on this particular Sunday it did not. In the training set, precooling on Sundays or early Monday morning was used in all but one case (under similar weather conditions). If the training and test sets were chosen differently, the estimation might have been different again. The time precooling started, if it took place at all, differed as well. Some days precooling would start at 8 o'clock on Sunday evening and sometimes at 3 o'clock on Monday morning. Many times the load would drop considerably again between 7 o'clock and 8 o'clock on Monday morning. Apparently, the building operator overestimated the load needed to precool the building. Sometimes, early Monday morning, the load would be the peak load of the day. The above observation further emphasizes the need for quality training data for obtaining good prediction accuracy from a neural network. The training data should contain adequate samples to capture the general behavior of the building load characteristics.

The overall problem with test building 3 is the inconsistency or high noise in the data set. This is probably caused by the building being mainly operated manually. Since there is more than one building operator, the load characteristics could be influenced by the difference in operation. One could say that the input variable "human behavior" is not incorporated and will probably never be incorporated. Furthermore, it might be that an important input variable is not available, which could explain the high noise level.

Test Building 4

The Zachry Engineering Center is a multipurpose building located on a Texas university campus. The building contains classrooms, laboratories, faculty/staff offices, and a large central computer facility. The building has four floors and was built in the early 1970s. The building can be characterized as a high-mass structure. The occupancy has a weekday schedule of 08:00 to 19:00 hours. Furthermore, the building is used in the evening hours between 19:00 and 00:00 hours and is moderately used during weekends. Additional information on the building can be found in Haberl et al. (1993). The data were obtained from the LoanStar program carried out in Texas. Additional information concerning this program can be found in Claridge et al. (1991).

In the previous study (Breekweg 1995), this data set was used exclusively. Therefore, the model designed during that

TABLE 10
Information on the Zachry Engineering Center Data

Sample frequency	Hourly
Size training set (NS)	1500 data points
Size validation set (NS)	169 data points
Size training set (CBR & Combi Net)	1669 data points
Size test set	1093 data points
Mean of load in training set	654.76 (tons)
Standard deviation of load in training set	154.78 (tons)
Ratio Std./Mean	0.24

TABLE 11
Results of the Benchmark for the Zachry Engineering Center Using Seven Different Models

	MBE	CV	RCV	Training Time	Testing Time	Total Time
NS GRNN Global*	0.71	4.75	5.63	1 h, 17 m, 44.1 s	1 m, 10.1 s	1 h, 18 m, 54.2 s
NS BP Global†	1.34	6.98	7.77	1 h, 40 m, 52.8 s	1 m, 23.0 s	1 h, 42 m, 15.8 s
CBR GRNN Local	-0.30	4.09	4.68	X	X	18 m, 14.3 s
CBR BPLM Local†	-1.05	4.43	4.35	X	X	1 h, 40 m, 32.8 s
CBR Linear Local	-0.52	4.68	5.84	X	X	5 m, 1.1 s
CBR Average Local	-0.20	5.96	7.42	X	X	4 m, 54.9 s
Combi Net BPLM Global†	0.01	3.79	3.76	37 m, 12.3 s	16.1 s	37 m, 28.4 s

* It is suspected that long training time for this method is caused by the genetic algorithm.
† The training time for these methods is defined as the total training time over five trials.

study is used as a comparison in this study. Since the training set of the previous study differs slightly from the training set for this study, a new training phase was performed with the new training set. The whole data set covers six months of data. The last two months are, however, discarded since it was known that the building had some major changes over Christmas. Several full weeks were selected as the test set. Test data is never located at the end or beginning of a time series to avoid extrapolation. The rest of the data forms the training set. For the NS models, the training set is divided into two subsets, one smaller training set and a validation set.

Table 11 shows the results from the test with the Zachry Engineering Center. The results from these tests are much better than those on the previous data sets. The underlying function is captured well. Only the NS BP and the CBR with the average phase seem to perform significantly worse. The best result is obtained with the Combi Net. This model clearly captures the underlying function better than the other models. This is not very surprising since this model was specially optimized for this data set. The other models have to perform well without optimization on the given data set.

Also for this data set, the contribution factors found with NS models are used for weighting the inputs. Table 12 shows the results.

For this data set, the use of weights results in a better performance. The results of the CBR GRNN with prior weighting is even slightly better than the performance of Combi Net.

