ON-LINE BUILDING ENERGY PREDICTION USING ARTIFICIAL NEURAL NETWORKS

Jin Yang

A Thesis in

The Department

Of

Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

February 2004

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisisitons et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-91143-8
Our file Notre référence
ISBN: 0-612-91143-8

ABSTRACT

On-line Energy Prediction Using Artificial Neural Networks
Jin yang

A literature survey is provided to summarize the existing approaches to building energy prediction. The survey examines both the theory behind each prediction model and practical issues such as data pre- and post-processing. It also points out the pros and cons of each prediction method.

Artificial Neural Network (ANN) is identified in the survey as the most popular and effective way to predict building energy demand. The ANN theory is thoroughly reviewed in this thesis. In particular, the ANN prediction model is presented as a generalized nonlinear least squares method. In addition to discussing the architecture and training methods employed by an ANN, we also examine implementation issues such as how to select the input to an ANN through day-typing and how to remove data redundancy and reduce the dimension of the input vector space via principal component analysis (PCA).

While most of the existing ANN models for building energy prediction are static in nature, this thesis focuses on developing dynamic ANN models that can evolve over time. The dynamic ANN models developed in this thesis are capable of adapting themselves to unexpected changes in the incoming data. When the dynamic model is combined with an automated data acquisition system, it can be used to provide real-time online building energy prediction.

A number of experiments have been performed to test the effectiveness of the ANN models developed in this thesis. The experiments use both simulated data and real data in a commercial building and a lab. It is demonstrated that it is relatively easy to build a static ANN model to predict the building energy demand at certain hour when the input elements of the ANN consist of the environment and operational variables measured during the same hour. However, in practice, the environment and operational data associated with hour t is often not available until the $t+1$ hour. Thus, the prediction model must rely on time-lagged measurements. The experiments in this thesis show that including as many lags as possible and then using PCA to remove redundancy seems to be an effective strategy. This strategy also provides the basis for the dynamic ANN model developed in this thesis. Numerical experiments show that they are quite effective.

ACKNOWLEDGEMENT

I would like to thank CANMET Energy Technology Center – Varennes Natural Resources Canada for providing me the energy measurement data, and for their financial support. I am also grateful to my advisors, Dr. Radu Zmeureanu and Dr. Hugues Rivard, for their guidance.

This thesis is dedicated to my parents,

Chuanhou Yang and Yuhua Zhao, and my brother Chao Yang

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ANN	Artificial Neural Network
AR	Autoregressive
ARMA	Autoregressive Moving Average
ARMIMA	Autoregressive Integrated Moving Average
ARMAX	Autoregressive Moving Average with Exogenous input
DDCV	Dual-duct Constant Volume
DDVAV	Dual-duct Variable Air Volume
EWMA	Exponentially Weighted Moving average
GFS	Generalized Fourier Series
HVAC	Heating, Ventilation and Air Conditioning
LR	On-line Recursive
MLR	Multivariate Linear Regression
NLS	Non-linear Least Square
PCA	Principle Component Analysis
TFS	Temperature Fourier Series

1. INTRODUCTION

1.1 Motivation and Objective

The objective of this thesis is to develop a flexible and robust model that can be used to predict hourly building energy demand. The prediction model is established by utilizing energy demand data collected in the past. It is designed in a way that it is dynamic and adaptive in nature. That is, the model can evolve and adapt itself to new features in energy demand pattern as new data becomes available.

The prediction of building energy demand plays an important role in building management. An accurate and reliable energy prediction scheme can be used to help building managers understand the diurnal and seasonal variations of energy demands and the variation of energy demand due to change in scheduling and other retrofit measures. When combined with an automated energy data collection apparatus, a fully automated energy prediction system becomes a powerful tool for identifying maintenance problems that often contribute to unnecessary increase in energy use, and for determining the best energy control strategies (Dhar et al. 1999a). Due to this wide spectrum of demands and applications, the subject of developing an effective energy prediction scheme for residential and commercial buildings has received considerable attention in recent years.

1.2 Background

An automated energy prediction system is often built on top of a mathematical prediction model consisting of several parameters. The model parameters are estimated using existing data that typically include energy demand and temperature measurements recorded in the past. A variety of prediction models have been proposed in the literature. These models have been implemented and tested on various types of buildings. However, there is yet no consensus on which model is the best. One of the purposes of this thesis is to summarize and compare the existing building energy prediction models through an extensive literature survey.

The literature survey presented in this thesis covers a wide range of computational prediction models. These models include time-series models, Fourier series models, single and multiple regression models, Artificial Neural Network (ANN) models and Fuzzy logic models. Both the strength and pitfalls of each model are discussed. The survey provides not only the general design philosophy for each prediction model, but also some discussions on the implementation details and the performance associated with these models.

Among all the papers surveyed, a large number of them are concerned with using ANN models to predict either long-term or short-term energy use in commercial buildings. Therefore, in this thesis, we take a closer look at the theoretical and practical aspects of using ANN to predict energy usage.

With the exception of time-series models and a few ANN models, most of the surveyed literatures focus on *static prediction*, a prediction scheme that involves a single prediction model that does not evolve over time. In a static model, once the estimation of the model parameters is completed, the model is fixed. To obtain an accurate static model, a large volume of historical data is required to estimate the model parameters. A dynamic prediction model that constantly updates model parameters based on newly available energy measurement data alleviates the need to archive and to retrieve a large volume of historical data. As the energy data collection process is automated, the entire process of retrieving new measurements, updating the model and making short-term energy prediction can be performed in 'real time'. This dynamic prediction scheme will be referred to as *on-line* prediction in this thesis.

Two variations of ANN on-line prediction models are proposed in this thesis. They differ in the way past data is selected for learning. In the first model, the model parameters are obtained by accumulating historical data. The second model makes uses of a fixed volume of data shifted in time to estimate the ANN model parameters.

The performance of these two ANN models is tested on two data sets. The first data set consists of simulated energy data associated with a building in Montreal. The quantity to be predicted is the total electric demand (in kW) of the building. The second data set consists of real energy demand and temperature measurements collected from sensors installed in the HVAC system of the CANMET Laboratory in Varenes. Both the heating and cooling energy demand are predicted from a variety of environmental and operational data.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 contains a literature survey that examines the existing approaches to building energy prediction. The survey provides some background information on each prediction model, and discusses how each model is used in practical applications. Both the theoretical and the implementation aspects of each method are examined. A comparison of different methods is provided at the end of the chapter. In Chapter 3, the ANN model is presented in great details. The discussion focuses on the issues of how to choose appropriate input, the internal structure of the network, training methods, and adaptive ANN models. The relationship between ANN training and the classical methods for solving nonlinear least squares problems is presented. The technique of principal component analysis (PCA) used to reduce the dimension of the input and to remove redundancies in the data is also introduced. Two adaptive ANN on-line prediction models are introduced. Computational experiments are presented in Chapter 4 to demonstrate the effectiveness of ANN models on two data sets. Both static prediction models and adaptive (online) models are tested. The accuracy of the models is reported. The thesis ends with a conclusion, list of contributions and suggestions for future work.

2. LITERATURE SURVEY

Building energy prediction is a complex issue. Building energy demand depends on many factors such as climate conditions, building characteristics, and the type of heating, ventilation, and air conditioning (HVAC) equipment used. The quantities to be predicted may include electrical demand, thermal (heating and cooling) loads or energy demand. It is pointed out in (Katipamula et al.1998) that a reliable prediction model must be based on sound engineering principles. In this chapter, a literature survey is presented to provide an overview on the start-of-the-art models currently used to predict building energy demand. Prediction models proposed in the surveyed articles can generally be grouped into four categories: 1) Regression Models, 2) Time Series Models, 3) Artificial Neural Networks, 4) Other Methods. Each of these categories is described further below. The basic concepts behind each model as well as its strengths and weaknesses are described. How each model was used to make various building energy predictions is summarized, and the effectiveness of each model reported in the literature is also discussed. Most of the papers surveyed in this thesis use either the *Coefficient of Variation* (CV) or the *Root Mean Square Error* (RMSE) as the metric for measuring the accuracy of the prediction. The CV value of a prediction is defined as (Curtiss et al. 1995)

$$\mathrm{CV} = \frac{\sqrt{\dfrac{\sum_{t=1}^{n}[y_{pred}(t) - y_{data}(t)]^2}{n}}}{|\bar{y}_{data}|} \qquad (1)$$

where $y_{\text{pred}}(t)$ is the predicted energy use at time t, $y_{\text{data}}(t)$ is the measured energy use at time t, and $\bar{y}_{data}$ is the average of the measured data.

The RMSE value associated with a prediction is defined as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}[y_{pred}(t) - y_{data}(t)]^2}{n}} \quad (2)$$

Other criteria for measuring prediction accuracy such as Mean Bias Errors (MBE), R-Square (R^2) are presented in Appendix A. The chapter ends with the results of the shootout experiment, a comparison and a conclusion.

2.1 Regression Model

Regression analysis is a data fitting technique that makes use of a function constructed from realistic engineering principles to model a particular physical phenomenon. An independent variable of the function is often called a regressor. The function itself is called a regression model. A regression model usually contains a number of adjustable parameters called regression coefficients. These coefficients are chosen to minimize the discrepancy between the measured data and the corresponding values calculated with the regression model. Once these parameters are determined, the regression model can be used to predict desired quantities.

Regression models have been demonstrated to be effective for building energy predictions in a number of experiments (Fels 1986; Kissock et al. 1992; Ruch and Claridge 1992; Kissock et al. 1998; Katipamula et al. 1998). Usually regression models can be classified into two categories: a single variable regression model and a multivariate regression model. For instance, a single variable regression function relates the energy demand to the change of outdoor temperature (Fels 1986; Kissock et al. 1993), and a multivariate function takes climate conditions, building usage, system characteristics and the type of HVAC equipment used into account (Fels 1986; Boonyatikarn 1982; Sullivan and Nozaki 1984, Katipamula et al. 1998).

A single variable regression model based solely on ambient-temperature is described in detail in (Kissock et al. 1998) and the literature cited therein. It is pointed out that the use of an ambient-temperature regression model can reduce the influence of changing weather conditions on building energy demand so that retrofit savings may be more accurately measured. Another advantage of this single variable model is that it avoids the multicollinearity problems often encountered in multivariable regression models. Multicollinearity occurs when the regressors are linearly dependent. As a result, the least square solution is non-unique. This problem leads to model uncertainty.

Four regression functional forms (2-parameter, 3-parameter, 4-parameter, 5-parameter) are provided in (Kissock et al. 1998) along with the contexts in which they should be used. The authors pointed out that the selection of a regression function should be based on both the best-fit criterion (minimizing the least squares error) and the expected relationship between energy demand and weather for the particular heating and cooling system being considered. They also discussed the appropriate choice of a time-scale for data measurements used for accurately determining the regression parameter. A case study is provided in (Kissock et al. 1998) to demonstrate the effectiveness of the regression model. When the proposed prediction models are applied to an engineering center located in Texas A&M, both coefficient of determination (R^2) and coefficient of variation (CV) values associated with the prediction of cooling energy demand are shown to be equal to 80% and 8% respectively. For heating energy prediction, the R^2 and CV values are 88% and 26% respectively. Average daily savings are 44.0 ±5.7 GJ/day for cooling modeling, and 37.2 ±4.8 GJ/day for heating modeling.

A methodology for developing multivariate linear regression (MLR) models is proposed in (Katipamula et al. 1998). Several multivariate regression functions for predicting cooling energy demands are given for both dual-duct constant volume (DDCV) and dual-duct variable air volume (DDVAV) HVAC systems. The authors also discussed the choice of appropriate time resolution when hourly measurement data is available. For example, related operational parameters change each hour but they are generally constant on a monthly or daily basis. Therefore, monthly and daily energy demand show much less scattering compared to the hourly energy use. The MLR regression model is applied to five commercial buildings located in central Texas with four different time scales: Monthly, Daily, Hourly and Hour of Day. Coefficients of variation (CV) and mean bias errors (MBE) were used as criteria for accuracy evaluation. The values of CV ranged from 5% to 15% while the MBE values ranged from 2% to 12%. Daily energy prediction provided the best performance. Its CV and MBE values reach the lowest level among those predicted. Compared to single-variable models, MLR models typically have somewhere between 12% and 54% lower CV values. Based on these results, it is concluded in Katipamula's work (Katipamula et al. 1998) that MLR models - are more accurate than a single variable regression model.

2.2 Time Series Model

The pattern of building energy usage is typically cyclical due to earth rotation, weather conditions and human activities, which vary with hours in a day, and days of the year. From this point of view, a set of time series data is needed to understand the changes in energy usage. If the history of energy use is viewed as a time series, then it is natural to use standard time series analysis techniques to forecast energy use in future time. A time series model consists of a set of observations of a continuous-time variable (for example, energy use) measured at equally spaced time intervals (Harvey 1992). Time series models offer the advantage that they take into account the time while traditional statistics techniques tend to ignore the time dimension.

The most frequently used time series models are: autoregressive model (AR), autoregressive moving average model (ARMA), autoregressive integrated moving average model (ARIMA), autoregressive moving average with exogenous input model (ARMAX), exponentially weighted moving average (EWMA), and Fourier series model. Each of these models is described further in the sub-sections below.

2.2.1 Autoregressive (AR) model

An AR model assumes that the current predicted energy is linearly related to the energy use detected at an earlier time. A typical autoregressive model can be expressed by

$$y_t = \Phi_1 y_{t-1} + \cdots + \Phi_p y_{t-p} + \varepsilon_t, t = 1,2...T. \quad (3)$$

where y_t represents the energy demand at time t, p is the order of the autoregressive model, and Φ_i, i=1,2,...p, are coefficients to be determined, ε_t is the disturbance or white noise. This autoregressive model is often denoted by AR(p).

The advantage of this model is its simplicity in parameter calculation. But its success depends largely on the assumption that the present value is a linear combination of the previous ones. This assumption is not always valid in reality. In (Liu et al. 1996), AR is used and compared with other techniques. It is discovered that the prediction errors of an AR model increase as the order of the model is increased. This is especially true when model order is larger than five. However, it is also shown in Liu's paper that the AR model he developed is less accurate than a neural network or a fuzzy logic model when signal is highly nonlinear.

2.2.2 Autoregressive moving average (ARMA) model

The AR model introduced above can be extended to include a moving average of the predicted data to produce an autoregressive moving average (ARMA) model. An ARMA model equation can be written as

$$y_t = \Phi_1 y_{t-1} + \cdots + \Phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}, \quad t = 1,2...T. \qquad (4)$$

where the additional parameters $\theta_j, j = 1,2,...,q$ account for the moving average of past disturbance. This time series model is often denoted by ARMA(p,q). Both the AR and the ARMA time series models are considered to be stochastic models. The stochastic nature of these models is reflected from the random disturbance term ε_t. If the roots x of the characteristic polynomial

$$x^p - \Phi_1 x^{p-1} - \Phi_2 x^{p-2} - \cdots - \Phi_p = 0 \quad (5)$$

are bounded by 1 in absolute value, then both AR(p) and ARMA(p,q) are known to be stationary stochastic models. One of the characteristics of a stationary stochastic model is that the observation y_t is expected to fluctuate around a mean value μ (Harvey, 1992).

2.2.3 Autoregressive integrated moving average (ARIMA) model

In practice, a stochastic process is not always stationary. Because an ARMA model assumes that the time series fluctuate around a certain mean value, it cannot accommodate sudden changes in the dynamic process. An autoregressive integrated moving average (ARIMA) model can overcome this difficulty by incorporating differences of the observations in the model. The first difference between the observation y measured at time t and time t-1 is defined by $\Delta^1 y_t = y_t - y_{t-1}$. The d-th difference of y at time t can thus be defined inductively by $\Delta^d y_t = \Delta^{d-1} y_t - \Delta^{d-1} y_{t-1}$. If the d-th difference of the observation satisfies an ARMA(p,q) model, then the time series itself is said to satisfy an ARIMA model, and it is usually denoted by ARIMA (p,d,q) (Harvey 1992). This model is sometimes referred to as a Box-Jenkins model. Kimbara et al. experimented with the ARIMA model (Kimbara et al. 1995) and found the performance of ARIMA to be better than a two dimensional AR model.

2.2.4 Autoregressive moving average with exogenous input (ARMAX) model

An ARMAX model of energy prediction model consisting of the parameters output y_t, input u_k, and noise v_k is introduced in (MacArthur, et al. 1989). The model can be described by the following formula

$$A(z^{-1})y_t = B(z^{-1})u_k + C(z^{-1})v_k, \qquad (6)$$

where z^{-1} is the time delay or lag operator, defined by $(z^{-1})y_t = y_{t-1}$, and

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_p z^{-p}, \\ B(z^{-1}) &= 1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_p z^{-p}, \quad (7) \\ C(z^{-1}) &= 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_p z^{-p}, \end{aligned}$$

defines how delayed output, input and noise are combined. The coefficients $a_i, b_i, c_i, i = 1,2,...p$, are to be determined. This model represents a relationship between energy use and ambient temperature.

Several new models and applications have been implemented based on the ARMAX model. MacArthur and coworkers (MacArthur et al. 1989) presented an on-line recursive (LR) algorithm. Their paper provides a comparison between the predicted total daily load and actual daily load. The agreement appears to be quite satisfactory. The authors showed that typical prediction error was on the order of 5%. Hong-Tzer and coworkers (Hong-Tzer et al. 1996) combined an ARMAX model with a genetic algorithm to improve the accuracy of the prediction, the case study presented there demonstrates that this method provides a high accuracy.

2.2.5 Exponentially weighted moving average (EWMA)

Another way to combine time series data in a prediction model is to take an exponential weighted moving average of the time series using an exponential smooth constant λ (Seem and Braun 1991). Mathematically, the model can be described by

$$y_t = \sum_j \lambda(1-\lambda)^j y_{t-(j+1)\Delta}, \qquad (8)$$

where y_t is the predicted energy usage at time t, λ is an exponential smoothing constant, and Δ is the time step. The advantages of this model are: i) predictions can be made from the previous forecast and observation so that the whole set of history of past data is unnecessary, ii) less effort is needed to estimate the model parameters. EWMA is used in (Seem and Braun 1991) for predicting electric demand of a building in Madison during the twentieth week of the year. The results are compared with that generated from a harmonic model. It is concluded that a significant improvement can be achieved using EWMA.

2.2.6 Fourier series models

It has been observed that hourly energy use in almost all commercial buildings exhibits some periodic patterns (Dhar et al. 1998). This observation provides motivation for using a Fourier Series model to predict energy use. A Fourier series model decomposes the time series as a linear combination of sines and cosines with distinct frequencies. It is also referred to as frequency analysis of a time series model. A standard linear least squares (regression) technique can be used to determine the coefficients of linear combination to minimize the difference between the predicted and measured energy data. The number of terms used in the Fourier Seriers Model can be initially determined by Mallow's Cp criteria (Dhar et al. 1998). It can be reduced by using a reasonable partial R-square (residual square) cutoff.

It is pointed out in (Dhar et al. 1998) that to make effective use of a Fourier Series Model, energy data should be separated into different groups based on their distinctive energy usage characteristics. This is also known as day-typing. A separate Fourier Series Model is developed for each day-type (for example, weekdays and weekends). Different day types that do not show significant statistical difference may be aggregated after a statistical test known as Duncan's multiple range test (Ott 1988) has been performed.

Model equations are provided in (Dhar et al.1998 and Dhar et al.1999a,b) to fit both weather independent and weather dependent data. For weather independent data, the model equation consists of three parts. The first part uses hour of day as the independent variable; the second part uses day of year as the independent variable; and the third part contains a mixed term using both hour of day and the day of year as the independent variables. A case study involving a large institutional building that contains classrooms, labs, offices and computer facilities called the ZEC building from University of Texas A & M is used to demonstrate the effectiveness of the model. Coefficients of determination (R^2) are around 85% to 91% and coefficients of variation (CV) are around 6% to 15%. In particular, the authors discussed how energy data is grouped (into day-types) and the number of Fourier terms required to accurately predict energy usage.

For weather dependent data, such as hourly cooling and heating, two model equations have been developed in (Dhar et al. 1999b). If the outdoor dry bulb temperature, outdoor specific humidity and horizontal solar flux are available, a Generalized Fourier Series (GFS) model equation that takes these variables into account can be used. When humidity and solar data are unavailable, one can use a Temperature Fourier Series (TFS) model. The temperature dependent model must take into account 1) the nonlinear relationship between energy usage and outdoor temperature, and 2) the interaction between outdoor temperature variation and time (hour of day). The detail of this model is discussed in (Dhar et al. 1999a). Both model equations are applied to analyze energy usage in the ZEC building (Dhar et al. 1998 and Dhar et al. 1999a, b). The reported Coefficients of determination (R^2) values for different day types are between 85% and

92%, and the reported coefficients of variation (CV) values are between 6% and 21%. For cooling simulation, the reported R^2 values of a GFS model are a little higher than those of a TFS model, and the reported CV values are a little lower than those of a TFS model. For heating energy prediction, it is concluded that TFS models perform better than GFS models. This is probably because a TFS model only partially captures the variation of humidity and solar energy when cooling energy is predicted. It is well known that heating energy use should not be dependent on these variables. Both models compare favorably with the winners in the Predictor Shootout I (Kreider and Haberl 1994).

2.3 Artificial Neural Networks

Artificial Neural Networks (ANN) is a type of Artificial Intelligence technique. It can describe a nonlinear relationship between the input and output of a complicated system. The theory of artificial neural networks has evolved over the last fifty years and its development is motivated by the desire of trying to understand how the human brain works. Practical use of artificial neural networks has recently become common in scientific research and commercial applications. ANN is successfully used in the field of medicine, military systems, financial systems and power systems. It is also capable of solving different kinds of problems such as signal processing, pattern recognition, planning, control and search, image processing and computer vision.

Similar to a human neural system, an artificial neural network is an information processing structure that consists of a number of input units and output units connected in a systematic fashion (Kreider et al. 1995). Between the input and output units, there may be one or more hidden layers, each consisting of a number of units called neurons, nodes or cells. The connections between units lying on different layers are assigned with varying weights. Information processing occurs at neurons. Input signals (or data sets) are fed in from the input layer. They follow all possible connection paths to reach the next layer. Along each connection link, the signal is weighted by a weighting factor. Multiple weighted signals entering into the receiving neuron are then summed and passed through an activation function. An activation function represents a basic operation of one neuron: summing its weighted input and applying them to the output. The output from the activation function becomes the input for the next phase of network flow. Eventually, the weighted signals reach the output units. Then these weights are summed and passed through an activation function to produce the final output.

2.3.1 ANN static predictions

The main advantage of an ANN model is its self-learning capability. In Chapter 3, the mechanism through which an ANN learns is explained. Various design choices of an ANN and learning algorithms for improving the reliability and flexibility of an ANN prediction model are also examined.

The use of ANN in building energy prediction has been investigated by many researchers (Curtiss et al. 1993; Curtiss et al. 1994; Curtiss et al. 1996; Kawashima et al. 1995). Although the ANN models described in the surveyed articles share some similarities, they differ significantly in implementation details. Each article proposed an ANN that is tailored towards a specific type of energy prediction under a specific building environment.

Earlier studies (Curtiss et al. 1995; Curtiss et al. 1994) have demonstrated the success of the use of ANN as predictors for hourly building energy demands. These ANN models contain less than a dozen inputs. In one case, predictions are made to obtain the whole building energy use in the next three hours using measured energy demand in the immediate past one or two hours. The accuracy achieved is reported as Coefficient of Variation (CV) of 5% to 10%. In another case, prediction is made without the knowledge of the actual energy demand for the immediate past hours. This problem is known to be more difficult and the results appeared to be less accurate in general.

An alternative approach proposed by Curtiss et al. (Curtiss et al. 1995) makes use of internally generated "past" data during the training process of an ANN. To be more precise, the predicted results are fed back into the network to make future predictions. This approach is referred to as *recurrent neural networks*. They report that recurrent neural networks can provide accurate predictions for cooling and heating loads of a test building without using actual load from the immediate past few hours. An example is provided in (Curtiss et al. 1995) to compare the quality of a recurrent neural network with a non-recurrent neural network. The networks used dry bulb temperature, humidity ratio, horizontal solar flux, wind speed, hour of day, weekday /weekend binary flag as input. One or two hidden layers containing five or six neurons were employed. Both chilled and hot water loads were predicted. It is shown that recurrent neural network is effective, but not as accurate as the non-recurrent prediction obtained by training the neural network on measured data.

A case study that involved a large building at the University of Colorado is presented in (Anstett and Kreider 1993). An energy monitoring system called BEACON was installed in the building along with an artificial neural network (ANN) simulator. An optimized backpropagation method called flat spot elimination is adopted in the training. The training method appears effective for improving the prediction capability. Multiple networks are trained and evaluated using different network configurations and parameter settings. A series of independent variables such as month of year, scheduled hours, high outdoor temperature, low outdoor temperature, water temperature, and cafeteria sale are used as inputs for water, steam, electric and natural gas prediction. The prediction of electric demand and steam are shown to be quite accurate. The reported R^2 and RMS values are 96.9% and 0.7kWH respectively for electric demand, and 96.7% and 0.97LBs respectively for steam prediction. The reported RMS values associated with the water and natural gas prediction are 0.9 GAL and 0.09CCF respectively. The reported R^2 values associated with the water and natural gas prediction are 95.6% and 96.4% respectively. Based on these experiments, it is concluded that: i) the flat spot elimination method is useful for modifying the weights and is easy to implement; ii) the performance of each single network is better than combining the four networks together; and iii) increasing the number of input variables could improve the generalization ability of an ANN.

In (Kawashima et al. 1995), several hourly thermal load prediction techniques are tested and compared. ANN proved to be the most accurate model. The proposed ANN consists of fifteen inputs (seven ambient temperature measurements, seven solar insolation measurements obtained six hours before the prediction period, and the thermal load at time t-24). One hidden layer with thirty-one units and one output (thermal load) are used in this particular ANN model. A backpropagation training method is adopted. In the training process, hourly loads are updated by multiplying the ratio of the total predicted and observed loads. The learning rate is reduced gradually using a "three-phase annealing procedure" (Kawashima et al. 1994) to accelerate the process. A case study in (Kawashima et al. 1995) showed that the accuracy of the ANN model improves when more input variables are used.

In (Curtiss et al. 1996) it is shown that an ANN is also able to predict the pre-retrofit chilled water demand so that the energy savings between the pre- and post-retrofit periods can be calculated. In this case study, a six-month pre-retrofit data set is selected as inputs of the network. The network consists of one input layer with eight units, two hidden layers and one output layer. The input variables contain hour number (0-23), ambient dry-bulb temperature, horizontal insolation, humidity ratio, wind speed, weekday/weekend binary flag, past hour's chilled-water demand and second past hour's chilled-water demand. It is shown that there is no obvious discrepancy between the measured energy demand and the predicted values. It is also demonstrated that the designed ANN can predict energy demand for commercial building as well as for residential buildings.

In a more recent article (Curtiss et al. 1998a), an ANN was used to predict a 13-day chiller power demand and 5-day ice storage tank charging and discharging in HVAC plant systems. The building investigated is a 10,000 square feet commercial building with a maximum cooling load of 229 KW. Six inputs were selected for chiller power demand prediction: three past histories of three-way valve position, evaporator inlet temperature and evaporator outlet temperature, and ambient temperature. The architecture consists of these six input units with one hidden layer of five nodes and two outputs (thermal load and chiller power demand). Coefficient of variation (CV) and root mean square error (RMSE) are used as criteria for evaluation of prediction performance. For chiller load and demand modeling, the reported CV values are 6.82% and 12.1% respectively, and the RMSE values are 1.17 ton and 2.1KW. For ice tank discharging, the CV value is 13.1% and the RMSE value is 2.1 ton. These simulation results show that a neural network can model the equipment performance with fewer inputs and less computation time. It also has the ability of self-calibration.

A general framework is established in (BreekWeg, et al., 2000a, b) for selecting an appropriate neural network model to predict building energy demand. The authors examined various aspects of a neural network model including the choice of an activation function, the number of hidden layers and training methods, etc. The authors emphasized design issues related to the general applicability of a neural network model to building energy prediction. Due to the variation in data characteristics, the authors suggested that a general prediction model should not be completely data driven. They maintained that a general model should have the capability to preprocess the data to extract only the relevant information. They also pointed out that a good model should address the problems of missing or inconsistent data and take into account additional physical considerations of the building whenever possible. In these papers, a neural network is viewed as a nonlinear modeling technique for the energy demand function approximation. The authors pointed out that cased-based reasoning (CBR) can be used as a learning technique for local training. In CBR, a new training subset is chosen from the entire input data for each new test case. This local training technique is faster compared to the other algorithm described in this paper: clustering. With the clustering approach, the data set is partitioned into several disjoint subsets using some predefined similarity measure. In the presented case studies, five buildings are tested using the proposed prediction models. The performance results were mixed. The prediction model performs well for some models but poorly for others. However, the CV values were within 5% in most cases.

An ANN for predicting the hot water supply temperature of the same building in (Curtiss et al. 1998a) is discussed in (Curtiss et al. 1998b). To minimize the excess peak energy use, an ANN was used to predict the hot water supply temperature and boiler outlet temperature. Four input and two output variables are selected for the network. The input variables are: hot water supply temperature, boiler outlet temperature, boiler stage controller output (0,1,2,3), three-way valve controller output (Voltage~0%-100%). The output variables are: hot water supply temperature and boiler outlet temperature. The net has two hidden layers and used an inverted triangle method to determine the number of units in every hidden layer (Wasserman, 1989). For these two cases, the training data is collected every 15 seconds, input and output layers use linear activation function while the hidden layers use sigmoid activation function. In the two simulations, ANN works with PNN (predictive neural networks) controller to optimize the energy use and its performance is compared with that of a proportional integral and derivative (PID) controller. The authors showed that a PNN can understand which controller output were needed to maintain the setpoint. So the deviation of a PNN is smaller than that of PID controller.

Electricity use of whole power system is also an important part of energy demand. Applications of ANN predictor in the power system is also demonstrated to be effective by many researchers in (Alves et al., 2000; Bakirtzis et al., 1995; Drezga and Rahman 1999; Khotanzad et al. 1995; Kim et al., 2000; Lee et al., 1992;Lu et al., 1993, Mohammed et al.,1995; Yoo and Pimmel 1999). When selecting input variables, most of these researchers only use previously recorded load and temperature as major input. Compared to the input of HVAC prediction, the input selection for electric power system is much simpler. Some researchers use the load data of the past 24 hours, others use the load of previous day at the same time. Besides using recorded temperature at previous time, Khotanzad et al., 1995, Lu et al., 1993 and Mohammed et al., 1995 also used forecasted temperature of previous or current hour as input. Since date and hour are also factors that cannot be ignored, some researchers use sine and cosine to indicate the date and hour because energy use is periodic. Design of ANN architecture for electric power prediction is also simple. Most of these researchers used only one hidden layer, only a few of them used two hidden layers. They use Mean Absolute Percentage Error (MAPE) or Standard Deviation Percentage Error (Details can be found in Appendix A) to test the accuracy of the ANN predictors. It is shown that the results are quite satisfactory. Most errors are within 5%.

A heterogeneous artificial neural network that contains different ANN types is used for short-term load forecasting for different regions (Piras et al. 1996). The authors show that this model is better than a single model through the combination of the supervised part and unsupervised part. The supervised part of the model is used to analyze the process in sub-models capturing the local features in the data and suggesting regression variables. The unsupervised part called mutilayer perceptron provided the approximation of the underlying function. These two parts are combined to predict the short-term load in five European regions. The mean absolute percentage errors (MAPE) are used to evaluate the performance. It is shown that most of the forecasting errors are lower than 5%.

2.3.2 ANN on-line predictions

In most energy prediction applications, researchers use outside temperature, date, time, day types (weekday/weekend) as input variables, some of them also use solar radiation as input. Outputs predicted by these neural networks are hourly thermal load, electricity usage, chilled water use, etc. Usually one hidden layer is enough to make a prediction. Sometimes some researchers use two hidden layers. There is no published systematic way to decide the input variables and number of neurons in the hidden layer. Experience and knowledge are important for designing a good artificial neural network.

The literature review presented above primarily focused on static prediction. In the following paragraphs, the literature about dynamic or on-line predictions is summarized.

Compared to the number of static predictions, the number of papers on on-line predictions found in the literature is limited. Some examples have been found in the field of electric load forecasting for power systems.

An on-line algorithm consisting of supervised and unsupervised learning for short term ANN load forecasting is presented in (Djukanovic et al 1995). Thirty-eight input variables are selected. These variables include:

- maximum temperatures and minimum temperature for previous two days and the current day,
- twenty-four hourly loads for the current day and
- eight hourly recent loads before the prediction hour.

An unsupervised learning algorithm is used to classify these thirty eight inputs into one of five subsets consisting of different day-types stored in a data set. These five subsets include:

1. working days (excluding Monday);
2. Saturdays;
3. Sundays;
4. Mondays;
5. Holidays.

Then supervised learning is used to train the ANN associated with the above five classes, and to forecast hourly power load. The output of the supervised learning network consists of the next twenty four hourly loads to be predicted. All ANNs except the one used to predict the power load on holidays are retrained weekly. For subset 1), forecast is made sixteen to forty hours in advance. If the hourly power load is to be predicted for week *k*, the training set used consists of the data associated with weeks (k-1)... (k-4) and the weeks (k-47), (k-48)... (k-55), where (k-47), (k-48) ...(k-55) are the data associated with the weeks at the same time in the previous year. For subsets 2), 3) and 4), data used for training is associated with weeks (k-1), (k-2)...(k-6) and weeks (k-46), (k-47)...(k-56). For holidays, the same day from the previous year forms the training set. This method is characterized by its self-revision and adaptation. The network is retrained weekly using a slightly different set of data to modify the weights of the network. These subsets used for the weekly training session forms a window. This window moves forward through removing old measurements while adding newly measured data. It allows the incorporation of possibly most recent changes in climate factors, regulation constraints and localized phenomena. This window evolves with time, hence is referred to as a moving window.

This algorithm is used to forecast the load for the Electric Power Utility of Serbia. The data associated with year 1990 are provided to predict the hourly load associated with 1991. The results of this case study show that both absolute percentage errors and absolute percentage errors are below five percent.

In (Khotanzad et al. 1995), a combination of three separate models is used for short-term load forecasting. They are weekly, daily and hourly models.

Both the daily and the weekly models consist of seven ANNs. Each ANN is used to forecast the hourly load of the i-th day of the week, but the input selection strategies for these two models are different. For the weekly model, the selected inputs are hourly load, actual temperature and the temperature forecast for the i-th day of the previous week. For daily models, the actual hourly load and the hourly temperature measurements from the previous day as well as the forecasted hourly temperature of the i-th day are selected as inputs. Weekly ANN models are updated weekly once the new data is available at the end of the previous week. Daily models are updated on daily basis when the new data are available at the end of the day. The outputs associated with both models are the 24 hourly load forecast of the i-th day.

The hourly module consists of twenty-four ANNs, one for each hour of the day. The input of the i-th ANN consists of

- the actual load, temperature and humidity at the i-th hour of the previous day;
- the actual load, temperature and relative humidity of the i-th hour from two days ago;
- day of the year;

These ANNs are updated daily. The network is retrained at the end of each day when the new measurements become available. The new data is combined with the previous data to achieve higher accuracy (accumulative training). The retrained network is used to forecast the hourly load for the next day.

The outputs from all three ANNs are combined to provide the final load forecast for each hour. If we denote the forecast for the load at the k-th hour from the weekly module by LW(k), the forecast from the daily module by LD(k), and the forecast from the hourly module by LH(k), then the final load forecast has the form

$$L(k) = a1(k)*LW(k) + a2(k)*LD(k) + a3(k)*LH(k),$$

where the coefficients a1(k), a2(k) and a3(k) are also determined in a dynamic fashion through a Recursive Least Square procedure (RLS). Similar to on-line update of ANN weights, the coefficient a(k) is updated by the RLS algorithm on a daily basis.

$$J = \Sigma\beta^{N-j}(L(j) - L(j))^2$$

Where, j=1..N, N is the total number of the forecasts made from the beginning till present, β is a weighting factor to forget the old data, hereβ=0.99. The results of one-day ahead forecast show that mean absolute percentage errors for twenty utilities across the US such as Texas Utilities Electric Co., Alabama Electric Co., Florida Power Corp etc. are pretty low, they are all below 5%.

Similar to the above example, three ANN adaptive mechanisms are utilized in (Mohammed et al. 1995) to accurately forecast the short-term electric load for power systems. These three schemes are Daily Adaptation (DA), Weekly Adaptation (WA) and Monthly Adaptation (MA).

In a DA mechanism, a two-stage training algorithm is used to provide the load forecast. The ANN inputs used for the training at the first stage consist of

- the temperature forecast of the hour,

- the load of the hour from the previous day,
- the maximum and minimum temperatures of the previous and present day

A set of ANN weights are produced through the training in the first stage to capture the general, day by day trend of the electric load. These weights are saved as the starting weights for the initial ANN training associated with the next day. During the second-stage training, the ANN is refined and enhanced to capture special features of the day for which the forecast is made. The second stage training uses a subset of data consisting of those that share similar temperature conditions with the day being forecasted as well as data from the previous five days. Once the forecast is made, the weights obtained from the second stage are discarded and only the weights from the training in the first stage are kept. Then the above process is repeated to forecast the load for next days.

The adaptive mechanisms are applied to the Florida Power and Light Company for load forecasting. The result shows that the mean absolute percentage error for all hours forecast in 1990 are within 6%.

Sforna and Proverbio designed a recurrent neural network on-line system for short-term load forecasting in (Sforna and Proverbio 1995). The system consists of two windows that are for different uses. The first window is a short-term training window. This window is used for training and load prediction. It is controlled by operators. The operator can stop training at anytime when they think the forecast is satisfactory. Otherwise a relearning activity is available. The training set for this window is the data associated with five previous weeks, this is the default option. When rapid changes occur, especially near summer or seasonal holidays, only the data associated with the previous week is used for training. Eleven inputs are selected. They are combined with the loads of the same day in previous weeks and the loads of the day before the current day. These eleven inputs are listed as follows:

- load at d-7 for Monday
- load at d-7, d-1 for Tuesday
- load at d-7, d-1 for Wednesday
- load at d-7, d-1 for Thursday
- d-7, d-1 for Friday
- d-7 for Saturday
- d-7 for Sunday

The network has two hidden layers. The first hidden layer has nine neurons. The second hidden layer has four neurons. One output is hourly load of the next day i.e. day (d+1).

The second window of the system is an on-line correction window. Two types of on-line correctors are designed to minimize the prediction error. The first one is called on-line error corrector. This corrector predicts prediction errors two hours in advance through learning predicted errors associated with the previous three or four hours by a dynamic network. The second corrector is an on-line error estimator. The estimator is used to compute the forecast errors when the current actual load is available.

This model is used to forecast the electric load of the Electric Power Utility in Serbia. The average errors obtained from the forecast for twenty four hours are all within a range of five percent. Operators plays very important roles in determining the accuracy of the forecast but their effect is not obvious. The forecast with the operator's help performed better than the one without the operator. With the operator, the mean prediction error is 1.18%, while it is 1.93% without operator.

Instead of forecasting the short-term load directly, an ANN is used to forecast the relative increment in load in (Charytoniuk and Chen 2000). The relative load increment is defined by d(t+1) = (L(t+1)-L(t))/L(t), where L(t) is the energy load at time t. The rationale for this approach is that the relative load increment is much smoother than the actual hourly load. Charytoniuk and Chen proposed to use an adaptive scheme that involves a moving window. The network is daily retrained. For a given date, the 3 recent weeks of the current year data and the 2 weeks around the given day of the past year data are used for training. This set of data forms a window for training and retraining. The window is moving by discarding the old measurements while adding newly available measurements. The data within the window is sampled every 3-5 minutes. The proposed method reduced the error by 47% and training time by 66% compared to the traditional forecaster.

2.4 Other Methods

In addition to the three types of prediction models described above, other models such as the fuzzy set model (Tobi and Hanafusa 1991; Mori and Kobayashi 1996; Youn et al. 2000; Liu et al. 1996) and Bayesian Regression model have been shown to be effective when used appropriately for certain types of prediction.

The fuzzy approach is an alternative technique used in system control, information processing and management (Shalkoff 1997). It provides a robust mathematical framework that is able to solve various problems such as nonlinear problems that include non-numerical domain knowledge. An obvious operation feature of fuzzy system is its ability to capture uncertainty. There are no absolute true or false values in the fuzzy set member function. Each variable can take a degree of truth. In general, fuzzy member function reflects the approximate relationships between observations and response. Detailed procedures of fuzzy system design can be found in (Shalkoff 1997).

Energy prediction can be viewed as a type of inference process. One technique for dealing with systematic inference is fuzzy logic. A fuzzy system is an inference model that is built upon a set of inference rules. Each rule consists of a premise (input), a consequence (output) and a set of membership functions that define the relationship between some premises and consequences. A typical rule can be described as follows (Mori and Kobayashi 1996): Rule k: if x_1 is A_{1j}^k, x_2 is A_{2j}^k, …, and x_n is A_{nj}^k, then y is w_k, where A_{ij}^k is the j-th membership function associated with the input variable i, where i=1…n.

In (Mori and Kobayashi 1996) each membership function is constructed as a localized piecewise linear function. Each rule is characterized by its "true value", which is defined as the product of the membership function: $\mu_k = A_{1j}^k \cdot A_{2j}^k \cdot \ldots \cdot A_{nj}^k$. It is easy to verify that the product of all "true values" is 1. An output variable can be expressed as $y = \sum_k \mu_k w_k$, where w_k is a weight. The optimal weights can be obtained through a supervised learning process. One way to obtain an optimal set of weights is to minimize the cost function $E = \frac{1}{N}\sum_{l=1}^{N}(y_l - d_l)^2$, where d_l is the target of the l-th data set and y_l is the output associated with the l-th data set, using a gradient based algorithm such as the steepest descent algorithm.

A key aspect of fuzzy system modeling is the determination of the number and location of the member functions. In (Mori and Kobayashi 1996), a special coding mechanism was introduced to allow each configuration of the member function be represented by a bit sequence. The optimal sequence can be obtained through a simulated annealing process. Simulated annealing was proposed in (Mori and Kobayashi 1996) as a promising way to set the number and location of the member functions. However, no computational details were provided in that paper.

In addition to the cost function, the quality of a fuzzy system model can be evaluated by an index function such as (Mori and Kobayashi 1996)

$$f = \frac{E}{E_{\max}} + \beta \cdot m, \qquad (9)$$

where E_{max} is defined to be the model error with membership functions at both edges, β is some parameter, and m is the number of membership functions. This function can be used as the energy function for the simulated annealing process used to find an optimal configuration for the fuzzy system.

A fuzzy system has the same ability as ANN in approximating any functions. According to (Mori and Kobayashi 1996), it has the advantage of capturing cause and effect in the inference process.

In (Mori and Kobayashi 1996), a fuzzy system is used to forecast short-term (one hour ahead) power system load. The input variables are chosen to be the power load at time t, the difference between the power load at time t and the average power loads, and the difference between the averages calculated at time t and $t+1$. The results show that accuracy of the prediction is comparable with that obtained from a multi-layer perceptron (MLP) neural network prediction model on the same training data set. However, a better accuracy is achieved by the fuzzy inference on a testing data set than that obtained by an MLP neural network.

In (Liu et al., 1996), three methods were used for short-term load forecasting: fuzzy logic, neural networks and the autoregressive (AR) model. A simulation is carried out for the next 24 -hour load trend forecasting in winter with the metered history data. Then, the forecasted data is compared to the real load. The RMS value is used as the criterion to evaluate prediction performance. Fuzzy logic and neural networks are better than AR since the RMS values of these two approaches are 1.0% while the RMS value of the AR model is 7%.

The details of a Bayesian regression model that was declared as the winner of the 1994 Energy Prediction Shootout competition are presented in (Mackey 1994). The author views neural network learning as "an inference of the most probable parameters for a model, given the training data." He points out that the main objective of neural network training is to minimize an error function that describes the discrepancy between the output of the nonlinear map (implicitly defined by the neural network) and the measured data. Minimization often includes regularization to avoid noise amplification. He argues that within a probabilistic framework, one can view the error function as the logarithm likelihood for a noise model, and can view a regularizer as a prior probability distribution over the parameters. It then follows that the minimization of a regularized error function is equivalent to the inference of the parameters of the prediction model, given the measured data.

By making use of Bayes' theorem, which basically states that the posterior probability distribution of model parameters is proportional to the product of the likelihood and the prior probability distribution, the author proposed to determine the regularization parameters by maximizing what he called the "evidence" term. He also introduced the concept of Automatic Relevance Determination that "puts a prior probability distribution over the regression parameters that embodies the concept of relevance." The regularization constants for redundant inputs can automatically be detected because they tend to be large.

Some implementation details such as the architecture of the neural networks, pre- and post-processing of the data are discussed. The author provided some explanation on why some of the local trends in the testing data are not correctly predicted. Both coefficient of variation and mean bias error showed this particular prediction model is superior to other models used in the competition.

2.5 The Great Energy Predictor Shootouts

The accuracy not only depends on the model itself but also depends on the quality of the data used to perform the prediction. It is somewhat difficult to compare all the models proposed in the surveyed articles directly because many of them used different data sets to predict different quantities. Fortunately, this issue has been addressed, to some extent, by two competitions called the Great Energy Predictor Shootout I and II held in 1994 and 1996 respectively. Many research groups participated in these two competitions in which the same data set was provided to test various prediction models. The conclusions of these competitions have been described in (Kreider and Haberl 1994) and (Haberl and Thamilseran 1996).

In the first competition, two data sets generated from the same building, a university engineering center were made available to the contestants. One data set consisted of hourly measurements of chilled water, hot water and whole-building electricity use for a four-month period. Weather data and time stamps were included. The other data set consisted of solar radiation measurements. Some of the dependent variables were withheld from each of the data sets. They are used to test the accuracy of the submitted prediction model. The quantities predicted include the whole building electricity, chilled and hot water use. The performance of each prediction model is determined by the CV and MBE criteria. A Bayesian non-linear model (Mackey 1994) was identified as the most accurate prediction model among the contestants. It was found that neural networks-based models generally performed quite well compared to traditional statistical approaches. It was also found that all methods performed better when predicting the chilled water use than when predicting the hot-water use (Kreider and Haberl 1994).

In the second competition, energy data generated from two different types of buildings, one an engineering center, the other a business building, were provided to the contestants. Both buildings had been retrofitted to include more efficient HVAC systems. Hourly energy measurements from both the pre-retrofit and post-retrofit period were included in the data set. Some data were removed from the pre-retrofit period for testing the accuracy of the contestants' prediction model. The winner of the competition used a combination of ten neural networks with two hidden layers of 25 units each (Haberl and Thamilseran 1996). To many people's surprise, one of the contestants showed that a carefully assembled statistical model could perform as well as the neural network approach.

2.6 Generalization and Comparison

Choosing a building energy prediction model is not an easy task. The prediction methods described in the above literature review all have distinctive features. Some of them perform better than others, but none of them is perfect. The regression and time series models are based on classical mathematical theory. Thus, the behavior of these models is well understood. The estimation of model parameters is usually straightforward. However, these models tend to work well only for energy systems that are well behaved. They are usually not amenable to dramatic changes in energy use due to unexpected events. The artificial neural network model generally works better for buildings that exhibit highly nonlinear energy demand patterns. However, the success of using ANN depends on a number of design issues such as the choice of input and output, the number of hidden layers, the number of neurons used in each layer and the training algorithms used.

To provide guidance on how to choose an appropriate building energy prediction model, the advantages and disadvantages of all models discussed are summarized in the following table.

Table 1. The advantage and disadvantage of various energy prediction models

	Advantages	**Disadvantages**
Regression method	A simple function consisting of related regressors. Relatively fewer parameters are required. It reduces the time and effort needed to determine the values of the parameters	i) It does not accurately reflect the hourly energy variation with time. For different buildings with different environment and weather conditions, much effort and time must be spent on selecting time scales and regressors to find a best fit model. ii) Auto-correlation or multicollinearity problems must be considered when evaluating the performance of prediction because they tend to lead to model uncertainty.
Time series model	i) They can capture the relationship between the hourly energy use and time variation given a set of time series data. ii) Fourier series models can even provide a better accuracy of prediction with less computation effort by multiplying the periodic parameters sine and cosine	i) The success of AR models only depends on the assumption that the present value is a linear combination of the previous ones. ARMA and AR models both work under the stationary condition. In most cases this is invalid. The prediction errors of models increase as the number of variables is increased. ii) ARIMA model and ARMAX models can handle the changes in the unstationary process, but many types of parameter estimation are still required. At the same time these models cannot ignore the auto-correlation between the variables because it strongly impacts the accuracy of the prediction. iii) Fourier series models are much easier and provide better performance compared to the above time series models and other techniques. But it is based on the assumption that "hourly energy use in almost all commercial buildings is periodic" (Dhar et al. 1998). If dramatic changes happen this model may not be the proper choice. To resolve dramatic changes, high-frequency Fourier components must be included in the model. This would dramatically increase the computational cost.

Artificial Neural Networks	i) ANN models are capable of approximating a multivariable nonlinear function without knowing previous relationship between variables; ii) ANN models are able to reduce the time of parameter estimation through learning from examples and updating their learned knowledge automatically iii) It is possible to ignore excess data that is unimportant, and to concentrate on the more salient inputs (Curtiss et al. 1994);	Typically operate like a black box. It is not known what happens between the output and the input, and no meanings are applied to the weights. i) Hard to distinguish structure from noise in the data, and ANN tends to memorize noise. ii) It might not be able to adapt to dramatic changes. For example, the power load - temperature relationship may exhibit unstable behavior. That means that the results from a prior training session may not be able to handle the unexpected changes in this relationship. Some remedies can be found in (Kawashima et al. 1994; Yoo and Pimmel 2000), but a significant amount of effort is still needed.

In (Kawashima et al. 1995) different forecasting methods are used and compared. It is found that Artificial Neural Networks provide the best performance. The reported CV and mean bias errors (MBE) values are 10% and 0.5% respectively. By contrast, the reported CV and MBE values associated with the ARIMA the model are 24% and 1% respectively. In (Dhar et al. 1998) Fourier series models were compared to ANN and the winners in the Great Energy Prediction Shootout. The reported CV values of Fourier series models are the lowest in four out of eight cases when compared to the performance of ANN. The performance of Fourier series is also comparable to the winners in the shootout competition as described before. Based on the above information and description, Fourier series models and Artificial Neural Networks based models are the two most recommended methods found in the literature survey. Hybrid prediction models that combine more than one technique are also recommended. Such models include Fuzzy-Neural Networks and Baysian Regression models. These models usually generate accurate results but the design of these models is more complicated than any of the methods described in section 2.1 to 2.5.

If the energy use pattern is highly non-linear with respect to the independent variables such as weather and operational variables, a single physical model expressed by an explicit mathematical formula (e.g., a regression model or a time series model) may not be enough to accurately describe the complicated time-dependent relationship between the inputs and outputs. Moreover, physical models tend to cause problems such as multi-collinearity that leads to model uncertainty. By contrast, an Artificial Neural Network (ANN) model is more flexible in terms of modeling a nonlinear mapping. It does not rely on a fixed mathematical description of the physical phenomenon. Once the input and output data are provided, ANN models able to describe a highly nonlinear phenomenon.

A number of building energy prediction models have been examined in this chapter. A brief description of each of the prediction methods examined has been provided here to illustrate how these prediction models are used in practice and the type of prediction performance achieved by each model. The advantages and disadvantages of these models are summarized in Table 1. Much emphasis has been placed on the Artificial Neural Network model because it has the unique advantage that no clear relationship between the input variable and output needs to be defined before the model is used in the prediction process. Once the input and output are selected and fed into an ANN, the relationship between the input and output is identified through a self-learning process. The accuracy of the prediction can be improved by adjusting the architecture of the ANN.

Because of this unique feature, the time and effort that are normally required to establish a proper mathematical model in a conventional prediction methodology can be saved. The ANN based models appeared to perform well in the two Great Energy Shootout competitions. The ANN model has also been successfully used in HVAC plants and power system as an energy predictor and controller. Besides the advantages summarized in Table 1, ANN is also easy to use. It can be installed on almost any desktop computer and the memory requirement for running an ANN based building energy predictor is small.

Although ANN models have been used in a number of building energy prediction applications, the design of an effective ANN prediction model is still not a trivial task. Designing an effective ANN model requires experience in choosing the appropriate inputs, the number of neurons per hidden layer and the number of hidden layers per network. Further study of ways to improve the accuracy and efficiency of ANN based prediction methods is still needed. In addition, the following problems should be addressed in the future for building energy prediction methods:

- For different buildings with different environment and weather conditions, different independent variables and time scales must be selected to ensure that prediction errors can be minimized. Much experience and knowledge is building-dependent (Anstett and Kreider 1993). This is a general problem that cannot be avoided.