TABLE 12
Results of the Benchmark for the Zachry Engineering Center With and Without Prior Weighting the Inputs

	MBE	CV	RCV	Total Time
CBR GRNN Local $w_i = w_j$	-0.30	4.09	4.68	18 m, 14.3 s
CBR GRNN Local $w_i \neq w_j$	-0.25	3.40	3.59	17 m, 48.7 s

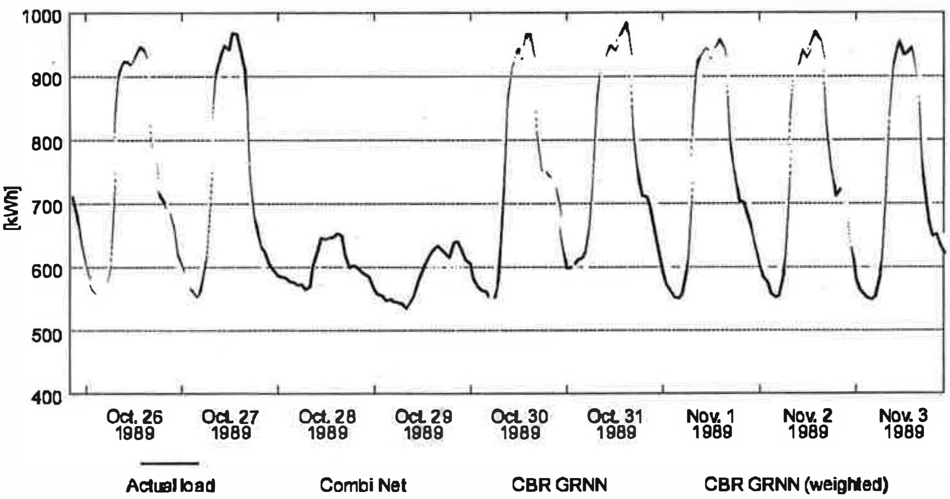


Figure 9 *Graphical representation of a section of the test set of the Zachry Engineering Center data set. The actual data and the corresponding estimations of three of the models are shown.*

Compared to the other data sets, this data set has a very low noise level and is fairly easy to model. The CBR BPLM model and especially the NS BP model suffered from outliers. The NS BP model probably ended up in a local minimum. The CBR BPLM model got stuck in local minima as well. Since, this model trains for every test vector, the influence of a local minimum is only noticeable for this test vector.

Test Building 5

This building is a medium-sized bank building located in Switzerland. The building total floor area is about 31,000 ft^2 and it has only 6000 ft^2 of cooling space. The building had a total consumption of 468 MWh for natural gas and 478 MWh of electricity at the time of this study. The main difference between this data set and the previous data sets is that it consists of daily time-series rather than hourly time-series. The training set consists of a little over one year of data. This training set is, however, split up into a smaller training and validation set. The test set consists of data collected during the following year. The results for this data set are shown in Table 14.

Here the best results are obtained with the CBR BPLM model. But one cannot speak of a significantly better result. The result of the test with weighted inputs is shown in Table 15.

For this data set, the results with weighted inputs are slightly better and comparable with the performance of the CBR BPLM model.

Figure 10 illustrates that the Swiss national holiday on August 1 is followed by all four models. The difference in performance occurs at the end of August and the beginning of September. All the models, specially the NS GRNN models, estimate the load too high.

TABLE 13
Information on Test Building 5 Data

Sample frequency	Daily
Size training set	345 data points
Size validation set	40 data points
Size test set	371 data points
Mean of load in training set	1761.84 (tons)
Standard deviation of load in training set	411.94 (tons)
Ratio std./mean	0.23

TABLE 14
Results of the Load Estimation for Test Building 5 Using Six Different Models

	MBE	CV	RCV	Training Time	Testing Time	Total Time
NS GRNN Global*	1.88	8.02	8.95	1 m, 51.1 s	7.1 s	1 m, 58.2 s
NS BP Global†	1.79	7.06	7.91	11 m, 0.5 s	7.4 s	11 m, 7.9 s
CBR GRNN Local	1.53	7.19	7.25	X	X	2 m, 6.6 s
CBR BPLM Local†	1.67	6.89	7.06	X	X	15 m, 12.7 s
CBR Linear Local	2.19	7.77	8.45	X	X	18.0 s
CBR Average Local	1.70	7.15	7.62	X	X	16.6 s

* The long training time of this method is caused by the genetic algorithm.
† The training time of these methods is the total training time over five trials.

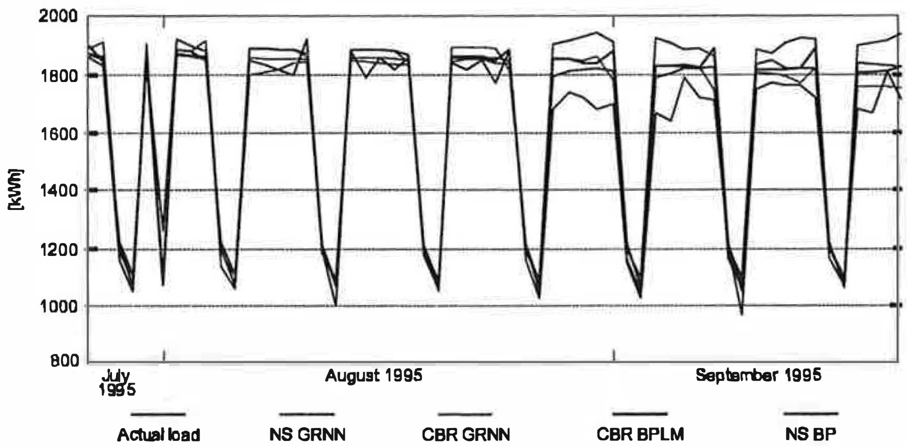


Figure 10 *Graphical representation of a section of the test set of the test building 5 data set. The actual data and the corresponding estimations of four of the models are shown.*

TABLE 15
Results of the Benchmark for Test Building 5 With and Without Prior Weighting the Inputs

	MBE	CV	RCV	Total Time
CBR GRNN Local $w_i = w_j$	1.53	7.19	7.25	2 m, 6.6 s
CBR GRNN Local $w_i \neq w_j$	1.61	6.91	7.07	2 m, 3.5 s

After analyzing the training set in the same time period (in 1994), it can be concluded that the underlying function of the training set and the test set are different given similar inputs. The load was approximately 100-200 kWh/h higher in the previous year under similar conditions. Although during the rest of the year (except for April/May) estimations are better, it might also be caused by a lacking input variable.

CONCLUSIONS AND RECOMMENDATIONS

The goal was to design a model for the energy usage of commercial buildings as a function of weather conditions and building occupancy. Furthermore, it must be possible to implement the model on any given data set of a building without going through an expensive redesigning phase.

Conclusions

At the beginning of this study, expectations of the possible performance of such a benchmark model were based on the previous study results obtained with the Zachry Engineering Center data set. In the absence of a qualitative measure of how accurate a model has to be, a performance goal was set based on the Zachry data. Since this data set was the first data set that was available, there were high expectations for modeling buildings in general. The accuracy of the benchmark in the previous study was very promising. The accuracy was well under a 5.0% CV margin. Therefore, a maximum CV value of 5.0% was set as a target.

Table 16 gives an overview of the accuracy of the benchmark test using the CV. From these data, it is clear that this target is not reached for any of the data sets other than the data from the Zachry Engineering Center.

TABLE 16
The CV Results of the Five Benchmark Tests

CV	Sim. Data TB 1	TB 2	TB 3	TB 4	TB 5
NS GRNN Global	16.45	20.65	35.05	4.75	8.02
NS BP Global	22.96	18.53	43.34	6.98	7.06
CBR BPLM Local	15.48	27.04	21.45	4.43	6.89
CBR GRNN Local $w_i = w_j$	11.03	23.31	24.07	4.09	7.19
CBR GRNN Local $w_i \neq w_j$	10.10	24.27	22.11	3.40	6.91
Combi Net BPLM Global			35.07	3.79	

Apparently, the Zachry Engineering Center data set is very consistent and has a underlying energy usage function that is estimated well with this model and the inputs (and the model of the previous study). The other data sets, especially for test buildings 2 and 3, are much harder to model. Unfortunately, buildings differ greatly and data sets are often not as easy to model as the data set from Zachry Engineering Center.

From these tests, several conclusions can be made.

1. *Data sets.*
   - The differences between building data are already relatively large with only four data sets at hand. This is not enough to state if it is possible to design one general applicable model. More data sets must be collected.
   - Not only is there a lack of data sets but most of the data sets used are also too small in size. To build a reliable data-driven model, enough data must be available. Therefore, new data sets must have enough data to make a good model, especially if there seems to be a lot of noise. For example, with the test building 3 data, many data are needed to compensate for this noise. This data set was also a bad data set in the sense that many data points had to be omitted since they were corrupted. One-third of the data set was omitted because these values were far from what was to be expected or did not include enough input dimensions.
   - Three of the four buildings operate mainly during office hours since they are office buildings. The other building, the Zachry Engineering Center, operates with a similar schedule. Other building types, such as hospitals, factories, hotels, or swimming pools, are not tested at all. Besides the purpose of the buildings, one can also select on the basis of location.
   - One could say that the lack of building data is the most significant shortcoming of this study.
2. *Human influences.* Test building 2 is operated mainly manually, causing human behavior to be incorporated in the energy usage function. It can be expected that manually operated buildings will be hard to model since the human factor is hard to incorporate. However, it is possible to use the tests on such data sets to convince the building owner that the building is operated far from optimally, and money could be saved by using automated control.
3. *Different models.*
   - The BP methods are not robust enough to come up with the same performance every time. This is due to local minima. The performance depends too much on the initial values of the neural network. This is also the reason why BP methods were trained five times. The CBR GRNN model does come up with the same result at every trial. The results of the NS GRNN method are influenced by the breeding pool

of smoothing parameters chosen for the genetic algorithm.