- To accurately and successfully model complex buildings, a certain level of experience is required with selecting the proper inputs (Kreider et al. 1995). This is especially true in neural network models. To present a good performance of prediction, all the methods must define day-type groups to deal with energy use on different types of days such as weekday, weekend and holiday.

Almost all ANN models for building energy predictions presented in the surveyed literature focus on static predictions. In a static prediction, the prediction model is set up in advance using historical data. It is highly possible that this model maybe invalid when new data that records the energy and environmental changes association with a different and more recent time period becomes available. In this case, a dynamic prediction that can adapt itself to such changes in the energy demand pattern is desirable. This is especially true for short-term energy prediction. In this thesis, dynamic ANN models are proposed for on-line building energy prediction.

3. ON-LINE BUILDING ENERGY PREDICTION USING ARTIFICIAL NEURAL NETWORKS

The ANN model is identified in the literature survey as the most powerful and flexible prediction model. There are many variations of ANN models, each well suited for a particular application. Although the basic concept of ANN is easy to understand, there are several issues one should address carefully in designing an effective ANN prediction model. In this chapter, several key issues related to applying an ANN model to building energy prediction are examined. The discussion focuses on how to choose appropriate input, the internal structure of the network, training methods, and adaptive ANN models. The methodology for solving nonlinear least squares problems between ANN training and the classical methods are presented. The technique Principal Component Analysis (PCA), which is used to reduce the input dimension and redundancies in the data is introduced. Two dynamic ANN models are proposed.

3.1 Input and Output

The output of the ANN used for energy prediction is often easy to choose. It is usually the total electric, gas or chiller energy consumed by a building, a quantity that is of interest to the building manager. The input vector to an ANN varies from one application to another. In building energy prediction applications, typical input elements include outdoor dry-bulb temperature, wet bulb temperature, horizontal solar flux, and hour of the day, etc. For long-term prediction, one may also include day of the month and month of the year as input variables. For some applications, one may not have access to all environmental variables that contribute to variations of energy usage. In this case, the accuracy of the ANN prediction will be limited by the incompleteness of the measured data. In other applications, the measurements may contain environmental data that do not actually contribute to variations of energy usage. These measurements should be pruned from the list of input elements to the ANN predictor for two reasons:

- If some measurements are irrelevant to the energy usage to be predicted, they carry no useful information and contribute only noise to the ANN output (Dodier 1995). Removing these type of measurements from the list of inputs can improve the accuracy of the prediction.
- Removing irrelevant measurements can also reduce the volume of the training data significantly, thereby improving the efficiency of an ANN predictor.

There are several ways to detect irrelevant input elements to ANNs. Dodier proposed using the Wald's test (Wald 1943) as a criterion to determine the relevant input variables. The Wald's test is based on the theory of hypothesis testing commonly used in statistical experiment design. In Dodier's scheme, all measurements are selected as inputs to ANNs for initial training. The decision on which input variables to retain is based on the probability distribution of the weights associated with each neuron in the input layer. The Wald's test assumes that the distribution of the weights follow a Gaussian distribution. This assumption only holds when the volume of data is large enough. It is not clear how large a volume of the data is required for the Wald's test to be valid. Thus, this approach is not pursued in this thesis.

Mackey used a technique called 'automatic relevance determination' (ARD) to choose input variables (Mackey 1992). The ARD method is very close to the Wald's test. The major difference is that an ARD test is a "soft" test. At the end of an ARD test, irrelevant measurements are not completely pruned from the list of inputs. They are simply weighted by significantly smaller weights.

Ohlsson (Ohlsson et al. 1994) used a statistical test for nonlinear correlation called 'Delta-test' to choose relevant input variables. A similar approach was taken by Charytoniuk and Chen (Charytoniuk and Chen 2000) to calculate the significance of the correlation between the input and output using non-parametric correlation. However, no details were provided in either one of these approaches.

Feuston and colleagues (Feuston and Thurtell et al. 1994) used a well-known multivariate statistical analysis technique called Principal Component Analysis (PCA) to assemble, synthesize and select relevant input variables among a large number of measurements. As a consequence, the neural network input vector does not consists of the original environmental and calendar variables directly, but linear combinations of these variables. The PCA technique is well established in other fields. It is conceptually simple, and it is also supported in the MATLAB ANN Toolbox. Thus, this technique is used in this thesis. The details will be presented in Section 3.4.

3.2 Internal Structure

Although ANN models for building energy prediction can come in different ways, the basic structures of these models are usually the same. Between the input and output layers are a number of hidden layers containing a set of neurons. Figure 1 shows a typical diagram for a multi-layer ANN. The input to the network is denoted by $p_i, i = 1,2,3$. The output of the network is denoted by y. There is only one hidden layer in the figure. The diagram contains a single output, a situation that is commonly seen in building energy prediction such as the total electric energy demand of a building. However, the structure presented does not prevent one from including additional output variables to the network.

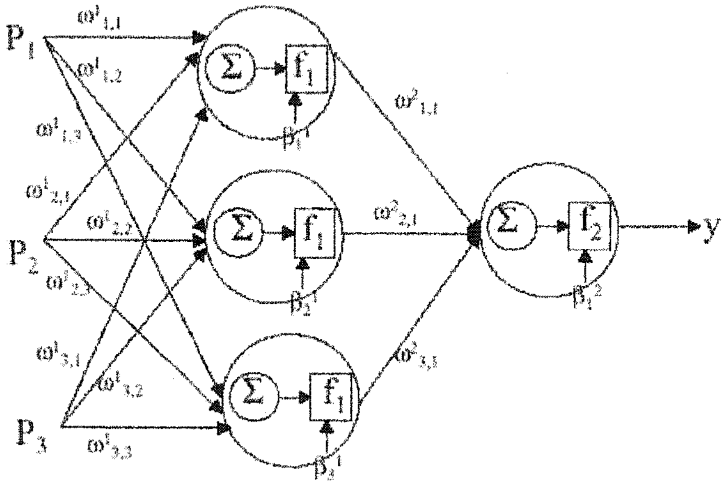


Figure 1.The basic architecture of a simple ANN(Matlab User's Guide, Version 4)

Each input to the network is weighted before it is fed into a neuron. The weight associated with p_i and the j-th neuron in the first hidden layer is denoted by $\omega^1_{i,j}$. The weighted input variables entering the j-th neuron are summed (denoted by the $\sum$ sign in the diagram) and shifted by a bias denoted by β^1_j . The result is then passed into an activation function f_1 which allows some information to pass through the network while inhibiting the propagation of other information. The standard activation functions include: the *Identity* function, the *Binary Step* function, the *Binary Sigmoid* function, and the *Bipolar Sigmoid* function (Fausett 1994). Non-linear activation functions are preferred in a hidden layer because of their flexibility in handling sophisticated input-output mappings. The activation function often contains a threshold that determines the level to which the input signal will be mapped. The simplest activation function has the form

$$f(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases} \qquad (10)$$

Here the threshold is zero. More sophisticated activation functions include the sigmoid type of function such as

$$f(s) = \tanh(\alpha s), \qquad (11)$$

where α is a slope parameter that plays the role of a threshold. Figure 2 shows the shape of this function.

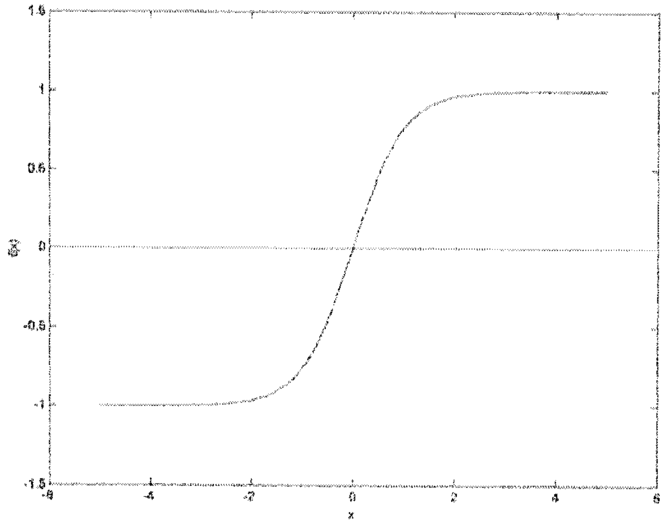


Figure 2. The bipolar sigmoid activation function

The ANN shown in Figure 1 is fully connected. The significance of the links between neurons in two adjacent layers is determined by the weights attached to the links. The combination of weights $\omega_{i,j}^{k}$ and biases $\beta_{i,j}^{k}$, transformed by activation functions f_k, make the output of the ANN a highly nonlinear function of the input. The nonlinear mapping can be analyzed as follows. If we let

$$W_1 = \begin{pmatrix} \omega_{1,1}^1 & \omega_{1,2}^1 & \omega_{1,3}^1 \\ \omega_{2,1}^1 & \omega_{2,2}^1 & \omega_{2,3}^1 \\ \omega_{3,1}^1 & \omega_{3,2}^1 & \omega_{3,3}^1 \end{pmatrix}, \quad p = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \quad \text{and} \quad b_1 = \begin{pmatrix} \beta_1^1 \\ \beta_2^1 \\ \beta_3^1 \end{pmatrix}, \qquad (12)$$

then the output from the first hidden layer can be expressed by

$$q = f_1(W_1 p + b_1). \qquad (13)$$

Continuing in this fashion by taking q as the input to the next layer, then the output can be obtained as follows:

$$y = f_2(W_2 q + b_2) = f_2\left[W_2 f_1(W_1 p + b_1) + b_2\right], \quad (14)$$

where

$$W_2 = \begin{pmatrix} \omega_{1,1}^2 \\ \omega_{2,1}^2 \\ \omega_{3,1}^2 \end{pmatrix}, \quad q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \quad \text{and} \quad b_2 = \begin{pmatrix} \beta_1^2 \\ \beta_2^2 \\ \beta_3^2 \end{pmatrix}. \quad (15)$$

3.3 Training Methods

A set of pre-assigned weights and biases would most likely lead to a network output that is quite different from the desired response. By allowing the weights and biases to vary based on the difference between the network output and the desired response, one can enable the neural network to learn from its own mistake, thereby improving the prediction accuracy.

To facilitate the learning capability of an ANN, one must provide a set of desired response or measured data to allow the ANN to correct itself by repeatedly modifying its weights and biases. The process of feeding an ANN with measured data so that it can recognize the pattern of the data by adjusting the weights and biases associated with various link is called a *training* process. Assuming the same data set is used to train an ANN, the quality of the training is determined by the way weights are modified in response to the prediction error produced by the ANN.

Using the notations established in Section 3.2, one can formulate the ANN training process as a way to solve a nonlinear least squares (NLS) problem. If it is assumed that *m* sets of input-output pairs $(p^{(\ell)}, y^{(\ell)})$, $\ell = 1,2,\ldots,m.$ are available for training, then the objective of the training process can be simply described by

$$\min_{W_1,W_2,b_1,b_2} F(W_1,W_2,b_1,b_2) = \sum_{\ell=1}^{m}\left[y^{(\ell)} - \hat{f}(W_1,W_2,b_1,b_2;p^{(\ell)})\right]^2, \quad (16)$$

where $y = \hat{f}(W_1,W_2,b_1,b_2;p) = f_2[W_2 f_1(W_1 p + b_1) + b_2]$.

The standard approach to minimize a function *F(x)* follows these steps(Fletcher 1987):

Choose a starting point x_0;

Choose a search direction s such that $F(x_0 + \alpha s) < F(x_0)$ for some step length α;

Choose an appropriate step length α and set $x_0 \leftarrow x_0 + \alpha s$;

If $F(x_0)$ is still too large go back to step 2;

Steps 2-4 above form one iteration cycle of the minimization procedure. In ANN terminology, it is referred to as an *epoch.*

There are a number of ways to choose a search direction. A standard choice is the negative of the gradient of *F(x)*, which is the *steepest descent* direction at point *x*. To compute the gradient of *F(x)*, one must calculate the derivative of *F* with respect to *x*, i.e.

$$\nabla F(x) = \begin{pmatrix} \dfrac{\partial F}{\partial x_1} \\ \vdots \\ \dfrac{\partial F}{\partial x_m} \end{pmatrix}. \quad (17)$$

It follows from the NLS formulation established above that one must compute the partial derivatives of $F(W_1, W_2, b_1, b_2)$ with respect to the weights and biases associated with all active neurons within an ANN in order to obtain a gradient descent search direction. The special structure of the neural network illustrated in Section 3.2 allows one to compute the partial derivative of F with respect to some of the weights and biases by making use of the chain rule. This is the essence of the *back-propagation* algorithm.

An alternative search direction can be calculated by approximating $F(x)$ near x_0 by a quadratic model (Fletcher 1987)

$$m(x) = F(x_0) + \nabla F(x_0)^T (x - x_0) + (x - x_0)^T \nabla^2 F(x_0)(x - x_0), \qquad (18)$$

where $\nabla^2 F$ is the Hessian of $F(x)$. If $m(x)$ and $F(x)$ are in good agreement near x_0, one may choose the search direction to be the minimizer of

$$m(s) = F(x_0) + \nabla F(x_0)^T s + s^T \nabla^2 F(x_0) s. \qquad (19)$$

The minimizer of the above quadratic form satisfies

$$\nabla^2 F(x_0) s = -\nabla F(x_0). \qquad (20)$$

The search direction s defined by the above equation is often called a Newton's direction (Fletcher 1987). The main advantage of using a Newton direction is that when x_0 is near the minimizer of $F(x)$, the convergence of the training process is quadratic. However, to obtain a Newton direction we must compute both the first and second derivatives of F and solve a linear system of equations at each x_0.

Simple Hessian updating formulae have been developed to save the amount of computation involved in Hessian calculation and in solving equation (20). One of the well-known updating formula is the Broyden, Fletcher, Goldfarb and Shannon (BFGS) formula (Fletcher 1987). If we use H to denote the Hessian of $F(x)$ at x_0, then the BFGS formula suggests that

$$\hat{H} = H - \frac{Hss^T H}{s^T Hs} + \frac{gg^T}{g^T s}, \qquad (21)$$

where $g = \nabla F(x_0)$, can be used to approximate the Hessian of $F(x)$ at $x+s$. Formulae like (21) define a class of optimization algorithms known as the Quasi-Newton algorithm. The solution to

$$\hat{H}s = -\nabla F(x_0) \qquad (22)$$

defines a *Quasi-Newton* search direction (Fletcher 1987).

Yet another way to choose a search direction is to take advantage of the NLS structure of the objective function in (16) and express the Hessian of $F(x)$ by

$$H = J^T J + \sum_{\ell=1}^{m} r_\ell \nabla^2 r_\ell \quad (23),$$

where $r_\ell = y^{(\ell)} - F(W_1, W_2, b_1, b_2; p^{(\ell)})$, and J is the Jacobian of the residual function

$$r(W_1, W_2, b_1, b_2) = \begin{pmatrix} r_{(1)} \\ \vdots \\ r_{(m)} \end{pmatrix} \qquad (24)$$

with respect to the weights $\omega_{i,j}^k$ and biases β_i^k. If either r_ℓ or $\nabla^2 r_\ell$ is small in some sense, then the Hessian can be approximated by $J^T J$. This approach yields the Gauss-Newton search direction (Fletcher 1987) defined by

$$J^T Js = -J^T r \,. \ (25)$$

Once a search direction is chosen, one must determine a step length α to complete one iteration of the minimization procedure. The step length is often referred to as the *learning rate* in ANN literature. If the search direction is chosen to be the gradient descent direction, then one can achieve $F(x_0 + \alpha s) < F(x_0)$ by choosing α to be sufficiently small. However, a tiny α can result in many training iterations (epochs), thereby slowing down the convergence of the training process.

When a Newton, Quasi-Newton or Gauss-Newton search direction is used, an additional constraint of the form $\|s\| \leq \Delta$ is often imposed to ensure that the x stays within the region in which $F(x)$ and the quadratic model $m(x)$ are in good agreement. Otherwise, a Newton-like search direction may not even be a descent direction. The constraint $\|s\| \leq \Delta$ is often called a trust region (a region where one can trust the quadratic model $m(x)$ to be an accurate approximation to $F(x)$). A Gauss-Newton search direction combined a trust region strategy for choosing an appropriate step length yields a method known as the *Levenberg-Marquardt* method (Fletcher 1987).

The MATLAB ANN ToolBox provides a variety of training methods. For example, one can use the command

```
net=newff([-1 2;-3 3;0 6],[3 7 1],{'tansig', 'tansig', 'purelin'}, 'trainlm');
```

to set up a feed-forward neural network (i.e., 'newff') with three input elements, one output and 1 hidden layer with three and seven neurons in the first and second layers respectively. The first argument of this function call specifies the numerical ranges of the three input elements (i.e., [-1 2;-3 3;0 6]). Sigmoid functions 'tansig' are used at the first and second hidden layers. A linear function 'purelin' is used at the output layer to allow the network to produce values outside of the range –1 to 1. The last argument of this function call specifies that the Levenberg-Marquardt method will be used to train the ANN (i.e., trainlm). The last argument can be replaced with any of the strings listed in Table 2 to invoke other training methods.

Table 2. Training methods available in MATLAB ANN toolbox

Argument	**Description**
traingdm	Gradient Descent Method
trainbfg	BFGS Quasi-Newton
trainscg	Scaled Conjugate Gradient
traincgf	Fletcher-Powell Conjugate Gradient
trainlm	Levenberg-Marquardt

3.4 Dimension Reduction

In practice, it is difficult to decide which independent variable from a long list of variables should be kept as an ANN input and which should be removed. Sometimes, two independent variables are clearly correlated. In this case, only one of them is needed as an input to the desired ANN. However, in many cases, the correlation among multiple variables is not so obvious. Thus, a more systematic way of removing data redundancy is required. Furthermore, it is often desirable to reduce the size (number) of the input to the ANN used for real-time energy prediction so that ANN training can be completed within a reasonable amount of time.

It is well known in multivariate statistical analysis (Johnson and Wichern 2002) that the task of reducing the dimensionality of the data and removing redundancy can be accomplished through the use of *Principal Component Analysis* (PCA). The main idea of PCA is to seek clusters of data points that can be used to represent the main features of the data. In PCA, each input variable (vector) that assumes n values in time can be viewed as a point in an n-dimensional space. Assume m vectors are potential candidates for the independent variables of an ANN model, and they are named as $x_1, x_2, \ldots, x_m$.

To identify a cluster that contains a number of vectors that are "close" together, one can seek a vector u such that the sum of the projections of $x_1, x_2, \ldots, x_m$ onto u is maximized. That is, one needs to solve

$$\max_{\|u\|=1} \sum_{i=1}^{m} (x_i^T u)^2. \qquad (26)$$

If one uses $X = (x_1, x_2, \ldots, x_m)$ to denote a matrix consisting of column vectors $x_1, x_2, \ldots, x_m$, the above maximization problem can be written as

$$\max_{\|u\|=1} u^T XX^T u \qquad (27)$$

Therefore, finding a unit-norm vector u that identifies a cluster of $x_1, x_2, \ldots, x_m$ is equivalent to computing the eigenvector of the variance-covariance matrix XX^T corresponding to the largest eigenvalue of XX^T. The eigenvector u is called a (dominant) principal component of the set of variables $x_1, x_2, \ldots, x_m$, the eigenvalue λ that satisfies

$$XX^T u = \lambda u \qquad (28)$$

can be used to measure the variances among the variables $x_1, x_2, \ldots, x_m$ with respect to the principal component u. The principal component u is capable of capturing the common features of variables $x_1, x_2, \ldots, x_m$. However, it is not sufficient to use u as a single input to the ANN to be designed. To identify additional feature of the data, one can seek an additional vector v that is orthogonal to u such that $\sum_{i=1}^{m} (x_i^T v)^2$ is maximized. The solution of this problem is associated with the second largest eigenpair of XX^T. Because u and v are orthogonal by construction, they are completely uncorrelated. The eigenvalues associated with u and v indicates the importance of these principal components in terms of representing the features of the original data set $x_1, x_2, \ldots, x_m$.

One can continue the above process to compute more principal components that are mutually orthogonal. When the eigenvalue associated with a particular principal component becomes negligibly small, the process can be stopped. The principal component vectors computed up to this point can be used as input to the ANN.

3.5. Other Issues

Typically one or two hidden layers are sufficient in an ANN designed for building energy prediction. The number of neurons in each layer may vary depending on the quantities to be predicted. The presence of redundant neurons does not pose a significant problem. A good learning algorithm tends to ignore the excessive parameters in the system. The redundancy can also be detected by "examining column dependency in the matrix of the hidden neuron outputs computed from the training data" (Chartoniuk and Chen, 2000).

To prevent an ANN from memorizing the noise in the training data, it may be advantageous to reserve a subset of the data for cross validation. The magnitude of the cross validation error can be used as a stopping criterion for the ANN training. It may also be advantageous to preprocess the data in advance so that data with distinctive features are grouped together. Instead of building a large ANN to accommodate the entire data set, several ANN models can be constructed to tackle each group of data.

3.6 Adaptive ANN Models

Unlike a time series prediction model, the ANN prediction model described above is static in nature. In a static prediction model, one tries to establish the model in advance and estimate the model parameters using either historical or experimental data. Once the model is built, it is rarely changed even though the presence of new data may indicate that the model is no longer valid. Such a model can estimate, for example, the savings in energy cost when a building undergoes a retrofit.

However, such a model is not suitable for short-term energy predictions that are intended to provide valuable information aimed at improving the flexibility and accuracy of an energy management system (EMS). For example, for the optimal control of HVAC systems, an optimum strategy is to seek to minimize the utility cost with efficient cooling allocation between the chiller and storage. The control actions taken by the supervisory controller are chiller operating level, start time of charging, and start time of depletion of storage etc. To determine these parameters, a prediction of the next hour and next day's building cooling load profile, an adaptive update of the predicted profiles, a building electric demand profile and a mechanism to use this updated load profile to determine the storage discharge profile are required by the optimal controller

Although it may be possible to accomplish the above tasks by a static prediction model, the drawback for using a static prediction model for building energy prediction is clear. In order to accurately predict the building energy usage for all seasons, a large-scale global ANN prediction model must be trained with a sufficiently large volume of historical data to capture daily, monthly and yearly variation of energy usage. Historical data that spans multiple years and distinctive environmental and weather patterns is usually difficult to obtain. Training an ANN using such a large volume of data is often a difficult task. Even if one has the data and completes the training process successfully, the ANN prediction model may still miss-predict energy usage when there is an unexpected change in, for example, weather conditions.

Therefore, it is highly desirable to develop a dynamic ANN model that can be constantly modified and updated as new environmental and operational data becomes available. The ANN prediction model should have an inherent self-revision and adaptation capability to adapt to abnormal weather, holiday and other condition changes.

Clearly, the ability to adapt to new input-output mapping patterns is the key in a dynamic prediction model. In building energy prediction, a dynamic prediction model is often called an "on-line" prediction model to reflect the fact that new environmental and operational data becomes available in a continuous fashion. In the following section, the design and implement of an online ANN model for building energy prediction is described.

The need for on-line prediction provides motivations for developing dynamic models. An intuitive way to build a dynamic ANN prediction model is to simply divide the time line into multiple segments, for example, segments of days or weeks, and to periodically build or redefine a prediction sub-model for each time segment. The sub-models can differ in many ways. One can certainly modify the ANN architecture of the sub-model from one time segment to another if it is believed that certain time segment requires a more complicated ANN architecture to capture a highly irregular relationship between the prescribed independent variables and the energy output to be predicted. However, it is often sufficient to just retrain an existing ANN model established for the previous time segment with a new set of data to obtain a new set of weights and biases. When a reasonable number of neurons (each associated with a weight and bias) are present in the hidden layer of the network, an ANN is often capable of modeling moderately complicated nonlinear mappings.

It remains to be decided how one should partition the time line, i.e., how often should the ANN sub-model be modified or retrained? Should it be retrained daily, weekly or hourly? (Mohammed et al. 1995). Usually the choice of the frequency with which the sub-model should be retrained is determined by the pattern of the data or the actual need. For example, if the temperature measurement shows large daily variation, it will be appropriate to retrain the network daily.

Another important issue to be addressed is what portion of the data one should use to retrain an ANN model. Two approaches are proposed in the next two subsections.

3.6.1 Accumulative training

It is known from the literature mentioned in Section 2.3.1 that an ANN can be retrained periodically by a set of augmented data infused with freshly recorded measurements. This type of training strategy is referred to as *accumulative training*. Accumulative training has the obvious advantage of being able to identify both the local (for example, daily) and the global (seasonal) trend of energy variation. Its main disadvantage lies in the fact that the volume of data accumulated continuously may become too large to be manageable. The larger the volume of the training data, the longer it takes to training the ANN. When the volume of the training data becomes too large, it may become difficult in practice to obtain the prediction result in real time. It is also possible that local changes in the accumulated training data set have smaller impact on the prediction results. This is because the local changes are viewed as small on a relative scale.

The concept of accumulative training is illustrated in Figure 3.

```
Algorithm: Accumulative training

Input: Initial training data set (which includes the environment, operational and energy demand measured during some period in the past)
Output: predicted chiller electric demand in future hours $t=1,2,\ldots,t_{max}$
1. Preprocess the training data to
   1) Normalize all variables so that they lie between $-1$ and 1;
   2) Use PCA to reduce the dimension of the input;
2. Define a training interval $\Delta t$, error tolerance, maximum number of epochs allowed etc.;
3.Train the baseline ANN until the MSE between the training output and the target is
   less than a prescribed tolerance or when a maximum number of epochs are used;
4. $t_0=1$;
5. While $(t < t_{max})$
      1) Query for new measurements that can be used as ANN input;
      2) Preprocess the ANN input in the same way that is done in Step 1;
      3) Use current ANN model to predict energy demand at the t-th hour;
      4) Post-process the output;
      5) $t=t+1$;
      6) If $(t=t_0+\Delta t)$
            a) Add the measurements collected between $t_0$and $t_0+\Delta t$ into the training data set;
            b) Retrain the ANN;
            c) Set $t_0=t$;
6. End while
```

Figure 3. Accumulative training algorithm

3.6.2 Sliding window

Alternatively, an ANN can be retrained just by a fixed amount of the most recent data. As new measurements become available, they are added into the training data set while some of the least recent data are dropped from the training set in such a way the size of the data set remains constant. This approach can be viewed as sliding a time window across a time series of measurements to select the training data. A dynamic ANN model based on this training technique is called a sliding window ANN. A dynamic ANN prediction algorithm based on the sliding window approach is outlined in Figure 4. The relative small (constant) size of the training data makes it possible to perform ANN prediction rapidly. The drawback of this approach is that the training data may only contain local information, and the prediction result may not accurately reflect the annual or seasonal change in energy usage pattern.

It is difficult to determine the ideal size of the window ahead of time. Clearly, the window size cannot be too large. Otherwise, it defeats the purpose of limiting the training data volume. However, if the window size is too small, the resulting ANN model maybe unable to completely capture the behavior of the energy demand.

```
Algorithm: Sliding window training

Input: Initial training data set (which includes the environment, operational and energy
demand measured during some period in the past)
Output: predicted chiller electric demand in future hours $t=1,2,\ldots, t_{max}$
1. Preprocess the training data to
  1) Normalize all variables all so that they lie between $-1$ and 1;
  2) Use PCA to reduce the dimension of the input;
2.Define a training window size W, training interval $\Delta t$, error tolerance, maximum
number of epochs allowed etc.;
3. Train the baseline ANN until the MSE between the training output and the target is
    less than a prescribed tolerance or when a maximum number of epochs are used;
4. Set $t_0 = 1$;
5. While ($t < t_{max}$)
        1) Query for new measurements that can be used as ANN input;
        2) Preprocess the ANN input in the same way that is done in Step 1;
        3) Use current ANN model to predict energy demand at the t-th hour;
        4) Post-process the output;
        5) $t=t+1$;
        6) If ($t= t_0+\Delta t$)
                a) Remove T oldest measurements from the training set and add the
        measurements collected between $t_0$and $t_0+\Delta t$ into the training set;
                b) Retrain the ANN;
                c) Set $t_0=t$;
6. End while
```

Figure 4. Sliding window training

4.COMPUTATIONAL EXPERIMENTAL RESULTS

This section contains the computational results obtained from applying the ANN model discussed in Chapter 3 to two data sets. The first data set contains simulated energy data produced by the DOE 2.1E energy analysis software (MICRO-DOE-2.1E User's Guide Acrosoft International Inc. Golden, Co. 1994). The building for which the energy prediction is made is located in the region of Montreal. The second data set contains real measurements collected from the CANMET Energy Technology Center in Varennes, Quebec.

The simulated data associated with the Laval Building is noise free. This means that the building is assumed to operate under normal conditions, and there is no measuring errors or operation mistakes. Because the simulated data is generated from a well-behaved energy model by using the DOE 2.1E simulation software, it provides an ideal scenario under which the variation of the energy demand is more "predictable". Such a data set makes it possible to quickly develop and test an ANN prediction model. Performing experiments on this data set serves as the first step towards testing a realistic ANN design. Therefore, the ANN models developed in this thesis are first tested on the Laval data set. As will be shown in Section 4.1, the ANN models developed in this thesis indeed perform well on this data set.

To demonstrate the effectiveness of these ANN models in a real world environment, experiments are performed in Section 4.2 to show the effectiveness of using the ANN models developed in this thesis to predict both the gas and chiller energy demand associated with the CANMET Lab. The CANMET data set contains a number of anomalies that make it more challenging to produce highly accurate prediction results. These problems will be discussed in Section 4.2. The experiments shown in Section 4.2 demonstrate the typical difficulties encountered in developing an ANN model to predict the energy demand in a real building.

Several experiments are performed for each data set. Because a static ANN model serves as the building block for developing a dynamic ANN, it is constructed and tested in Sections 4.1.3, 4.2.3 and 4.2.4.1-2 before the experiment associated with the dynamic models are shown in Sections 4.1.4, 4.2.4.3-4. This is because dynamic predictions can only be expected to produce accurate results when the results of static predictions are reasonably accurate.

Several experiments are performed in Sections 4.1.3, 4.2.3 and 4.2.4 to compare different parameter settings of the ANN prediction model and different input choices. Lag-free environmental variables are used as ANN inputs first in Sections 4.1.3.1, 4.2.3.1 and 4.2.4.1 for static predictions, because it is important to verify that the ANN models developed in this thesis can at least capture the instantaneous nonlinear mapping between the energy demand and the environmental variables. Because in reality the environmental variables measured at time *t* cannot be used immediately as the input to predict the energy demand at time *t*, the input variables to the ultimate ANN model developed in this thesis must consist of time lagged measurements. The experiments presented in Sections 4.1.2.2, 4.1.2.3, 4.2.4.2 and 4.2.4.3 demonstrate the effect of including time lagged measurements as ANN input and show that the prediction efficiency and accuracy can be improved by making use of the technique of PCA.

With a reliable and accurate static prediction model on hand, it becomes easier to modify the model to make it adaptive to new features of energy demand pattern present in the most recent measurements. As discussed in Section 3.6, this can be achieved through periodically retraining a static model. In Sections 4.1.4.1 and 4.2.4.3, experiments are performed to evaluate the performance of accumulative training. In Sections 4.1.4.2 and 4.2.4.4, experiments are performed to assess the accuracy and efficiency of a dynamic ANN trained by a sliding window.

Since the heating and cooling systems have different working environment and operation characteristics, they are treated separately in the case of the CANMET building. That is, the chiller and gas energy demands are predicted from two different ANN models. Experiments related to gas energy prediction are described in Section 4.2.3. Experiments related to chiller energy prediction for the CANMET building are described in Section 4.2.4.

4.1 Predicting the Chiller Energy Usage for the Laval Building

The Laval office building, located in Montreal, was built in 1972. The building has a total floor area of 10,410m^2 spread over a seven-floor office tower, an underground garage and a ground floor (with a bank, a restaurant and an office spaces). There is a central Variable Air Volume system, which provides cooling in the summer and ventilation all year to the office spaces from 7:30 am to 11:00pm, from Monday to Friday. The supply fan has a capacity of 38,000 L/s and a motor of 93 kW, and the return fan has 35,000 L/s and 56 kW. The cooling set-point temperature is 23-24°C. Direct expansion cooling coils are connected to four condensing units, each equipped with two compressors with a refrigeration capacity of about 90 kW. The supply air temperature is controlled in terms of the outdoor air temperature; it has a minimum value of 14°C when outside temperature is 9 °C or higher and a maximum value of 16 °C when outside temperature is -20 °C or lower. The system is also equipped with a dry-bulb temperature economizer system that closes the dampers to a minimum position when the outdoor temperature is too high.

The value to be predicted here is the electric demand by the chiller using an air cooled condensor. The data set consists of hourly measurements of the dry-bulb temperature, the wet-bulb temperature, the temperature of the water leaving the chiller, the temperature of the water entering the condenser, as well as the hourly electric demand by the chiller. These values are extracted from the DOE-2 output and are used as experimental data. To simplify the discussion, the notations defined in Table 3 will be used to refer to these variables.

Table 3. Variables used in the chiller electric demand prediction for the Laval building.

Variable	Description	Unit
Td(t)	Dry-bulb temperature	°F
Tw(t)	Wet-bulb temperature	°F
Tl(t)	Temperature of chilled water leaving the chiller	°F
Te(t)	Temperature of the air entering the condenser	°F
H(t)	Hour of the day	Hour
E(t)	Chiller electric usage at time t	kW

The electric usage of the chiller is assumed to be a function of the above quantities and the hour of the day. Thus, the goal of our experiment is to use measurements of the above environmental and operation variables collected at hours *t-i*, for *i=1,2,...*, as the input to an ANN to predict the electric energy consumed by the chiller at the *t*-th hour.

The data used in this section is considered noise-free. This means that the building operates under normal conditions with no measuring error or operation mistakes. A subset of the data is reserved for training purpose, while the rest is used to assess the accuracy of the model.

The experiments performed below are coded in MATLAB 6. The MATLAB ANN Toolbox is used to build, configure and train the network specifically designed to predict chiller energy usage.

4.1.1 The ANN Model

The ANN model used in the following experiments consists of one hidden layer in addition to the input and output layers. The input layer contains *n* neurons from which *n* different inputs are fed into the network. The output layer contains one neuron from which the predicted chiller electric usage will be extracted for the next hour. The hidden layer consists of $2n+1$ neurons (Hecht 1989). This three-layer ANN model was found to be sufficient for making a reasonably accurate prediction of the chiller electric demand. Adding more layers and/or neurons can potentially improve the prediction accuracy. However, using more layers and/or neurons per layer adds complexity to the ANN training time, and may degrade the computational efficiency of the ANN model. A number of experiments have been performed using different number of hidden layers. But the performance of the experiments does not show significant difference. In particular, when the number of hidden layer is increased, the accuracy of the prediction result does not necessary improve. So in the following experiments, only one hidden layer is used.

As discussed earlier, the MATLAB ANN Toolbox allows one to choose from several back-propagation algorithms to train the network. All of these algorithms were tested. When the data size is small, the Levenberg-Marquart (LM) algorithm appeared to be the fastest training algorithm (the mean square error of the ANN output approaches to zero at a quadratic rate). However, because the LM method must solve a linear system of equations in order to obtain a search direction, the computation becomes expensive when the number of input elements and the volume of the training data increase. Therefore, when the number of input elements or the volume of the data is large, the standard gradient descent algorithm is used for training the ANN.

Biploar sigmoid functions are used as the activation function in each neuron. In the output layer, a linear transfer function is used in addition on the output to allow the network to produce values outside of the range [-1, 1]. The training process is terminated when the mean square error (MSE) between the ANN output and the target values becomes less than 10^{-5} kW, or when a maximum of 500 epochs is reached, whichever condition is reached first. The initial weights and biases of the ANN are generated randomly.

4.1.2. Data processing

In addition to setting up the architecture parameters of the ANN as discussed above, the following issues need to be considered before an effective ANN prediction can be carried out.

1) Prediction for Working Hour Only

It has been recognized in the surveyed literature that it is important to adopt a day-typing procedure to separate energy data with distinct load patterns into different prediction groups. Energy prediction can be made within each group instead of on the entire data set. An obvious separation can be drawn between weekdays and weekends (holidays). In this simulation, the Laval building typically operates from 7 a.m. to 7 p.m. Monday through Friday. It is assumed that the chiller usage is zero during non-working hours. Thus, non-working hours are removed from the data set and prediction is made only for the working hours. With all the data, the implementation would be easier but the prediction errors increase as demonstrated below.

Two experiments were conducted by using measurements associated with June to predict the electric demand of the chiller in July. In the first test, the training process is carried out using only the working-hour measurements. For the second test, the entire measurements associated with the month of June were used. The prediction accuracy is measured by the coefficient of variation (CV) and the root mean square (RMSE) defined in Appendix A. The CV and RMSE values for both experiments are listed in Table 4.

Table 4. Comparison of the CV and RMSE values associated with excluding and including non-working hour measurements in the prediction model.

Training Data	CV [%]	RMSE (kW)
Working hour only	20	27.0
All hours included	67	34.4

Clearly, removing the non-working hour from the entire data set significantly improves the accuracy of the prediction. Therefore, the simulation results to be presented below assume that all the non-working hour data is removed.

2) Normalization

Because the numerical range of the input and output variables may be quite different for some applications, it is often useful to scale the input and output variables so that the training process does not suffer from severe numerical round-off effects. In the following experiments, all input and output variables are scaled to have values between –1 and 1. This scaling operation is accomplished by calling the MATLAB utility function

```
[dinn, minp, maxp, doutn, mint, maxt ] = premnmx(matin, matout);
```

where matin and matout are matrices containing the original input and output, and dinn and doutn are the scaled input and output matrices. The mint and maxt matrices are required to scale the ANN output back to its original units after the training process is completed. The postprocessing can be accomplished by calling the MATLAB utility function

```
y = postmnmx(yn, mint, maxt);
```

where yn is the normalized output produced by an ANN, and y is the unscaled version of the output.

3) Dimension Reduction

As mentioned in Section 3.4, it may be helpful to use the technique of principal component analysis (PCA) to reduce the dimension of input data and to reduce the redundancies among the data. The MATLAB ANN Toolbox provides a set of functions to accomplish this task. To perform PCA on a set of input vectors contained in a matrix called pmat, one must first call

```
[pn, meanp, stdp, tn, meant, stdt] = prestd(pmat, tmat);
```

to preprocess the network so that the normalized input (pn) and target (tn) both have zero mean and variance 1. A subsequent call to the function prepca produces a new input matrix pp that contains the principal components of the normalized pmat.

```
[pp, transMat] = prepca(pn, 0.01);
```

The matrix transMat contains the principal component transformation matrix, which is not used in subsequent computation. The output pp is obtained from the product of transMat and the input vector pn. The matrix pp does not contain the normalized version of the original input vectors. Instead, it contains a synthesized version of the original input. The number of vectors in pp may be significantly less than that contained in pmat (or pn). The level of reduction in dimension is determined by the second argument of prepca. This argument specifies that only those principal components that contribute more than a minimum fraction of the total variance are returned in the first output argument pp. These vectors will be uncorrelated. Here prepca eliminates those components that contribute less than 1% to the total variation in the data set.

4.1.3 Training and Testing – Static Prediction Model

Twenty-five percent of the data are set aside for testing. The rest of the data is used to train the ANN. In particular, the measurements corresponding to the second week of each month is put aside for testing or prediction. This is a rather arbitrary choice. Since the data set used in this section originates from a computer simulation. Such a choice is not unreasonable.

Based on visual inspection of the variation in Tl(t) and Te(t), it is concluded that these two variables behave in a similar way though they may have different uses. Thus, there is no reason to keep both of them as input elements to the ANN. Hence, $Te(t)$ is removed from the input.

4.1.3.1 Experiment 1 – Modeling the nonlinear mapping between Electric Demand *E*(t) and the temperatures

When $H(t), Td(t)$, $Tw(t)$ and $Tl(t)$ are used as the input variables to train the ANN, it is expected that the weights and biases obtained at the end of the training process will capture the natural dependency of the chiller electric demand with respect to $Td(t)$, $Tw(t)$ and $Tl(t)$ at the t-th hour. Because the number of input elements is relatively small, the Levenberg-Marquardt algorithm was used to train the ANN. Training only takes 4.9 seconds. Figure 5. shows that the mean square error (MSE) between the output of the ANN and the target electric demand decreases rapidly during training. The MSE reaches below 10^{-3} kW in less than 10 epochs.

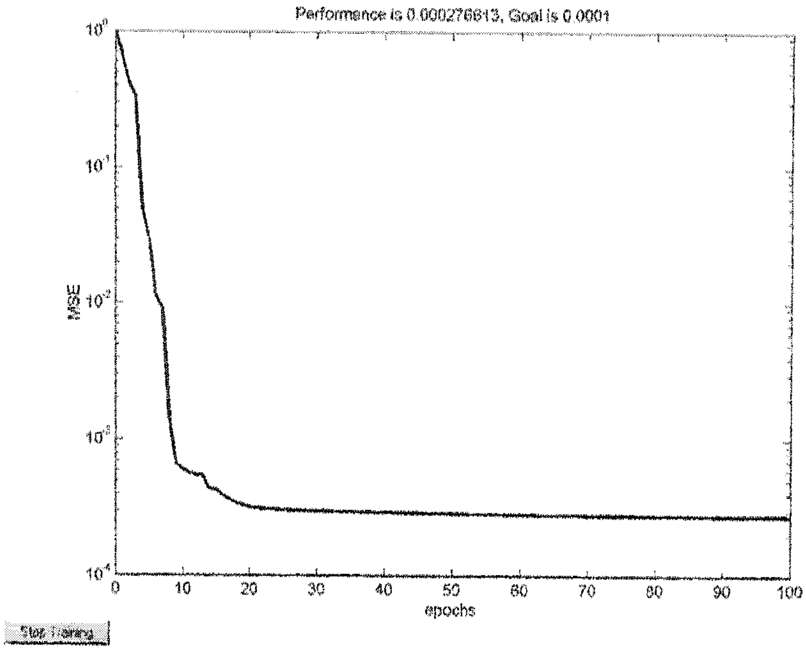


Figure 5. Training history

In Figure 6, the final output of the ANN upon the completion of training is compared with the target chiller electric demand. The difference between the solid (the ANN output) and the dash (the target data) curves is hardly visible.

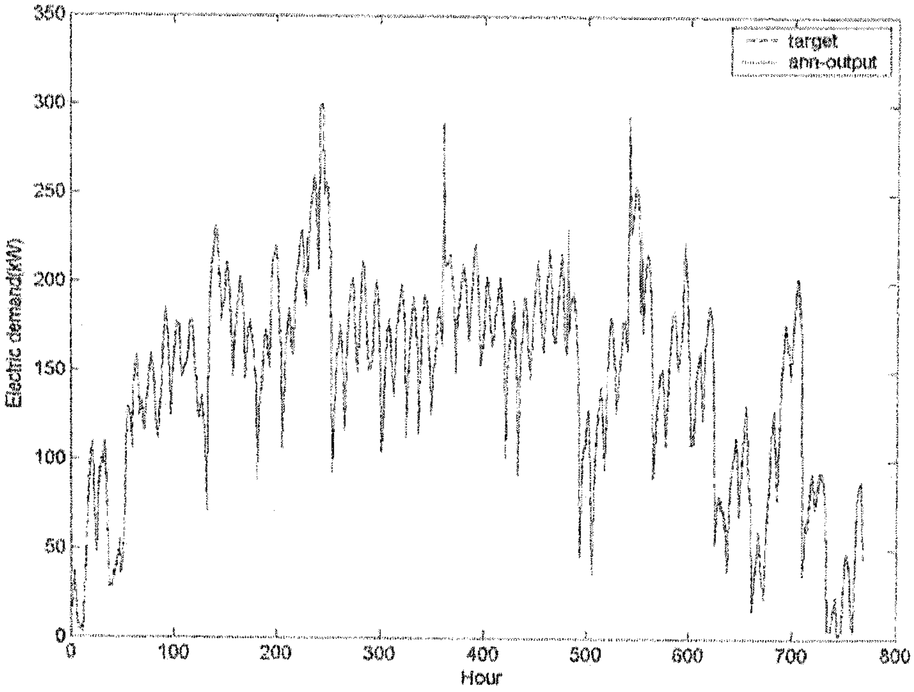


Figure 6. Comparison between the target and the ANN output during training phase

Figure 7 shows the difference between final output of the ANN and the target defined by *err(t) = Ep(t)- E(t)*, where *Ep(t)* is the predicted chiller electric demand obtained from the trained ANN. *err(t)* is within the interval [-6, 10] (kW).

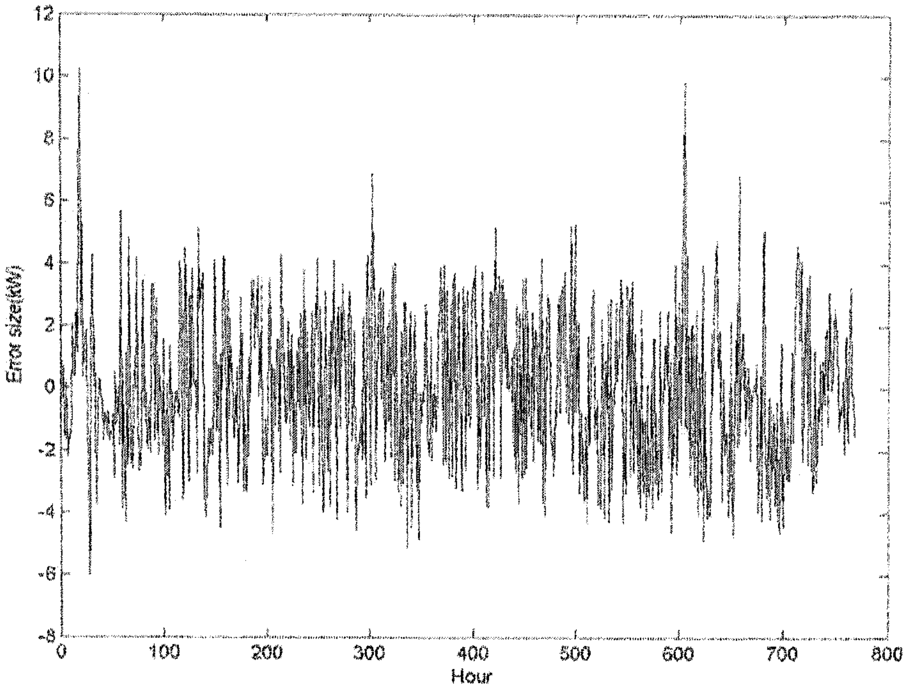


Figure 7. The difference between the ANN output and the target during the training phase

Once the training process is completed, the quality of the ANN prediction model is assessed by using it to predict the chiller electric demand associated with the second week of each month (the test data). It can be seen from Figure 8 that the ANN output matched the measured electric demand relatively well except in the places where the electric demand shows a significant fluctuation.

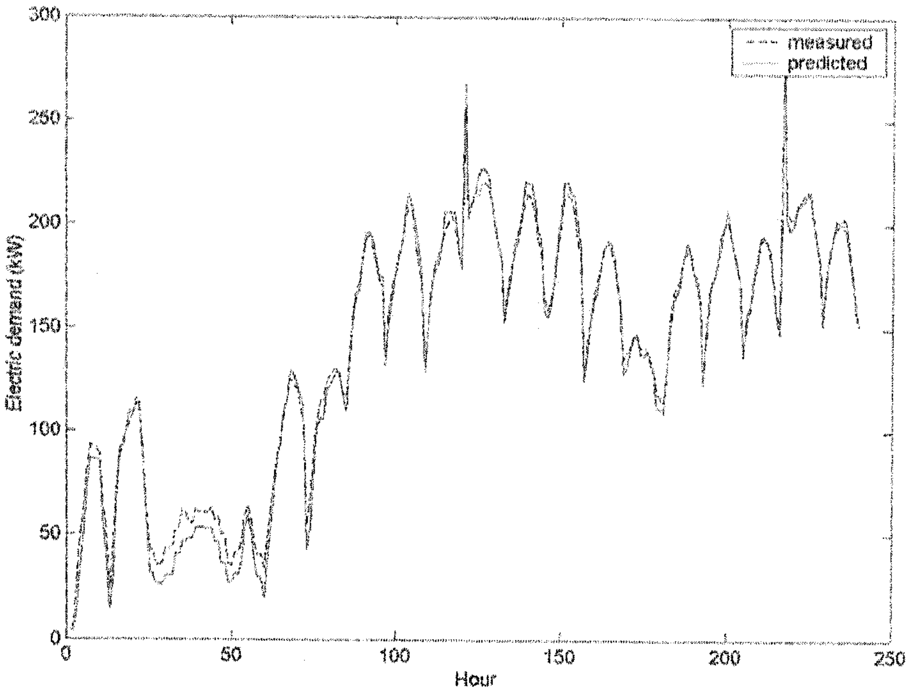


Figure 8.Comparison of the measured and predicted chiller electric demand

The distribution of the error is plotted as a bar graph in Figure 9. The figure shows that most of the error is in the range of [-7,10] (kW).

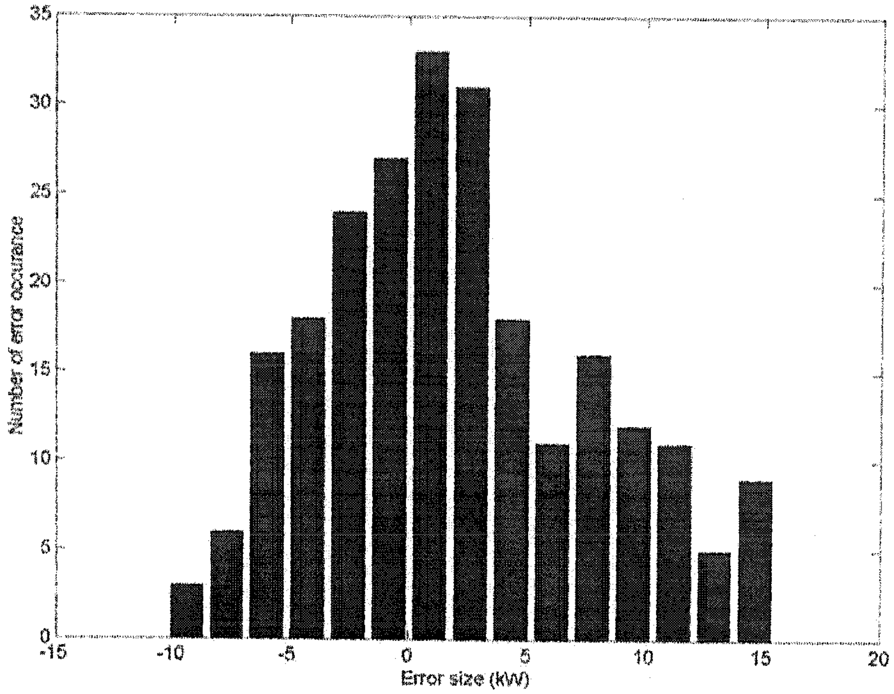


Figure 9. Error distribution detected in experiment 1 of section 4.1.3.1

The overall size of the error is measured by calculating the CV and RMSE values of the prediction error associated with the test. The CV and RMSE values from the testing data for this experiment are 4% and 6.10 kW respectively.

4.1.3.2 Experiment 2 - Using time-lagged measurements as inputs

Although Experiment 1 indicates that the ANN model constructed in that experiment can effectively capture the nonlinear relationship between the chiller electric demand E(t) of the Laval building and various temperature measurements, the model cannot be used directly in practice. The reason for this is that the input variables *Td(t), Tw(t)* and *Tl(t)* are not available until the end of the hour. Thus one must resort to using data collected in the past or time-lagged measurements as input to the ANN prediction model.

One possibility is to use *H(t)*, *Td(t-1)*, *Tw(t-1)* and *Tl(t-1)* as input variables to train the ANN to predict *E*(t). It is observed that this change of input vectors to the ANN leads to a longer training period. Training takes 11.4 seconds. Even when the Levenberg-Marquardt algorithm is used, the mean square error between the ANN output and the target decreases slowly to zero. In Figure 10., the final output of the ANN upon the completion of training is compared with the training target. Although one can see some deviation of the ANN training output from the target, the figure mostly shows a fairly close match between the two curves. In Figure 11. the difference between the final output of the ANN and the target is plotted. It is observed that the size of the error, *err(t)*, ranges from –100 kW to100 kW. This error interval is much larger than the one observed in Figure 7. However, a close examination reveals that this large error interval is attributed to a few "spikes" in the error curve. If one excludes these "spikes", most of the prediction error appears to lie within [-60, 40] (kW). Nonetheless, the presence of the large error "spikes" in the ANN training results indicates that it is somewhat difficult to capture the dependency between the temperature measurements collected in the past and the chiller electric usage at the current time *t*.