- None of the models is clearly better than the others in performance. For some tests, the best performance is obtained with the NS models and for others, with the CBR methods. When computation speed is taken into consideration as well, the CBR GRNN model performs well compared to the others. The question is, however, are the results of this model satisfactory? They certainly do not meet the goal of this study.
- The CBR module clearly makes the model faster if GRNN, linear interpolation, or an unconditional average is used. A clear benefit of local training over global training with respect to a better capability to adapt to local features cannot be proven.

Recommendations

1. The CBR part of the proposed models is based on a deterministic radius setting. This is far from ideal. It is probably worthwhile to build a model with the SCONN clustering algorithm since this algorithm seems to comply to the constraints set in Part I. With such a model, the Parzen density estimation and the relationship of the radius to that estimation are no longer needed. The design of such a model was not carried out since the SCONN clustering algorithm was not found until the end of this study.
2. To overcome the problem of dimensionality, dimension reduction could give some solutions. In Part I, the problem of dimensionality was discussed. Reduction of the input dimension through an algorithm should be a study by itself. One of the methods that could be useful is a (non)linear principal component analysis (PCA) using analytic methods and neural networks. In other studies, PCA has proved to be useful in dimension reduction. Using PCA or another technique to reduce the dimensions might have the following two advantages to the model proposed:
    - The number of dimensions is reduced and therefore the system becomes more reliable. Statements made are based on a smaller input space. This is, of course, only true if enough relevant information is kept in the reduced input space. The ideal situation would be if only redundant information is reduced.
    - The mapping from R^m to R^n could be done so that n (the number of dimensions after the dimension reduction) is a constant. This way the phases after dimension reduction are independent of the number of inputs before the dimension reduction. The second part of the system would become a lot more stable because it is not depending on the dimension anymore. In the case of a failing sensor, the number of measured input variables is reduced but the neural network still uses the same number of inputs.
3. The hourly benchmark signal is often less accurate at hours when the occupancy level changes (early in the morning and late in the afternoon during weekdays). The benchmark reacts too fast or too slow to the change in occupancy. For the simulated building 1 and test building 3, discontinuities caused such errors. These errors show up as peaks (positive or negative) in the error signal. Using filtering or calculating a daily average could reduce these kinds of errors, thereby improving the overall accuracy.

    Since there are not enough data available at the moment to design and test a general applicable model, it is not recommended to proceed with this project at the current time.

REFERENCES

Beale, M., and H. Demuth. 1994. Neural network toolbox for use with MatLab. Version 2.0a. Natwick, Mass.: The MathWorks Inc., January.

Breekweg, M.R.B. 1995. Energy management in commercial buildings using neural networks. Internship report, University of Twente, Enschede, the Netherlands.

Breekweg, M.R.B., and P. Gruber. 1996. Modeling the energy consumption of buildings by means of neural networks. *Conference Proceedings CESA'96: Symposium on Control, Optimisation and Supervision*, July, vol. 1, pp. 450-455.

Breekweg, M.R.B. 1996. General energy performance auditing of commercial buildings using non-linear modeling. Master's thesis. University of Twente, Holland.

Claridge, D., J. Haberl, D. O'Neal, W. Heffington, D. Turner, C. Tombari, M. Roberts, and S. Jaeger. 1991. Improving energy conservation retrofits with measured results. *ASHRAE Journal*, vol. 33, no. 10.

Haberl, J.S., and S. Thamilseran. 1996. The great energy predictor shootout II—measuring retrofit savings—overview and discussion of results. *ASHRAE Transactions* 102(2): 419-435.

Haberl, J.S., J.D. Bronson, S.B. Hinchey, and D.L. O'Neal. 1993. Graphical tools to help calibrate the DOE-2 simulation program: Comparative 3-D surface plots improve the ability to view small differences between simulated and measured data. *ASHRAE Journal*, January, p. 27-28.

Hassoun, M.H. 1995. *Fundamentals of artificial neural networks*. Cambridge, Mass.: The MIT Press.

MacKay, D.J. 1994. Bayesian nonlinear modeling for the energy prediction competition. *ASHRAE Technical Data Bulletin*, 10(5): 16-25.

Ward Systems Group Inc. 1995. *NeuroShell 2*, 3d ed. Frederick, Maryland: Ward Systems Group, Inc., August.