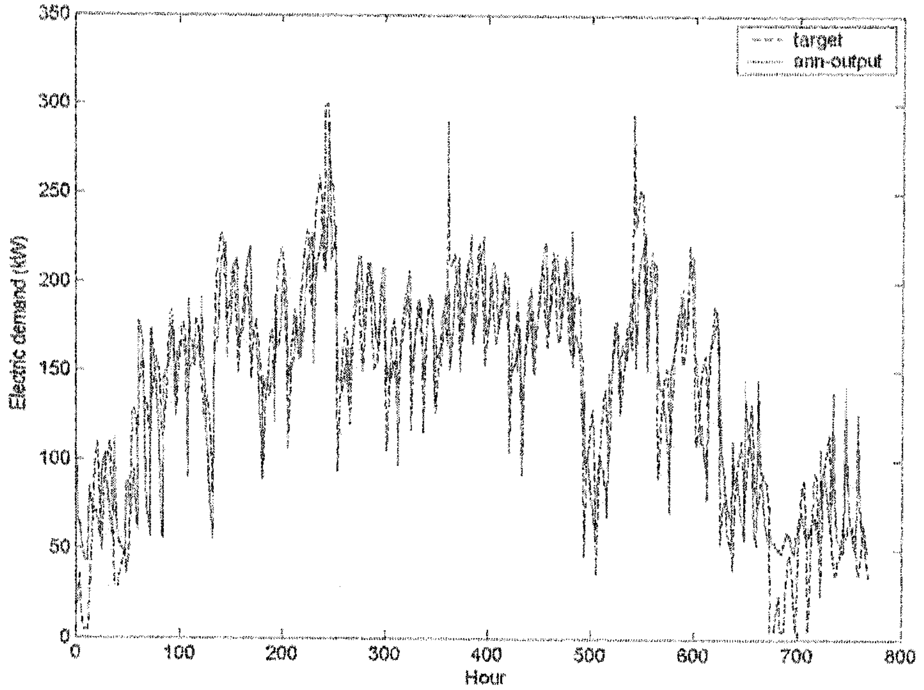


Figure 10.Comparison of the ANN output and the training target when the input consists of *H(t)*, *Td(t-1)*. *Tw(t-1) and Tl(t-1)*.

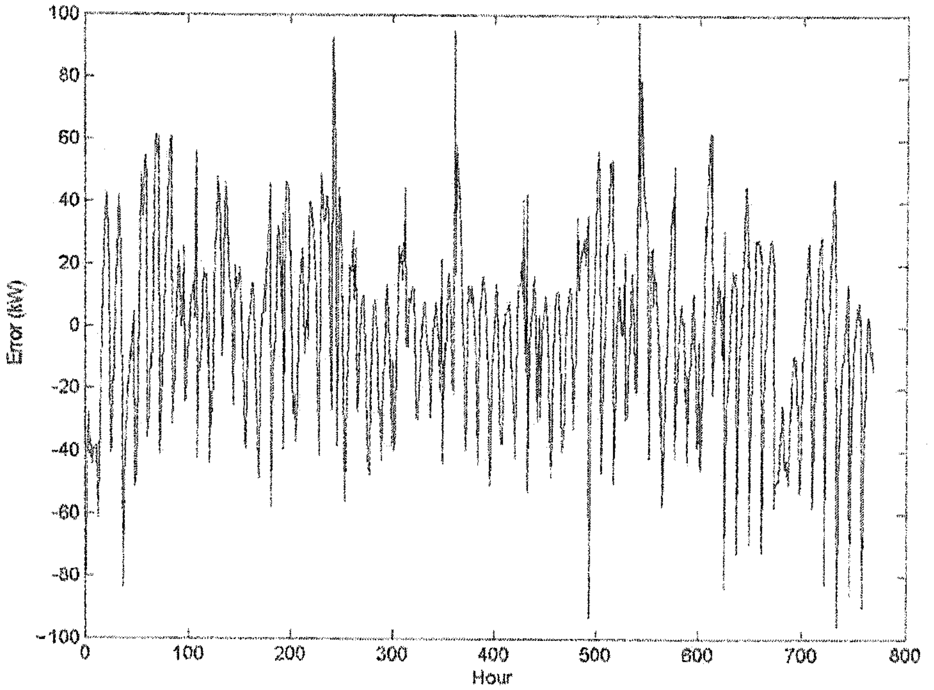


Figure 11. The difference between the ANN output and the training target when the input consists of *H(t)*, *Td(t-1)*. *Tw(t-1) and Tl(t-1)*.

This difficulty is even more pronounced when the ANN model is applied to the test data for prediction. The ANN output and the measured electric demand are plotted in Figure 12. This figure shows that the predicted electric demand matches the measured electric demand reasonably well most of the time, except near a few time locations where the ANN model significantly under-predicted and over-predicted the true electric demand. The most severe under-prediction seems to occur at the beginning of each second week.

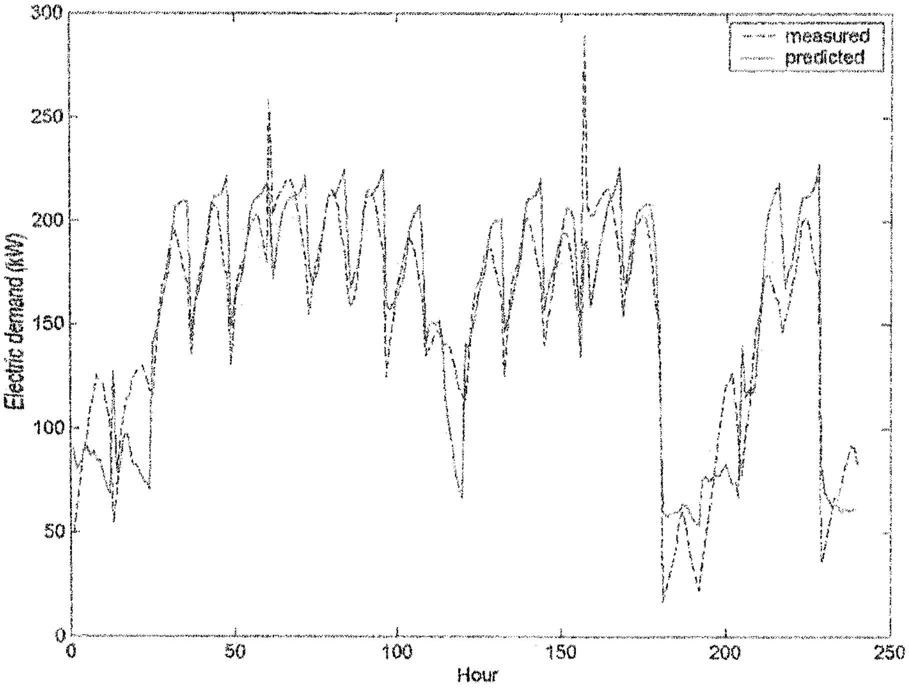


Figure 12. Comparison between the ANN predicted electric energy usage and the actual energy usage when the ANN input consists of *H(t)*, *Td(t-1)*. *Tw(t-1) and Tl(t-1)*.

This type of difficulty contributed to the relatively large CV and RMSE value observed in this experiment. The CV and RMSE values of testing data observed in this experiment are 16% and 25.46 kW respectively. It can be observed from the error distribution curve shown in Figure 13. that most of the error lies between –40 and 40 kW.

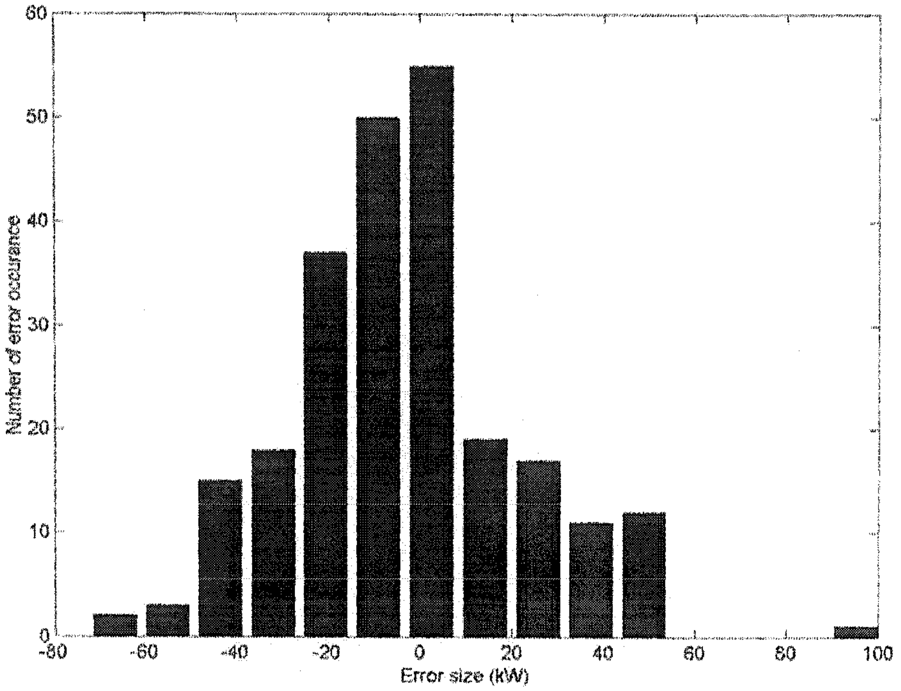


Figure 13. Error distribution when the ANN prediction is made by using of *H(t)*, *Td(t-1)*. *Tw(t-1) and Tl(t-1)* as the input.

4.1.3.3 Experiment 3 – Using additional time-lagged measurements as inputs and PCA to reduce the dimension of the input.

To improve the prediction accuracy, past history temperature measurements with longer periods of lag are included as additional input data to train the ANN model. The rationale for including additional temperature measurements collected in the past is that the extra input may help establish the trend of temperature variation and its impact on the present electric energy usage. However, the question of how many time-lagged variables should be included remains to be addressed.

Many time lagged input combinations with at most six hours delay are tried and tested. It is assumed that *H(t)* and *Tl(t-1)* are always chosen as input variables. Table 5 lists a few of these combinations and the resulting CV and RMSE values obtained from testing the corresponding ANN. Each row in the table specifies a particular combination of time-lagged temperature input. If a particular time-lagged variable is chosen as an input variable, a '×' is placed in the column labeled with that variable. It is seen from Table 5 that different combinations of the time-lagged temperature measurement can result in quite different prediction accuracy. It is interesting to note that including all temperature measurement collected at time *t-i*, for *i*=1,2,...6, does not necessarily gives the best performance.

Table 5. Comparison of using different combinations of time-lagged temperature measurements as the ANN input to predict chiller electric usage at time *t*.

Pattern	**Td**						**Tw**						**CV[%]**	**RMSE (kW)**
	t-1	t-2	t-3	t-4	t-5	t-6	t-1	t-2	t-3	t-4	t-5	t-6		
1	×						×	×	×	×	×	×	15.1	22.0
2	×	×					×	×					17.3	26.5
3	×	×	×	×	×		×	×	×	×	×	×	17.4	26.7
4	×	×	×	×	×	×	×	×	×	×	×	×	22.4	34.4

The best prediction result is obtained when the choice of input variable follows the first pattern listed in Table 5. With this set of input variables, the CV and RMSE values of the prediction (on the test data) dropped to 15.1% and 22.0 kW respectively.

It should be noted that when the number of input variables increases, it becomes more time consuming to use the Levenberg-Marquardt algorithm to train an ANN. This is due to the fact that the Levenberg-Marquart algorithm must solve a linear system to obtain a search direction for each epoch. The dimension of the linear system is proportional to the number of input variables. As the number of variables increases, so is the computational cost associated with each epoch. When the number of input variables becomes larger than six, the gradient descent algorithm appears to be a more efficient training algorithm. Even though the convergence rate of the algorithm is linear, the computational complexity within each epoch is lower than that associated with the Levenberg-Marquardt algorithm. Thus, the experiments performed above utilized the gradient descent algorithm in the training process.

The reason why adding all possible time-lagged measurements may not be effective is that adding too many measurements collected in the past may introduce undesirable redundancy. The redundancy in the ANN input makes it difficult for the back-propagation algorithm to capture the optimal weights and biases for the desired ANN model. Thus, a meticulous selection of time-lagged data must be used to train the ANN model. However, in practice, it is not possible to try all possible combinations of lagged timed temperature measurements before a prediction is made. As discussed in Section 3.4, this problem can be overcomed by using the technique of principal component analysis (PCA).

In the following experiment, PCA is used to select appropriate input data from *H(t), Tl(t-1), Td(t-k), Tw(t-k),* for $k = 1, 2, \ldots 6$. Six principal components that contribute to more than one percent of the variance of all past history data are retained. Training takes 9.8

seconds. With the help of PCA, the training time reduced to 9.9 seconds, the CV and RMSE values of the prediction are reduced to 7.2% and 11.01 kW respectively. Figure 14. shows that the predicted electric demand curve matches well with the measured energy curve. The CV and RMSE values of the prediction obtained in this experiment are much better than the ones obtained in Experiment 2. If the error spike around the 160-th hour is excluded, the CV and RMSE values would be much smaller. This phenomenon is also revealed in Figure 15. in which the distribution of the error is plotted as a bar graph. Most of the error lies within [-20, 20] (kW), except for a few outliers in Figure 15.

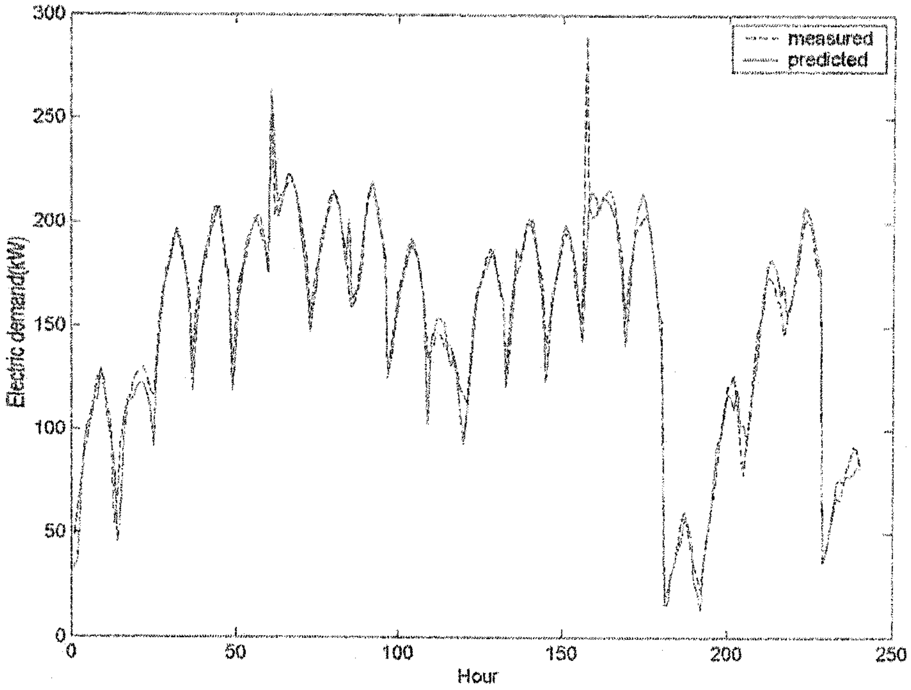


Figure 14. Comparison of the predicted and measured chiller electric energy usage when the ANN input consists of the principal components associated with all possible time-lagged temperature measurements.

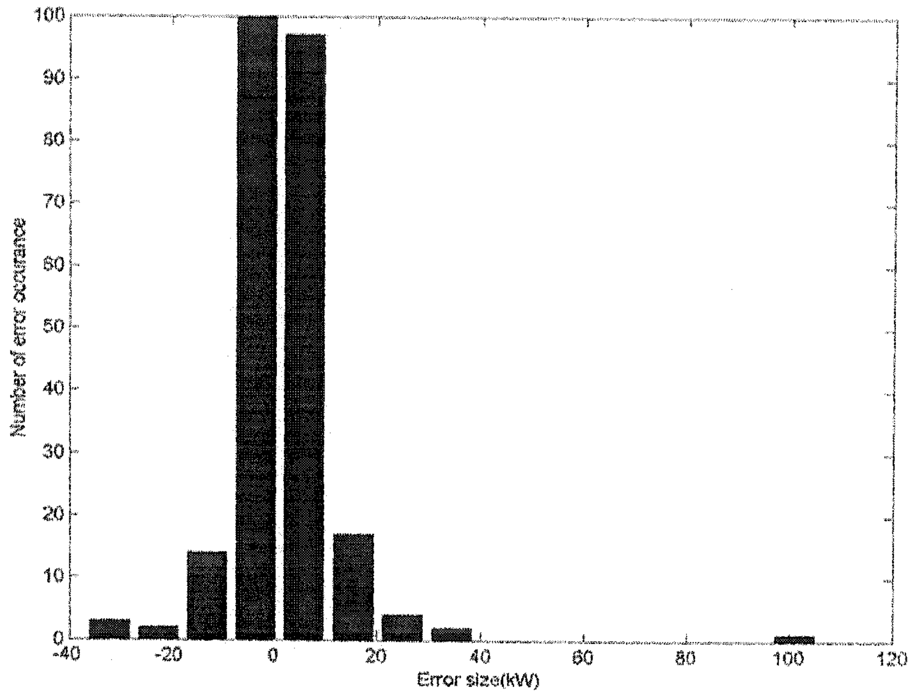


Figure 15. Error distribution associated with the PCA-enabled prediction.

To summarize the experiments performed in this section, the CV and RMSE values obtained in each experiment are listed in Table 6. Clearly, the nonlinear relationship between input variables *Td(t), Tw(t), Tl(t), H(t)* and output variable *E(t)* can be accurately described by the ANN constructed in Experiment 1. In practice, the prediction must be made using measurements collected in the past. Measurements collected from the immediate past hour or several hours back can be used to improve the prediction accuracy. When combined with PCA, the ANN can produce fairly accurate prediction in an efficient manner.

Table 6. Comparison of CV and RMSE values obtained from experiments in Section 4.1.3

Experiment (static)	CV [%]	RMSE (kW)	Training time(s)
1(instantaneous mapping)	4	6.10	4.9
2 (using one- hour lag inputs)	16	25.5	11.4
3 (using six- hour lag inputs)	7	11.4	9.9

4.1.4 Training and Testing: On-line Prediction Model

There are two simple ways to modify a static ANN to make it adaptive and suitable for on-line energy prediction. The modified ANN model can take advantage of new measurements that become available on a continuous basis. The first approach simply accumulate all the measurements collected up to time *t*, and retrain the ANN periodically using the entire set of measurements. This is presented to as *accumulative training* in Section 3.6. The second approach maintains a fixed amount of training data by discarding old measurements while adding new measurements. It is presented as a *sliding window* training model in section 3.7. In this section, computational experiments are carried out to examine the effectiveness of both approaches.

4.1.4.1 Accumulative training

In the following set of experiments, the temperature and chiller electric demand measurements are set aside for the month of June, and this portion of the data file is used to establish (through training) what is called a 'baseline' ANN model for the chiller electric demand prediction.

The 'baseline' ANN model is used to predict the chiller electric demand associated with the first day of July. Once the prediction has been performed, the hourly temperature and electric demand measurements associated with the day being predicted is added to the initial data set allocated for 'baseline' training. This updated data set is used to retrain the ANN model for carrying out subsequent predictions. The weights and biases that emerge from the 'baseline' model are used as the initial weights and biases during the retraining process. Since these initial weights and biases are likely to capture some features of the nonlinear mapping between the independent temperature variables and the electric energy to be predicted, it is conceivable that retraining will not take as long as the baseline training. This behavior is confirmed in our experiments. The results presented below compares the impact of different choices of input variables on the accuracy of the accumulative on-line prediction model. In particular, the possibility of using past temperature and electric energy measurements as input to train an ANN model is examined.

Experiment 4 - Accumulative training with time-lagged temperature measurements as input

Similar to static prediction, the input to the ANN model consists of a combination of the past history temperature measurements *Td(t-k),Tw(t-k)* and *Tl(t-1)*,where $1 \leq k \leq 6$. The output of the ANN is the chiller electric demand *E(t)*. PCA is applied to the time-lagged temperature measurements to remove the redundancy in the input and to reduce the dimensionality of the data. Only components that contribute to more than 1% to the variance in the potential input are retained. Six principal components emerged as the input to the ANN after PCA has been applied.

The baseline model is modified and retrained daily by including the actual electric demand and temperature measurements collected on the previous day in the training data set. No difficulty is encountered in the training process. The value of MSE between the target and ANN training output converges to zero rapidly. Training takes 50 seconds for the whole experiment. Figure 16 shows that the predicted chiller energy usage matches the actual usage reasonably well in general. However, mis-predictions can be observed at the beginning and in the middle of July (hour 120 and 145). The error distribution shown in Figure 17 indicates that most errors are concentrated in the range [-25,25] (kW). The CV and RMSE values obtained from testing are 15% and 28.26 kW respectively. These numbers are comparable to the CV and RMSE valued observed in static ANN models.

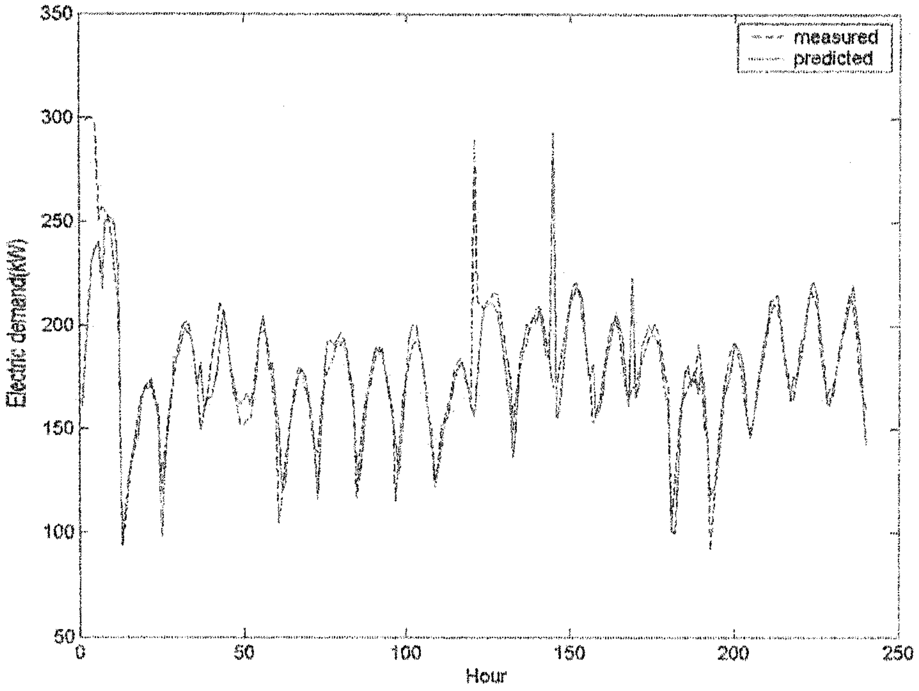


Figure 16. Comparison between the measured chiller electric energy usage (the dashed curve) and the ANN predicted energy usage (the solid curve) with an ANN trained incrementally.

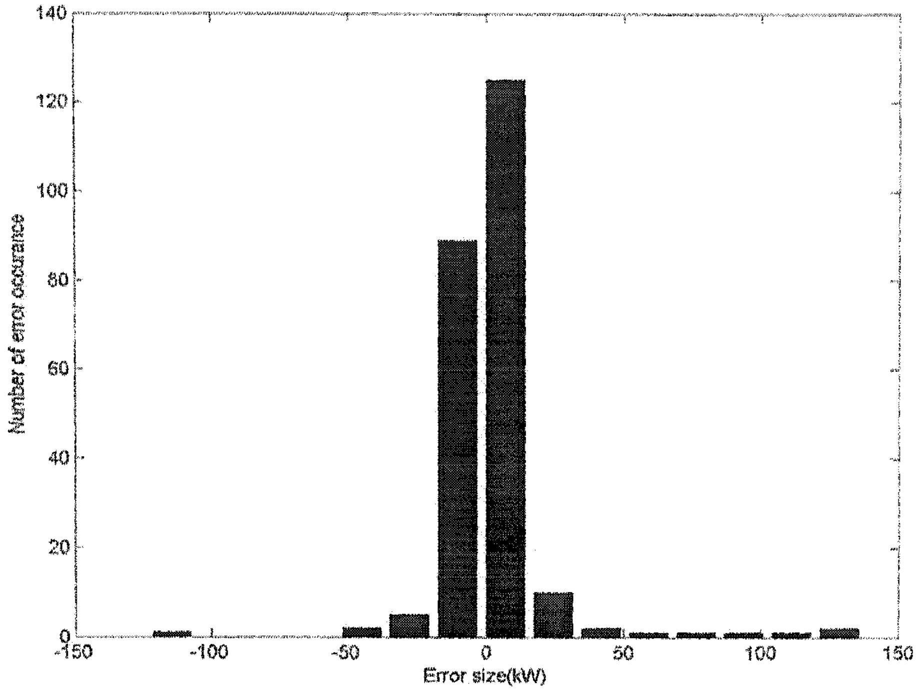


Figure 17. Error distribution associated with the accumulatively trained ANN used in Experiment 4.

Experiment 5 - Accumulative training with time-lagged chiller energy usage as inputs

It is conceivable that the chiller electric demand in previous hours can have some impact on the chiller electricity demand at the present hour. Thus, it is hoped that by including past energy usage data into the set of input variables, the accuracy of the prediction can be improved. In this experiment, *E(t-k)* is added into the set of input variables, the ANN is constructed and trained in a way similar to that carried out in Experiment 4.

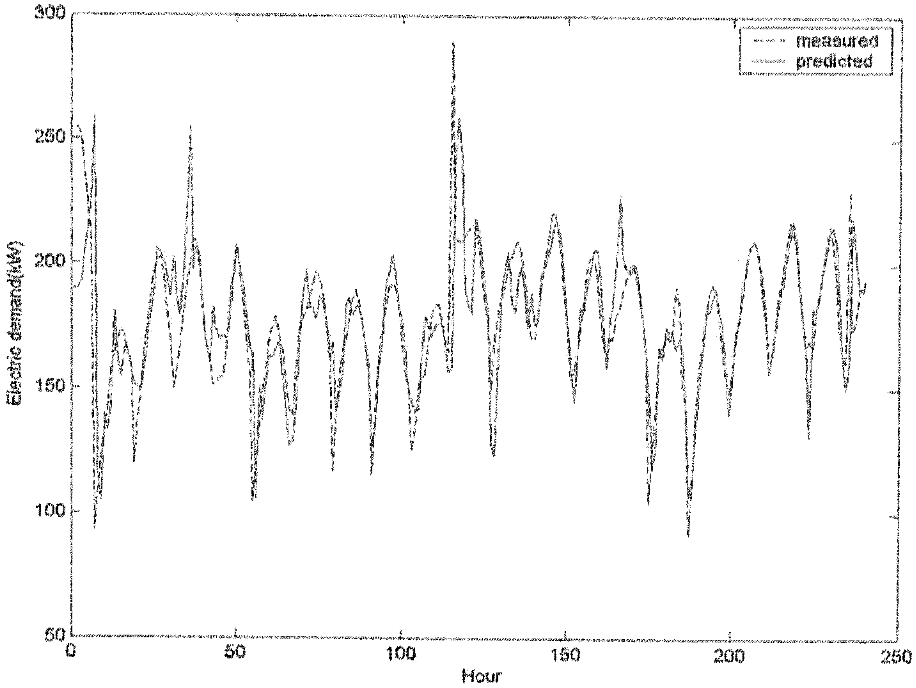


Figure 18. Comparison between the measured chiller electric energy usage (the dashed curve) and the ANN predicted energy usage (the solid curve). The ANN is trained incrementally using accumulated measurements collected in the past. The input variable consists of *Td (t-k), Tw(t-k), Tl(t-1), h(t)*, as well as *E(t-k)*, k = 1,2,...6,

It is shown in Figure 18. that the predicted energy usage matches the measurements reasonably well except at the beginning of the first week of August (hour 120 in Figure 18). The obvious under-prediction and over-prediction raises some concerns on the reliability of the prediction model. Figure19 shows that most of the errors falls within the range [-20,20] in kW. No significant improvement is observed from either Figure 18. or Figure 19. compared to the prediction results presented in Figure 16 and Figure 17. The CV and RMSE values obtained from this experiment are (see Table 7) are 17% and 28.92kW respectively.

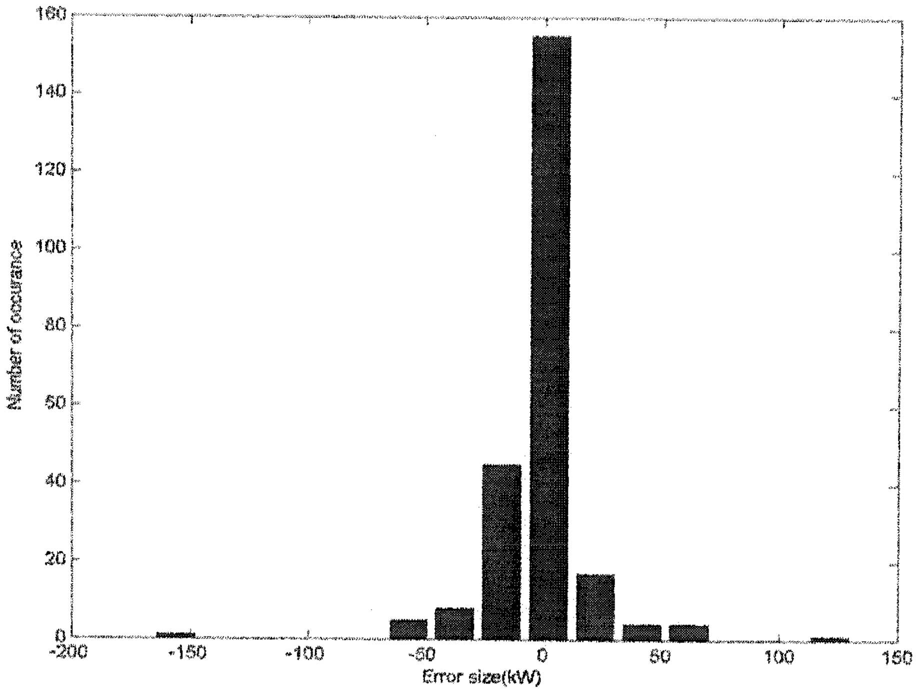


Figure 19. Error distribution associated with the accumulatively trained ANN used in Experiment 5.

Table 7. Comparison of CV and RMSE values associated with two experiments that use accumulatively trained ANN.

Experiment	CV [%]	RMSE (kW)
4	15	28.26
5	17	28.92

4.1.4.2 Sliding window training

As described in Section 3.6, it seems natural to consider limiting the volume of the training data by setting up a selection window before a forecast is to be made for a prescribed period of time. The training data will only consists of the temperature measurements and electric energy usage enclosed by this window. The selection window is shifted forward with respect to time as new data becomes available. The size of the window is fixed.

To compare with accumulative training approach, the ANN model used in the sliding window approach has the same architecture as the one used in the accumulative prediction model. The same parameter settings (such as the learning algorithm, training convergence tolerance and the maximum number of epochs allowed) are used for the following experiments. The experiments use variables *Td(t-k)*,*Tw(t-k)* and *Tl(t-1)*,where $1 \leq k \leq 6$ as the input to the ANN. The output of the ANN is the chiller electric demand *E*(*t*).

Experiment 6 - Sliding window training using temperature data collected in previous hours

Several choices of window sizes have been experimented. It is shown from Table 8. that experiments show that a window size of 20 working days seems to provide a reasonable balance between accuracy and computational complexity per on-line prediction cycle. Thus, all subsequent experiments use the window size of 20 working days.

Table 8. Comparison of results using different window size

	CV [%]	**RMSE (kW)**
10 days	25	45.5
20 days	4	8.05
30 days	2	3.00
40 days	46	82

The temperature measurements and the electric demand associated with the first twenty days of June were selected as the initial set of training data. The prediction is made on a daily basis. Thus, once the initial training is completed and a prediction has been made for the electric demand on the twenty-first working day in June, the hourly temperature and electric demand measurements corresponding to the first working day of June is dropped from the training data, and the temperature measurements and the actual electric demand associated with the twenty-first day is added into the training data. Consequently, the volume of training data remains the same, and the selection window is shifted forward in time by one day. It is found that training only takes 45 seconds for the whole experiment, which is less than that of accumulative training.

After PCA is applied to the initial set of time-lag temperature measurements to select principal components that contribute to more than 1% to the total variance in the temperature measurements, six principal components emerged as the ANN input. However, the number of principal components may change when the sliding window is updated. More or fewer principal components may appear as daily training and prediction move forward. When the number of principal components associated with the new training data set is different from the one associated with the previous training data, one cannot restart from the ANN model obtained from previous training cycle. Weights and biases must be reinitialized randomly, and the training may take more time.

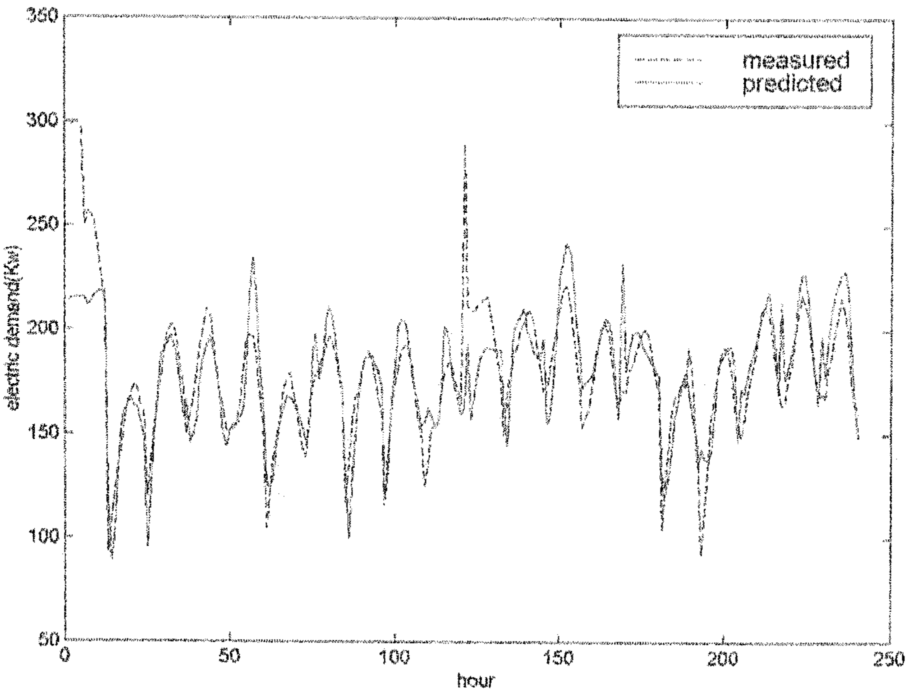


Figure 20. The comparison between the predicted and actual electric usage. The prediction is made using a dynamic ANN trained by the sliding window technique.

Figure 20.shows that the overall sliding window training produces a good prediction. The predicted electric demand matches with the actual load curve reasonably well except at some hours where the actual load shows some unexpected fluctuation. Figure 21. shows that the error distribution concentrates in the range of [-25, 25] (kW). The overall CV and RMSE values obtained in this experiment are 15% and 27.73 kW respectively.

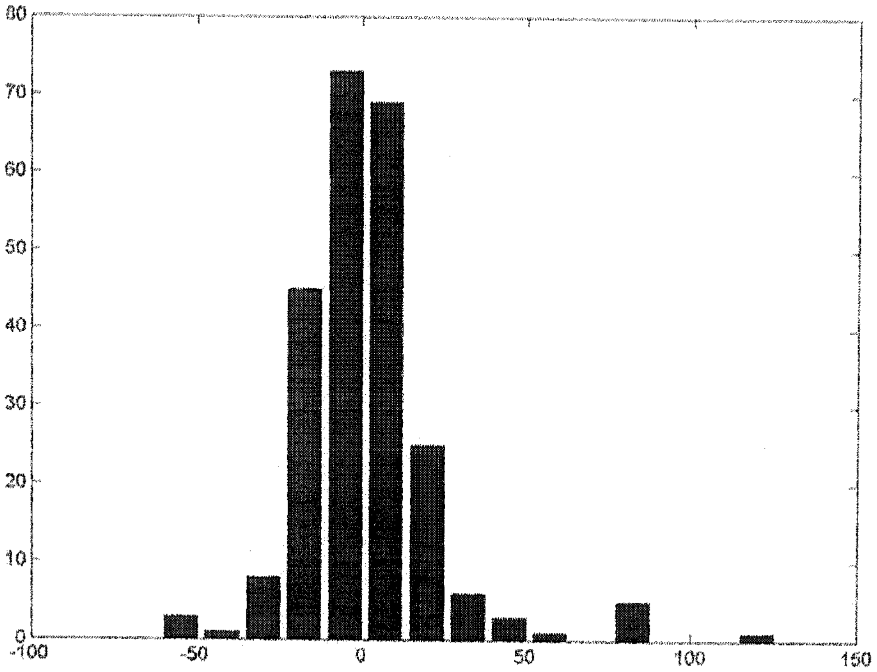


Figure 21. Error distribution associated with the adaptive ANN trained by the sliding window approach in Experiment 6

Experiment 7 - Sliding window training using temperature and energy measured in the previous hours as inputs

In this experiment, we investigate whether adding *E*(*t*-*k*) to the list of input variables: *H*(*t*), *Tl*(*t*-*k*), *Td*(*t*-*k*), *Tw*(*t*-*k*), $k = 1, 2, \ldots 6$, can improve prediction accuracy. The PCA technique is used to remove the potential redundancy in the data.

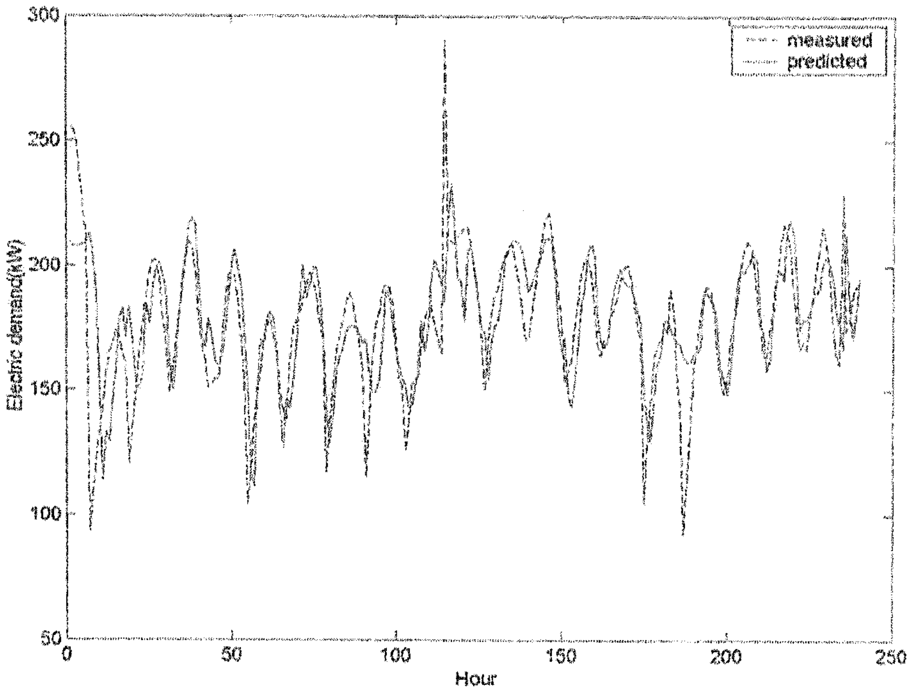


Figure 22. Comparison between the predicted and measured chiller electric energy usage. The prediction is made by an adaptive ANN trained using a sliding window approach.

Figure 22 shows that no clear improvement in prediction accuracy is observed when *E(t-k)* is added to the list of ANN input variables compared to Experiment 6. In particular, severe mis-predictions at the end of the first week of July (hour 120) remain. The error distribution pattern shown in Figure 23 is similar to the one shown in Figure 21 and it ranges [-25,25] kW. The CV and RMSE values of the prediction are 16% and 27.78 kW, which further indicates that adding E(t-k) into the list of input variable does not help much.

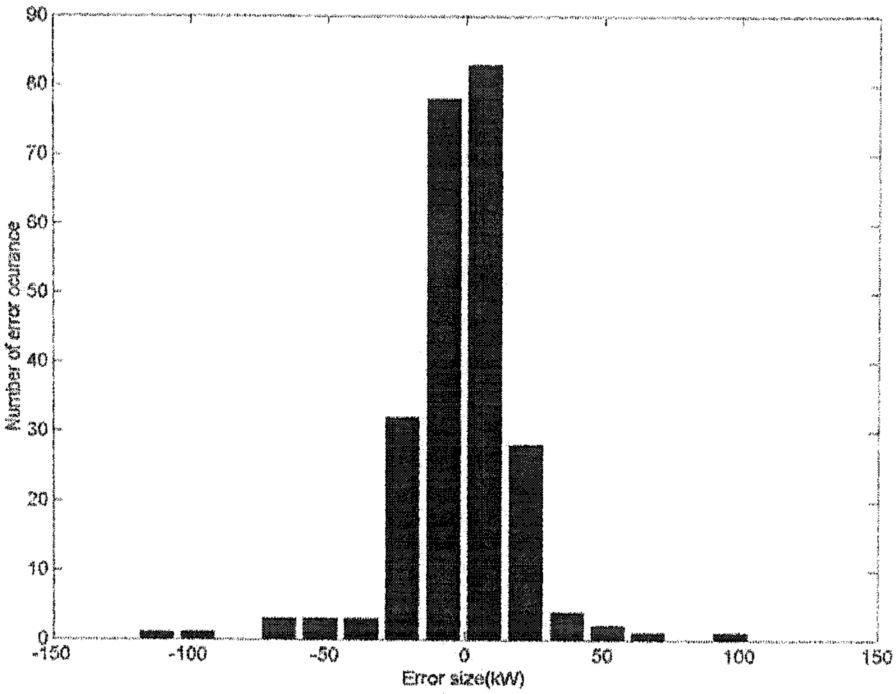


Figure 23. Error distribution associated with the adaptive ANN trained by the sliding window approach in Experiment 7.

Table 9. Comparison of CV and RMSE values associated with two experiments that use sliding window training.

Experiment	CV [%]	RMSE (kW)
6	15	27.73
7	16	27.78

Table 10. Comparison of training time for on-line predictions

Experiment	Training time (seconds)
4	50.1
6	44.8

4.1.5 Summary

The experiments carried out in this section demonstrated the success of using ANN to model the nonlinear mapping between the simulated temperature measurements and chiller electric usage for the Laval building. The experiments indicate that the chiller electric usage *E(t)* depends strongly on the temperature variables *Td(t), Tw(t) and Tl(t)* and hour of the day *H(t)*. However, for practical purposes, a weaker relationship is modeled between *E(t)* and *H(t), Td(t-k), Tw(t-k), Tl(t-k)*, *k*=1,2,...6. The use of PCA to reduce the dimension of the input and to remove the redundancy in the data allows us to include temperature measurements obtained several hours before the hour for which the prediction of the electric energy usage must be made. This turns out to be critical in using temperature measurements and electric energy usage collected in the past to predict the electric energy usage in future hours.

The ultimate goal of this section is not to construct and make use of a single static ANN model to predict future chiller electric energy usage. Instead, it is to develop a dynamic ANN model that is trained periodically so that the prediction model can adapt to new features in the data. The concept of accumulative training and sliding window training were both tested. Both approaches produced satisfactory results overall. However, due to the limited size of the training data, noticeable prediction error is observed when chiller electric demand undergoes a sudden and unexpected change.

4.2 Predicting the Gas and Chiller Energy Usage at the CANMET Center

This section contains the results of using ANN techniques to predict the gas enery demand and the chiller electric energy demand at the CANMET Energy Technology Center located in Varennes, Quebec. Unlike the simulated and noise-free data used in Section 4.1, the data used in the prediction for CANMET center consists of measurements obtained from sensors installed in the HVAC system of the building. Because the original data provided in this experiment is not prepared in a format that can be directly used by the MATLAB code developed in this thesis, the raw data must first be preprocessed and converted into the desired format. During the process of conversion, several problems associated with the completeness and accuracy of the data were discovered. Thus, first, the problems encountered and methodologies for addressing these problems are described before discussing the ANN experiments and results.

4.2.1 Data Processing

The original data obtained from CANMET Energy Technology Center is stored as a Microsoft Access (MA) file. The file contains measurements collected by Research & Development lab at the Center. The Access file contains various quantities measured hourly between 12:00 P.M. June 21, 2002 and 12:00 A.M. March 27, 2003, and between 11:00 A.M. May 8, 2003 and 0:00 A.M. July 10, 2003. The Access file is organized in a tabulated form with four columns and many rows. Each row corresponds to an hour. The columns are arranged in the following order:

1. The name of the variable measured
2. The hourly average value of the variable
3. The mode
4. The time stamp at which the variable is measured

The mode values appear to be '1's for all rows. Thus, this column is ignored in the data manipulation and conversion process described below. The rows of the original table appear to be sorted by the time stamp that appears in the last column of this table. There are typically several rows with the same time stamp indicating that several variables are measured in each hour. The rows associated with the same time stamp appear to be sorted by the variable names (listed in the first column) alphabetically.

Since the ANN energy prediction code developed in this thesis is written in MATLAB, the original data set must be converted into a matrix format that can be easily manipulated by MATLAB matrix operations.

4.2.1.1 Variable Selection

The potential candidates for the input and output variables to be used in the ANN prediction model are chosen based on the cooling and heating system diagrams provided by CANMET.

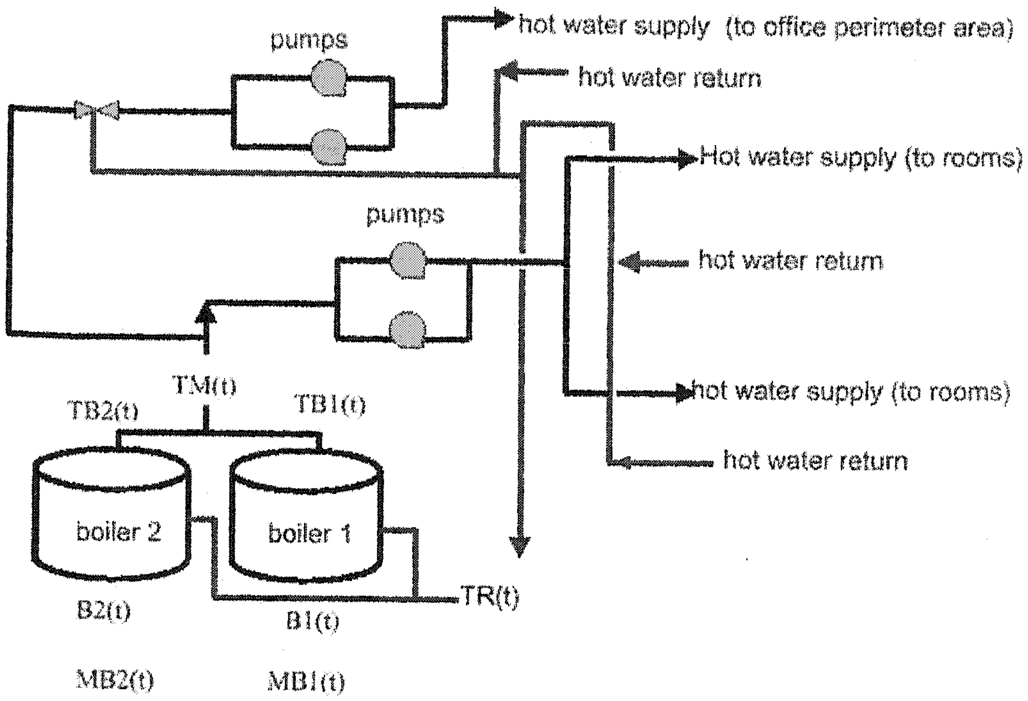


Figure 24. Heating system diagram

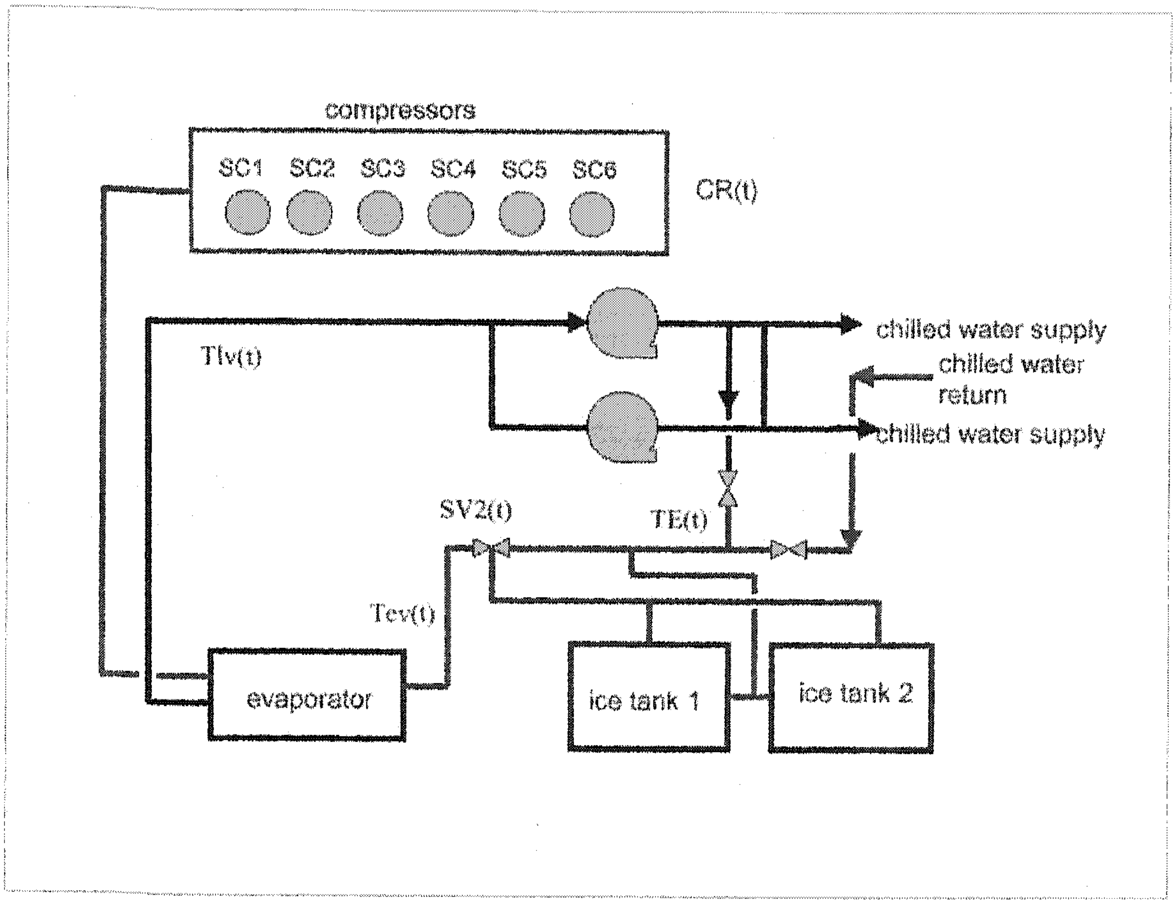


Figure 25. Cooling system diagram

The objective of this study is to predict the electric demand in kW by the chiller and the gas demand in kW of the boilers at a particular time *t*. Thus, the choice of output variables in an ANN are the electric demand of the chiller, which will be denoted by *C(t)*, and the gas demand, which will be denoted by *G(t)*.

As shown in Figure 24. and 25, the heating system of the building is completely separated from the cooling system. Thus, *C(t)* and *G(t)* will be predicted by two separate ANN models. Otherwise, if *C(t)* and *G(t)* are predicted by one network, chiller related variables are useless for *G(t)* prediction and they will become noise for ANN *G(t)* prediction, and vice versa for *C(t)* prediction.

Based on the heating system diagram, the variables listed in Table 11 are initially identified to be the independent variables that can potentially affect the variation of *G(t)*.

Table 11. Potential input variables related to the gas demand.

Variable	Description	Units
B1(t)	On/off status of boiler 1	
B2(t)	On/off status of boiler 2	
H(t)	Holiday indicator	
W(t)	On/off status of hot water	
MB1(t)	Percentage of maximum capacity boiler 1	%
MB2(t)	Percentage of maximum capacity boiler 2	%
TOD(t)	Outdoor temperature	^{0}C
TB1(t)	Temperature of the hot water supplied by boiler 1	^{0}C
TB2(t)	Temperature of the hot water supplied by boiler 2	^{0}C
TM(t)	Mixing temperature of *TB1(t)* and *TB2(t)*	^{0}C
TR(t)	Returning hot water temperature to boiler 1 and 2	^{0}C
WS(t)	Weekday schedule	

The numerical ranges of the measurements associated with some of the input variables and the output variable for the gas demand are listed in Table 12, along with the average values. This statistic calculations are made after outliers existing in the data is removed.

Table 12. Some statistics of the selected variables gas demand prediction.

Variable	Min	Max	Average	Unit
G(t)	0	625.5	114.07	kW
TB1(t)	0	100	59.07	^{0}C
TB2(t)	0	100	59	^{0}C
TM(t)	0	100	51.54	^{0}C
TR(t)	0	100	53.30	^{0}C
TOD(t)	-25.38	34.98	3.97	^{0}C
MB1(t)	0	100	3.79	%
MB2(t)	0	100	3.71	%

As to the chiller energy prediction model, the variables listed in Table 13 are initially identified to be the independent variables that can potentially affect the electric demand

Table 13. Potential input variables related to the chiller electric demand

Variable	Description	Units
$SC1(t)$	On/off status of compressor 1	
$SC2(t)$	On/off status of compressor 2	
$SC3(t)$	On/off status of compressor 3	
$SC4(t)$	On/off status of compressor 4	
$SC5(t)$	On/off status of compressor 5	
$SC6(t)$	On/off status of compressor 6	
$Te(t)$	Temperature of the chilled water entering the ice tank	^{o}C
$Tev(t)$	Temperature of the return chilled water entering the evaporator	^{o}C
$Tlv(t)$	Temperature of the supply chilled water leaving the evaporator	^{o}C
$Scw(t)$	On/off status of the chilled water	
$Smode(t)$	On/off status of ice mode control	
$Hum(t)$	Outdoor relative humidity	%
$TOD(t)$	Outdoor temperature	^{o}C
$SV1(t)$	Whether to bypass the cooling coil to fabricate ice (yes/no)	
$SV2(t)$	Percentage of return chilled water in the mixing with the water coming out of the ice tank	%
$HD(t)$	Holiday indicator	
$WS(t)$	Weekday schedule	
$CR(t)$	Electric current used by the chiller	Amp

The numerical ranges of the measurements associated with some of the input and output variables used for the chiller energy prediction are listed in Table 14, along with the average values. Variables appear to lie within the normal ranges after the outlier data is removed. The zero values detected in the chiller demand measurements indicate that the chiller can be completely shut off during some hours.

Table 14. Some statistics of the selected variables related to the electric energy used by the chiller

Variable	Min	Max	Average	Unit
C(t)	0	68.7	4.18	kW
Te(t)	-6.44	29.1	12.52	°C
Tev(t)	-4.38	29.4	11.93	°C
Tlv(t)	-6.39	29.4	12.73	°C
HUM(t)	0	99.1	63.41	%
TOD(t)	-25.38	34.98	3.97	°C
SV2(t)	0	100	61.42	%
CR(t)	0	76.28	10.99	Amp

4.2.1.2 Data Analysis

A closer examination of the data reveals that not all of these variables listed in the data file are measured at every hour. Furthermore, the number of measurements associated with each variable is different indicating a rather irregular missing pattern. For example, there are 3318 measurements for the variable *B1(t)*, but there are only 3179 measurements for *Scw(t)*. If measurements are made at every hour between 12:00 pm 6/21/2002 and 12:00 am 3/27/2003, the total number of measurements for each variable should be 6696. However, a quick inspection indicates the number of hourly measurements for each variable is roughly 50%-70% of the total number of hours between the beginning and ending period of the measurements.

By visual inspection, it is found that the gas energy usage shows an unusual distribution pattern. This problem is revealed in Figure 26 and Figure 27 in which the gas energy usage is plotted as a function of time. One can clearly see that the measured gas energy usage appears to fluctuate mainly among three levels: 156.3 KW, 312.7 KW and 469.1 KW.

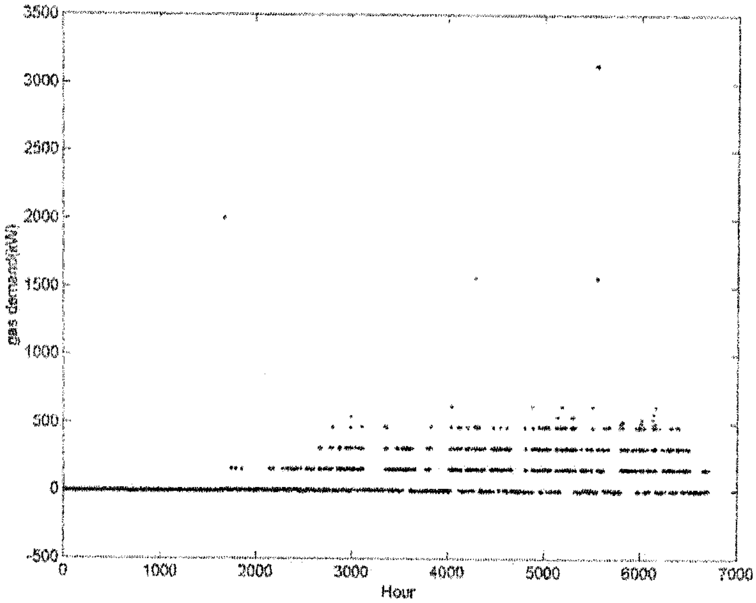


Figure 26. The gas energy demand pattern for the entire measurement period.

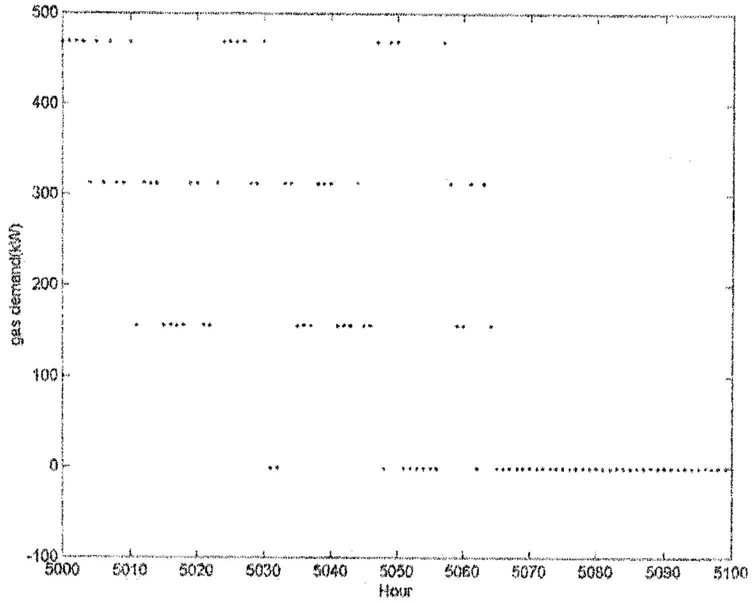


Figure 27. Zoom-in view of the energy usage pattern between the 5000-th and 5100-th hour in Figure 26

One explanation for this kind of fluctuation is that the gas metering is done in terms of pulses. Each pulse corresponds to 100 cubic feet of gas. Hence, there may be one, two, or three pulses per hour, which corresponds to the readings of 156 kW, 312 kW, and 469 kW.

An equally surprising observation is that the measurements for the chiller electric energy usage *C(t)* are completely missing between 06/21/2002 and 08/28/02. Sporadic chiller energy usage measurements are present in the data file between the period of 08/20/29 and 03/16/03. However, the number of measurements is far from complete. The load pattern of the chiller between 6/21/2002 and 03/2003 is shown in Figure 28. The negative readings correspond to the period in which the chiller usage data is missing.

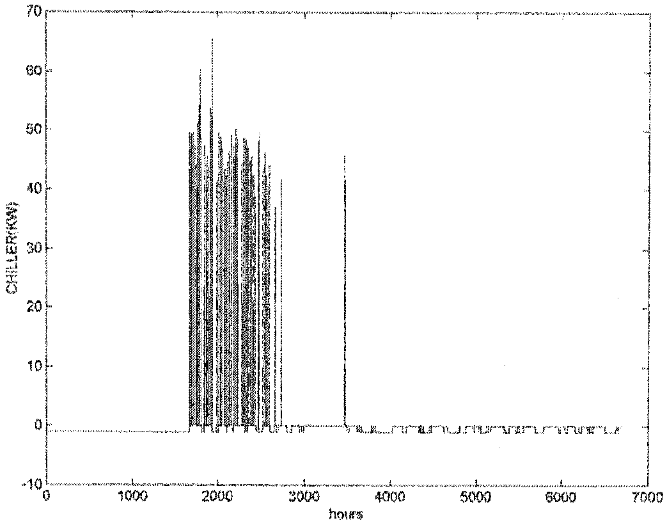


Figure 28. The electric energy usage pattern of chiller

It is clear from the above data analysis that the quality of the raw data provided by the CANMET center is quite poor. This makes it difficult to design, train and test an ANN.

It also makes it difficult to interpret the experimental results to be presented below. However, this situation is typical for the measuring systems installed in buildings to analyze and predict the energy use. The challenge is to design an ANN system to provide good on-line predictions of energy demand even if the data quality is poor. An attempt is presented in following sections.

4.2.2 The ANN Model

Just like the ANN model used to predict the chiller electric demand for the Laval building, the ANN models designed to predict the gas and chiller electric demand for the CANMET building consists of one hidden layer in addition to the input and output layers. The input layer contains n neurons from which n different inputs are fed into the network. The output layer contains one neuron from which either the predicted gas demand or the chiller electric demand is obtained. The hidden layer consists of $2n$+1neurons (Hecht 1989).

As usual, biploar sigmoid functions are used as the activation function in each neuron. In the output layer, a linear transfer function is used to allow the network to produce values outside of the range [-1, 1]. As will be shown shortly, that this choice of activation functions is not appropriate for gas energy prediction because the gas energy measurements fluctuate among a few discrete levels while the nonlinear mapping modeled by an ANN with sigmoid and linear activation functions is continuous in nature.

The training process is terminated when the mean square error (MSE) between the ANN output and the target values becomes less than 10^{-3} kW, or when a maximum of 500 epochs is reached, whichever criteria is reached first. The initial weights and biases of the ANN are generated randomly. Whenever possible, the Levenberg-Marquardt (LM) algorithm is specified to train the network. When the number of input elements or the volume of the data is large, the standard gradient descent algorithm is used for training the ANN.

4.2.3 Training and Testing – Static Gas Energy Prediction

There appears to be no previous attempts in the literature in using ANN to predict the gas energy usage of a commercial building. Part of the difficulty is that the gas demand does not seem to be a continuous function of the temperature and other environmental variables. This is observed in the CANMET data set as reported above. The discontinuity in the variable to be predicted makes it difficult to use a conventional ANN to model the relationship between the input and output because the nonlinear mapping rendered by an ANN is typically continuous (when biploar sigmoid activation functions are used). This difficulty is illustrated in Experiment 8 below.

The discrete nature of the gas energy usage provides some motivation for using a perceptron network to classify the input into different groups. However, this approach has not been successful either as we will illustrate in Experiment 10. It is conjectured that the incompleteness and poor quality of the data attributed to this failure.

4.2.3.1 Experiment 8- Modeling the nonlinear mapping between Gas Demand *G(t)* and other variables

In this experiment, a standard ANN is used to explore the nonlinear relationship between the variables listed in Table 11 and the gas demand *G(t)*. Since gas is only used in the winter, the data associated with the summer and the autumn is ignored. Unlike the complete data associated with Laval Building, too much data associated with the gas demand of the CANMET center is missing during December, 2002 and January, 2003. It is impossible to find the data associated with a whole week in each month for training. Especially in January, 2003, almost half of the data associated with January is missing. In this case, all the measurements collected in December, 2002 are selected as the training data and all the measurements collected in January, 2003 are used for testing or prediction. The gradient descent algorithm is used to train the network due to the relatively large number of input variables. The training process is terminated after 500 epochs have been completed or when MSE is reached 10^{-3} kW, whichever occurs first. Upon the completion of the training process, the mean squares error (MSE) becomes 0.06kW. This is above the convergence tolerance (set to 10^{-3}) set in advance. Figure 29 shows that there is significant mismatch between the measured and the predicted gas energy usage curve. In particular, the predicted (solid) curve is not able to capture the discrete nature of the gas energy usage pattern. The CV and RMSE values obtained in this experiment are 62% and 134.2 kW respectively.

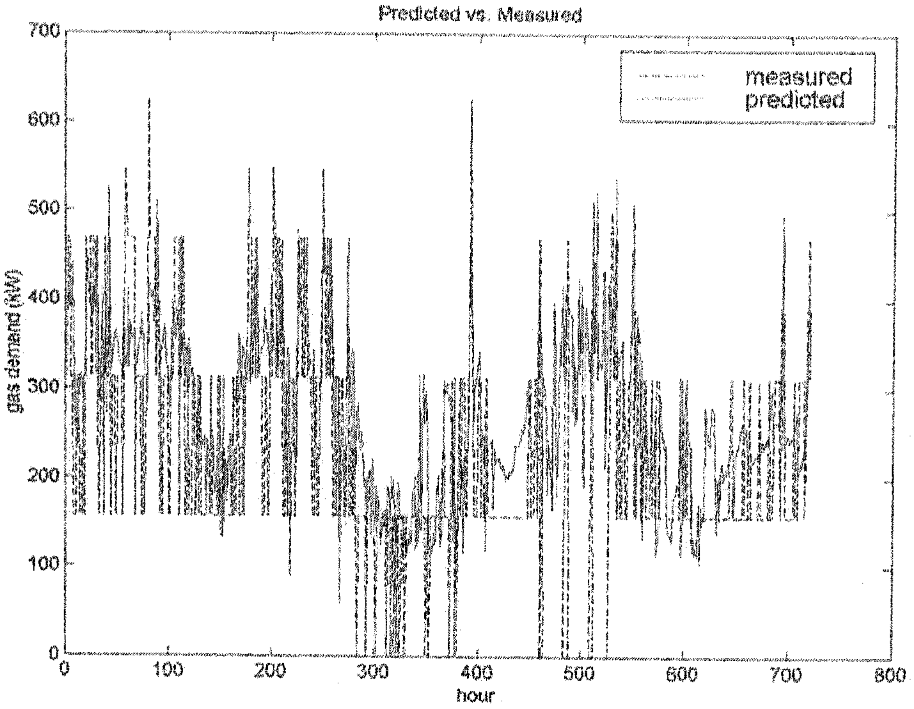


Figure 29. Comparison between the predicted and the actual gas energy usage.

4.2.3.2 Experiment 9- Gas demand prediction using measurements recorded in previous hours

This experiment demonstrates that the difficulty exhibited in Experiment 8 cannot be eliminated by simply adding temperature and gas energy measurements recorded in previous hours to the list of ANN input variables. To carry out this experiment, many measurements recorded in the past are included in the list of ANN input variables. In particular, the variables listed in Table 15 are used,

Table 15. The ANN input variables chosen in Experiment 9.

Variable	Description ($1 \le k \le 6$)
H(t)	Hour of the day
B1(t-k)	The on/off status of boiler 1 *k* hours before *H(t)*
B2(t-k)	The on/off status of boiler 2 *k* hours before *H(t)*
W(t-k)	The on/off status of hot water *k* hours before *H(t)*
MB1(t-k)	percentage of maximum capacity of boiler 1 *k* hours before *H(t)*
MB2(t-k)	Percentage of of maximum capacity of boiler 2 *k* hours before *H(t)*
TOD(t-k)	Outdoor temperature *k* hours before *H(t)*
TB1(t-k)	Temperature of the hot water coming out of boiler 1 *k* hours before *H(t)*
TB2(t-k)	Temperature of the hot water coming out of boiler 2 *k* hours before *H(t)*
TM(t-k)	Mixing temperature of *TB1(t-k)* and *TB2(t-k)*
TR(t-k)	Returning water temp of boiler 1 and 2 *k* hours before *H(t)*
G(t-k)	Gas energy used *k* hours before *H(t)*

The technique of PCA is used to reduce the dimension of the input and to remove redundancies in the input vectors. Only components that contribute to more than 1% of the total variation among the input vectors are selected. This choice yields roughly nine principal components (9 input vectors).

After the ANN with this particular set of input variables has been trained for 500 epochs, it is applied to the test data. Figure 30 shows that the predicted gas energy usage is nowhere close to the actual energy usage. The CV and RMSE values obtained in this experiment are 53% and 111.6 kW respectively.

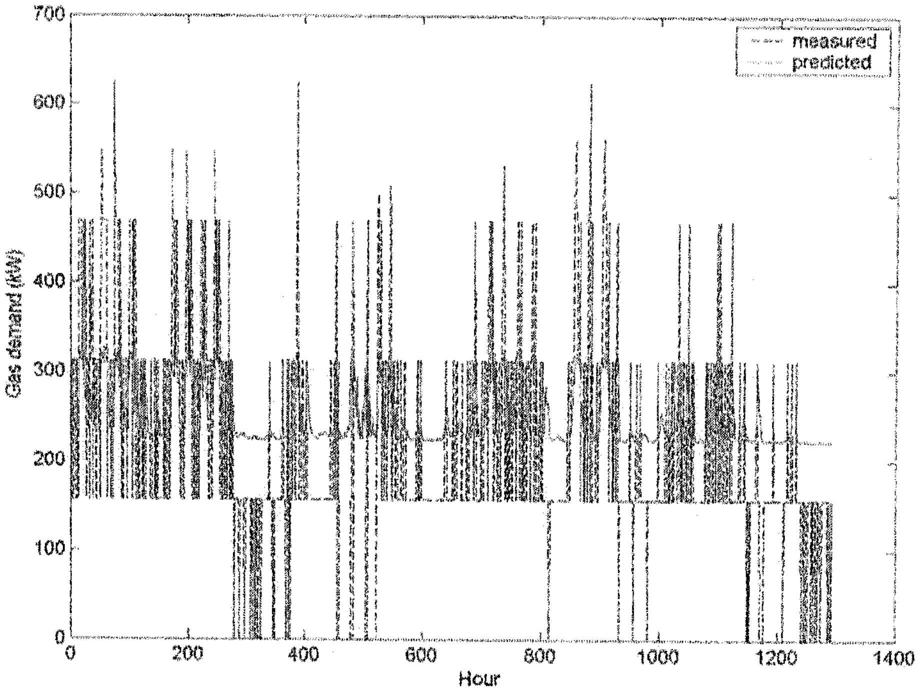


Figure 30. Comparison of the actual gas energy usage and the usage predicted in Experiment 9.

4.2.3.3 Experiment 10 – Classification of the input

The discrete nature of the gas energy usage provided motivations to use a single-layer perceptron network to identify the pattern of the input variables listed in Table 11. A single perceptron neuron produces a '1' if the net input into the transfer function is equal to or greater than 0; otherwise it produces a 0. A perceptron neuron uses a transfer function that gives a perceptron the ability to classify input vectors by dividing the input space into two regions. Outputs will be 0 if the net input n is less than 0, or 1 if the net input is 0 or greater.

The rationale for performing this experiment is simple: if the each gas energy usage level corresponds to certain temperature and operational patterns, then one should be able to classify these pattern using a perceptron network.

Thus, a simple perceptron is constructed in this experiment to explore this possibility. The input of the network consists of variables listed in Table 11. Three output units are used to match with the three discrete gas energy usage levels. Each unit can only assume the values of zero or one. The mapping between the ANN output units and the discrete gas energy levels is described in Table 16. For example, if the first output unit assumes the value of one, the second and the third output units assume the values of zeros, then the gas energy level is assigned to the level of 156.3 KW.

Table 16. The mapping between the perceptron output patterns and the gas energy usage levels.

Pattern	Output 1	Output 2	Output 3	Gas demand (kW)
1	1	0	0	156.3
2	0	1	0	312.7
3	0	0	1	469.1

To force the perceptron to produce discrete output, the binary pulse function

$$f(s) = \begin{cases} 1 & s \geq 0 \\ 0 & s < 0 \end{cases}$$

is used as the activation function.

Although this approach seems to have a great deal of potential in theory, the simple perceptron was not able to successfully classify the patterns of the input variables. Figure 31. shows that after the perceptron has been trained to classify the temperature and operational patterns, the mapping from the resulting classes to the discrete gas energy levels had only about 50% of accuracy (i.e., among the 300 predicted values, only 160 completely match the actual gas demand after statistics). The CV and RMSE values are 50% and 121.8 kW respectively.

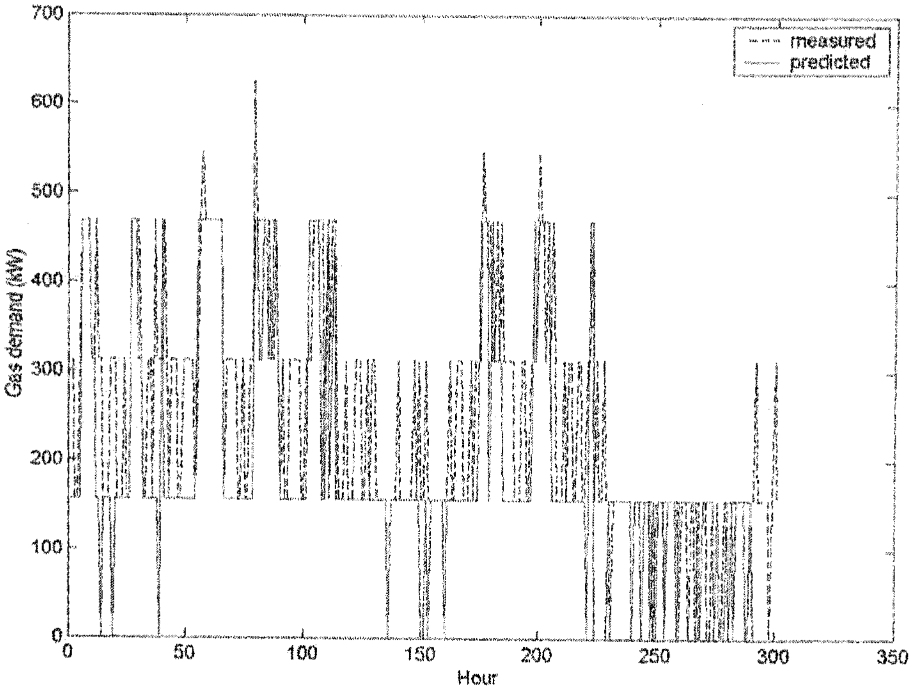


Figure 31. Comparison between the gas usage predicted by the perceptron and the actual gas energy usage.

It is conjectured that the error is mainly due to the incompleteness and the poor quality of the original data.

4.2.4 Training and Testing – Static Chiller Electric Demand Predictions

It is important to understand how the cooling system in the CANMET Building works before one constructs, trains and uses an ANN to predict the chiller energy usage of the building. It is shown in Figure 25 that the chilled water for the building can be generated by either ice tanks or chillers. When there is sufficient amount of ice left in either ice tanks, chillers are all automatically turned off. The electric energy used by the chiller is zero. Hence it has no relationship with respect to the temperature and other operational variables listed in Table 13. Thus, when there is sufficient ice left in the ice tank, there is no need to predict the chiller electric demand because it is zero. The chiller electric energy demand only needs to be predicted when the amount of ice left in both ice tanks is low and at least one of the compressors is turned on.

Ideally, one would use the amount of ice left in both ice tanks as an indicator to predict whether the chiller will be turned on in the next hour. A separate ANN can be constructed to predict the status of the chiller (on or off) based on the remaining ice level. However, the original data file provided by CANMET does not contain measurements related to the amount of ice in the ice tank. Thus, it is not possible to predict the status of the chiller using measurements collected in previous hours. However, the status of each compressor at time t is given by variables $SCi(t)$, $i=1,2...6$. This information will be used to determine whether the chiller electric demand usage should be predicted in the following experiments. In another words, a prediction of the chiller electric energy usage is made only when one of the compressors is turned on (indicated by $SCi(t) = 1$ for some $i=1,2...6$). Although this approach is not entirely practical, it is the best one can do given the existing data file. As a result, both training and testing are only performed on measurements associated with non-zero $SCi(t)$ values. When $SCi(t) = 0$, for some i, it can be deduced immediately that $C(t) = 0$, and no prediction is needed.

Although the chiller is usually turned on in the summer for cooling, the CANMET data set does not contain any chiller energy measurements corresponding to the summer months of 2002 (from June to August). Some measurements are available for the month of September and October. However, even for these two months, the measurements are far from complete. The incompleteness and rather low volume of the data makes it difficult for the training process to render an accurate ANN prediction model, as is shown next.

4.2.4.1 Experiment 11 - Modeling the nonlinear mapping between the Chiller Electric Demand *E(t)* and other temperature and operational measurements

In this experiment, we use all variables listed in Table 13 except the status variables (*SCi(t)*, $i = 1, 2, \ldots 6$) as the input elements to an ANN. Note that the measurements associated with a nonzero *SCi(t)*, for any $1 \leq i \leq 6$ are removed from the training data set ahead of time. With the limited amount of the data, 80% of the nonzero measurements (the measurement associated with 130 hours), which include the measurement associated with September, 2002 to May, 2003, are reserved for training. The remaining data including zero and nonzero measurements associated with May to July, 2003 is set for prediction. After training the network for a maximum of 500 epochs, the MSE becomes less than 10^{-3}kW. When the trained ANN model is applied to the testing data which contains both zero and nonzero chiller measurements, the ANN output is multiplied by the union of *SCi(t)*, $i = 1, 2, \ldots 6$. Thus, if *SCi(t)* = 0 for $1 \leq i \leq 6$, the predicted chiller energy usage *C(t)* will be exactly zero. Otherwise, the predicted *C(t)* equals what is computed by the ANN. shows that the predicted chiller energy usage matches with the actual chiller energy demand extremely well. This observation is further confirmed in Figure 33 and Figure 34, where the error and error distribution of the ANN prediction are plotted respectively. These figures show that the nonlinear mapping between the chiller energy usage and the environmental and operational variables listed in Table13 can be easily modeled even though the volume of the training data is low. The CV and RMSE values are quite small, they are 0.23 and 3.73kW respectively. Training takes 2.4 seconds only.

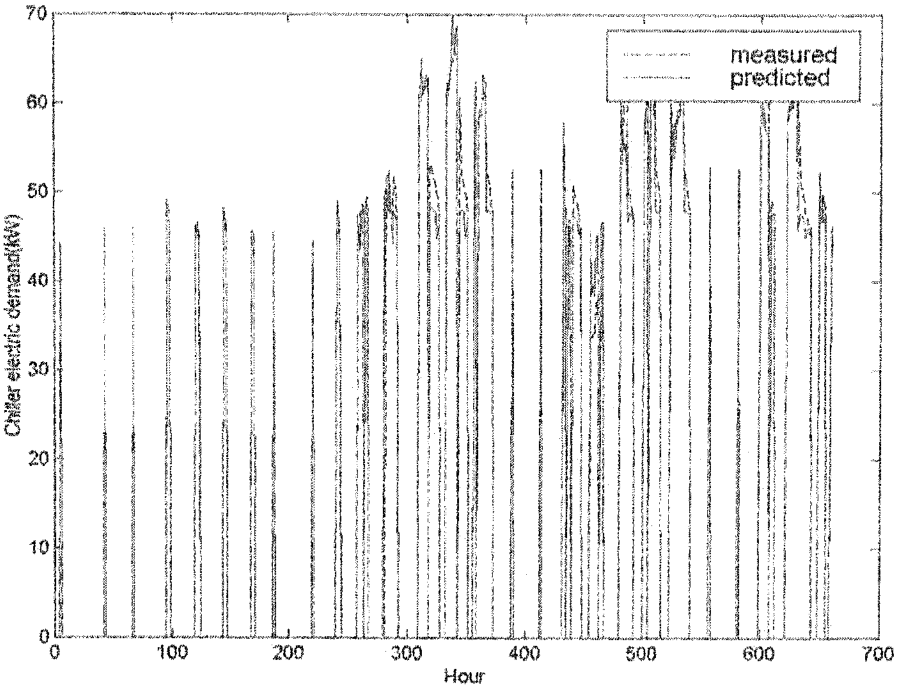


Figure 32. Comparison between the predicted and the actual chiller electric usage.

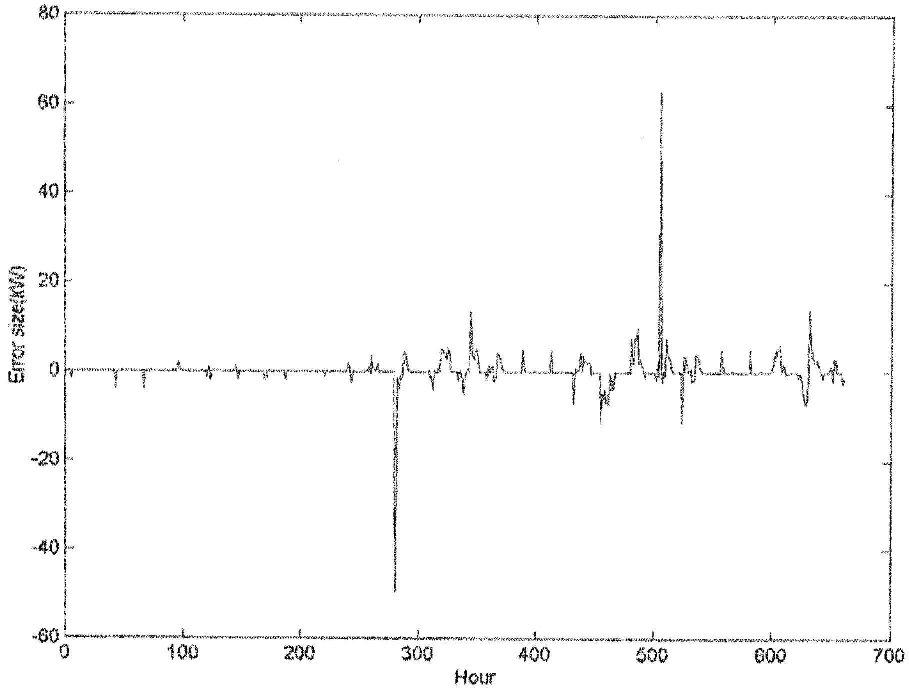


Figure 33. The difference between the ANN predicted and the actual chiller energy usage in experiment 11 of section 4.2.4.1.

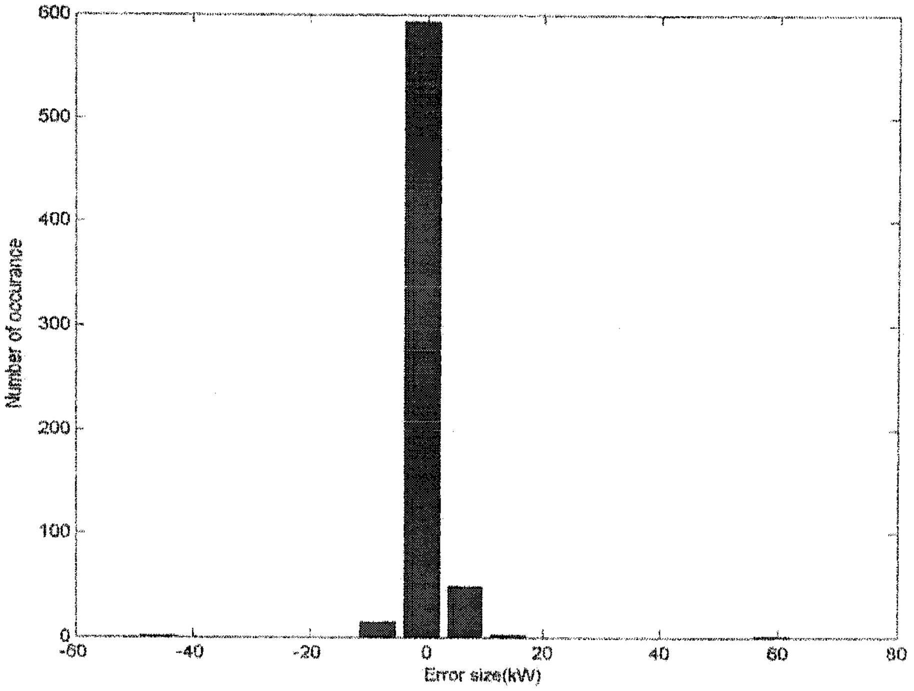


Figure 34. The distribution of the prediction error for Experiment 11 of section 4.2.4.1.

4.2.4.2 Experiment 12 - Chiller electric demand prediction with previous hour measurements

As discussed earlier, even though the chiller energy usage *C(t)* clearly depends on quantities that reflect the environmental conditions and the operational status of the cooling system (listed in Table 13) at time *t*, it is not practical to use the ANN model developed in Section 4.2.4.1 to make chiller energy prediction. This is because the present-time values of the environmental and operational variables are often not available until the end of the hour. To be able to predict future energy usage using present and previous measurements, one must develop an ANN model that takes only previous environmental and operational measurements as input.

In this experiment, an ANN model is developed that predicts *C(t)* based on measurements collected in previous hours. The input variables chosen for this ANN are listed in Table 17.

Table 17. The ANN input variables that consist of environmental and operational measurements collected in previous hours.

Variable	Description
H(t)	Number of hour
Te(t-k), $1 \le k \le 6$	Temperature of the water entering the ice tank
Tev(t-k), $1 \le k \le 6$	Temperature of the water entering the evaporator
Tlv(t-k), $1 \le k \le 6$	Temperature of the water leaving the evaporator
Scw(t-k), $1 \le k \le 6$	The on/off status of cooling water
Smode(t-k), $1 \le k \le 6$	The on/off status of ice mode control
Hum(t-k), $1 \le k \le 6$	Outdoor relative humidity
TOD(t-k), $1 \le k \le 6$	Outdoor temperature
SV1(t-k), $1 \le k \le 6$	Whether to bypass the cooling coil to fabricate ice (yes/no)
SV2(t-k), $1 \le k \le 6$	Whether to mix the returning cold water with the water coming out of the ice tank (yes/no)
CR(t-k), $1 \le k \le 6$	Current used by the chiller in Amp
C(t-k), $1 \le k \le 6$	Chiller energy usage in kW

PCA is used to reduce the dimension of the input and to remove redundancy in the data. Six principal components that contribute to more than 1% of the total variation are retained. This new network is somewhat more difficult to train. The training time, which is 5.7 seconds, is twice longer than that of the first experiment. The MSE of the ANN output at the end of the training process is around 10^{-2}kW. Figure 35. shows some visible discrepancy between the predicted and the actual energy load curve. However, the overall prediction accuracy provided by this ANN is satisfactory. Figure 36 shows that a large number of occurance of the error is within [-5,5] KW, which is quite reasonable. The CV and RMSE values obtained in this experiment are 26% and 4.28 kW respectively.

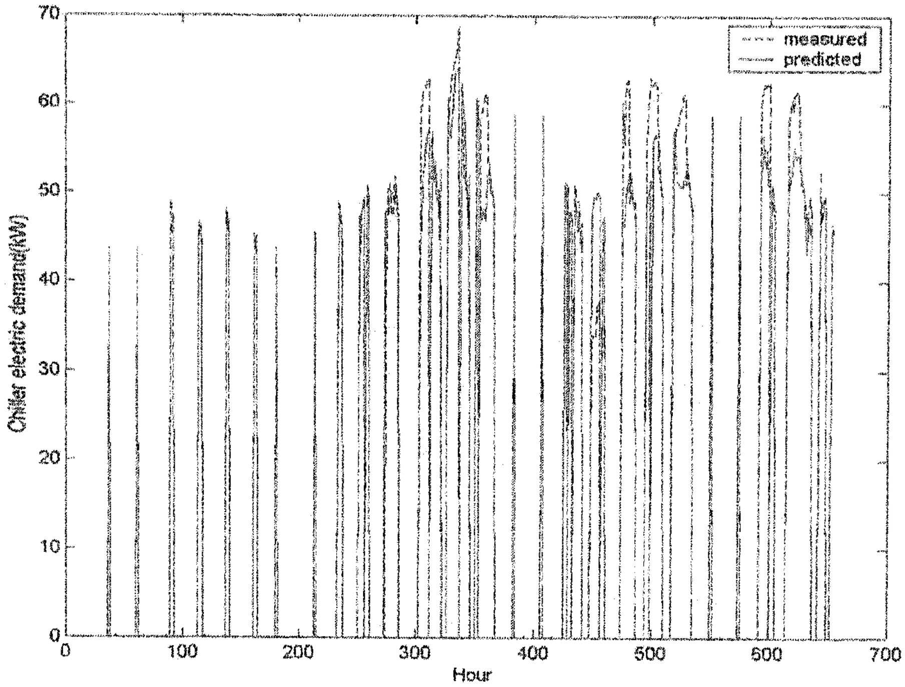


Figure 35. Comparison between the predicted and actual chiller electric energy usage in Experiment 12 of section 4.2.4.2.

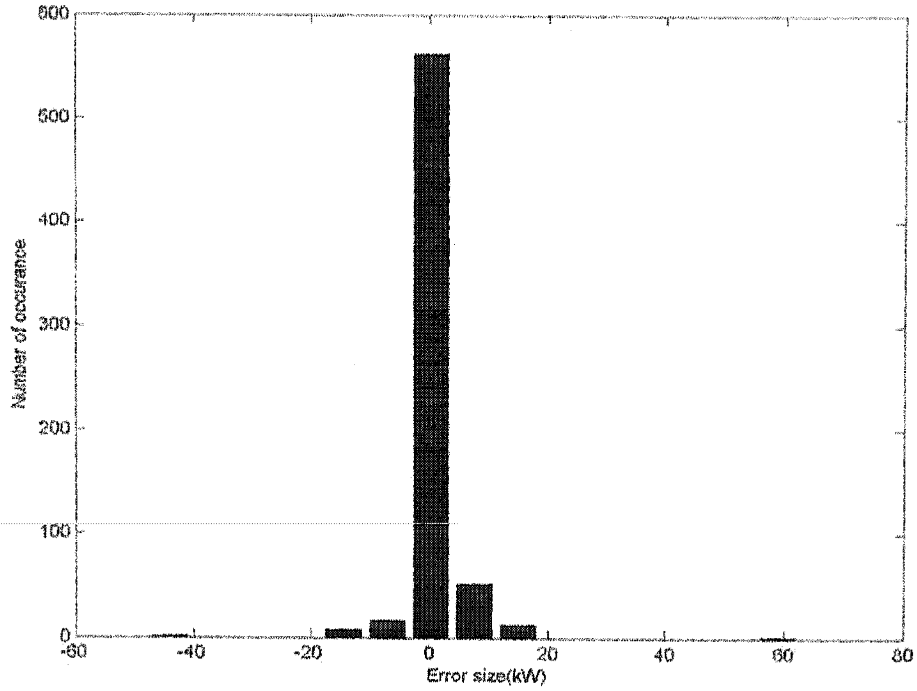


Figure 36. Error distribution detected in Experiment 12 of section 4.2.4.2

Table 18. Comparison of training time for static chiller demand predictions for CANMET Lab

Experiment	Training time (seconds)
11	2.4
12	5.7

4.2.5 Training and Testing – On-line Chiller Electric Demand Predictions

In this section, experiments with both accumulative training and sliding window training are carried out. The input data listed in Table 17 associated with September, 2002 to May, 2003, which include 130-hour nonzero measurement is reserved for training, The data including zero and nonzero measurements associated with May to July, 2003 is set for prediction. Unlike the static predictions conducted in Section 4.2.4, for on-line prediction, it is impractical to use variable SCi(t) to indicate the on-off chiller status at time t because SCi(t) is not available at the time predictions start . Thus it is reasonable to assume the SCi(t-1) is same as SCi(t) and to use SCi(t-1) instead of SCi(t) . Though the accuracy will be sacrificed a little, it is the best one can do if there are no other variables can be used to predict the on-off status at the time t. In the following experiments, on-line predictions using both variables SCi(t-1) and SCi(t) are carried out. The details and results are presented below.

4.2.5.1 Experiment 13 -On-line prediction using SC*i*(t) with accumulative trained ANN

The ANN developed in Section 4.1.4.1 serves as the building block for constructing a dynamic (on-line) prediction model. In this experiment, the chiller related variables measured between September 2002 and May 2003 are set aside for baseline training. Note that the number of hours during which the chiller is turned on is only 130. Thus, the volume of the training data is rather small. Once the baseline training is completed, the initial ANN model is used to predict the chiller electric usage for the next 24 hours. In this accumulatively trained on-line model, the ANN is updated daily by adding measurements that become available on the day chiller energy is to be predicted into the training data set. Note that only the data recorded during the hours at which a chiller is on are added to the training data set. That is, if the chiller is turned off for the next 24 hours, then the training data will remain unchanged.

Figure 37 shows that the predicted chiller energy usage matches well with the actual usage for the first 200 hours of the testing data (the first eight days in June 2003). At the end of the eighth day, the chiller energy usage exhibits a daily increasing pattern. This "unexpected" pattern presents some challenge for the ANN prediction model because the model has not been trained adequately for increased energy demand at that point. However, after newly measured data are added incrementally into the training data set, the accuracy of the prediction gradually improves. Figure 38 shows that a large number of the error occurrences are within [-5,5] kW. The small percentage of relatively large errors (around 40 to 50 kW) is associated with the mis-prediction for days 9-12 when the chiller energy usage undergoes a significant daily increase. The overall CV and RMSE values obtained in this experiment are 38% and 2.56kW respectively.

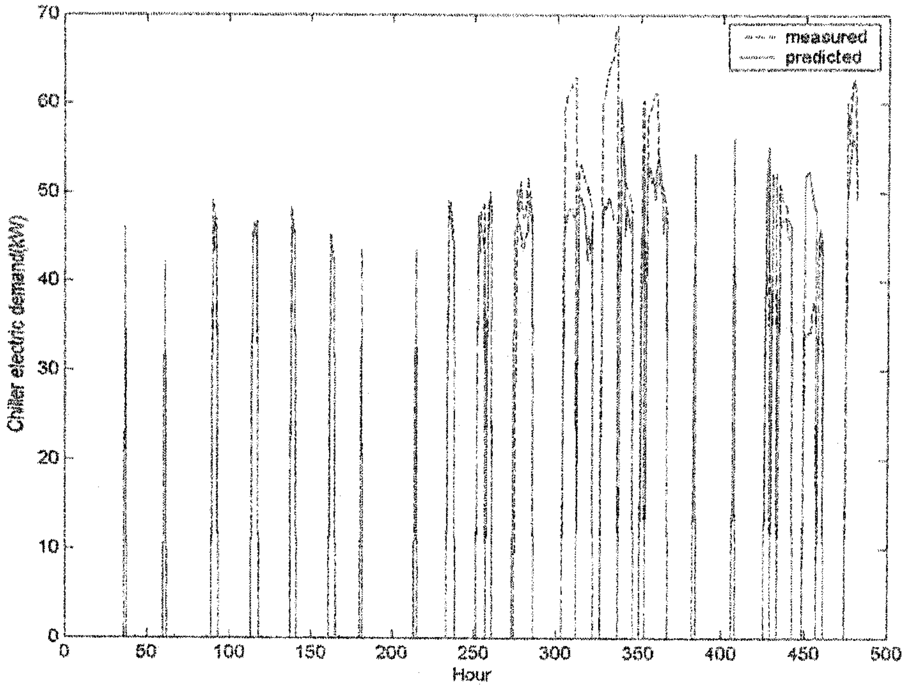


Figure 37. Comparison between the actual chiller energy usage curve and the one predicted by the accumulatively trained ANN developed in Experiment 13 in Section 4.2.4.3.

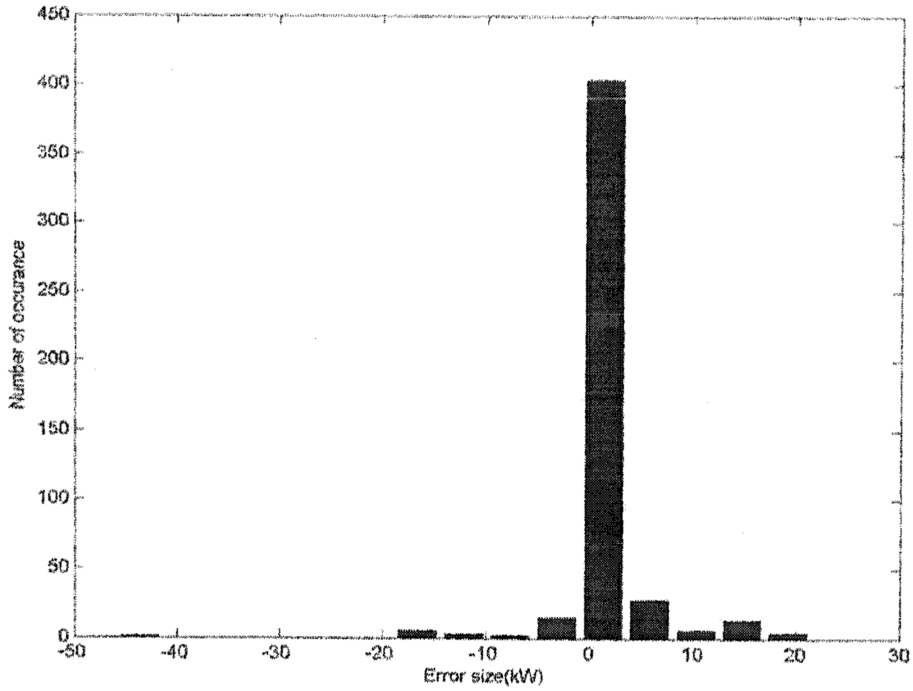


Figure 38. Error distribution in Experiment 13 of Section 4.2.4.3.

4.2.5.2 Experiment 14 – On-line prediction using SC*i*(t-1) with accumulative trained ANN

It is described in Section 4.2.4 that variable SCi(t) has a direct impact on the on/off status of the chiller. Though good performance is obtained in Experiment 13 by using SCi(t) to determine whether the chiller electric demand should be predicted, this is impractical because SCi(t) is not available at the time prediction is to be made. In this experiment, the on/off status of the chiller at the previous hour, SCi(t-1) is used. The rationale for using SCi(t-1) is simple. The chiller tends to be turned on or off for a period of time. Thus, if SCi(t-1) is 1 (or 0), then one would expect that SCi(t) is likely to assume the value of 1 (or 0) also. In this case, SCi(t-1) is a reasonable choice. The results of the experiment using SCi(t-1) will be compared to the one obtained in Experiment 13.

Figure 39 shows that the predicted energy usage matches the actual measurement fairly well at most time spots, but visible discrepancy is still observed between the ANN output and the true measurement at hours 300 and 500. It can be seen from Figure 40 that the distribution of error has a wider range than the one observed in Experiment 13, its range is [-50, 50] kW. The CV and RMSE values also become larger, they are 253% and 13.29kW respectively.

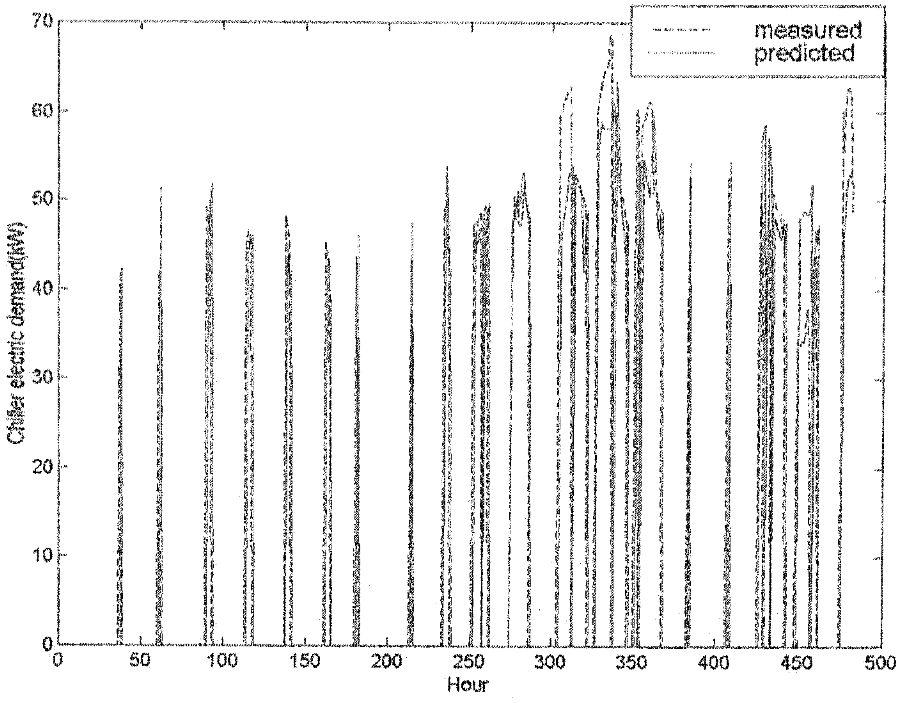


Figure 39. Comparison between the actual chiller demand and the one predicted by the accumulative trained ANN developed in Experiment 14 in Section 4.2.4.3

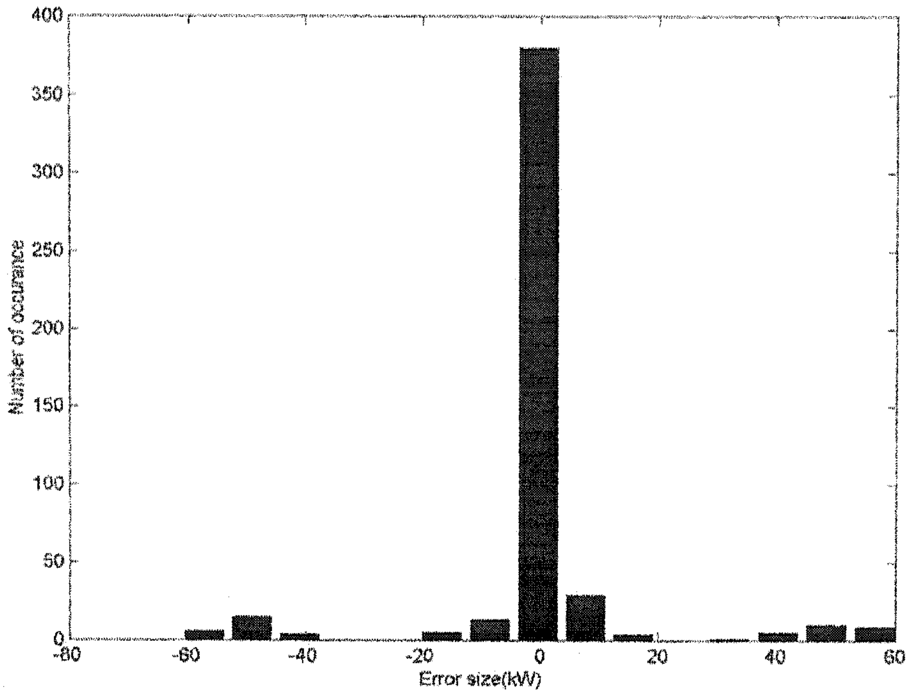


Figure 40. Error distribution in Experiment 14 of Section 4.2.4.3

4.2.5.3 Experiment 15 -On-line prediction using SC*i*(t) with sliding window training

It is advantageous to use a sliding window to manage the training process of a dynamic (online) ANN prediction model when the volume of training data becomes exceedingly large. In such a case, a time window with a fixed size is used to maintain a fixed volume of training data. As new data becomes available, old data is deleted from the training window. The sliding window has to be large enough to contain sufficient amount of data so that the training process can identify a time-dependent nonlinear mapping between the input and output. In the meantime, the window size has to be small enough to allow the training process to be completed quickly.

The data set provided by CANMET center has such a small volume that it is difficult to develop a dynamic model based on sliding window training. Just like Experiment 13, the chiller related variables measured between September 2002 and May 2003 are set aside for baseline training. Note that the number of hours during which the chiller is turned on is only 130. Once the baseline training is completed, we use the initial ANN model to predict the chiller electric usage for the next 24 hours. The ANN is updated daily by adding new measurements into the training data set and deleting some previous measurements from the training set. Note that only the data recorded during the hours during which a chiller is on are added to the training data set. The amount of data deleted from the training set equals the amount of new data added. This is to ensure the volume of the training data is constant. Network training and retraining only take 30 seconds for the whole experiement, which is 40% faster than that of accumulative training. Figure 41 shows that the limited size of the training data contributed to relatively large errors observed when the ANN is applied to the testing data. But the overall CV and RMSE values appears to be smaller compared to the result obtained from Experiment 13 using accumulative training in Section 4.2.4.3, they are only 9% and 4.39kW respectively

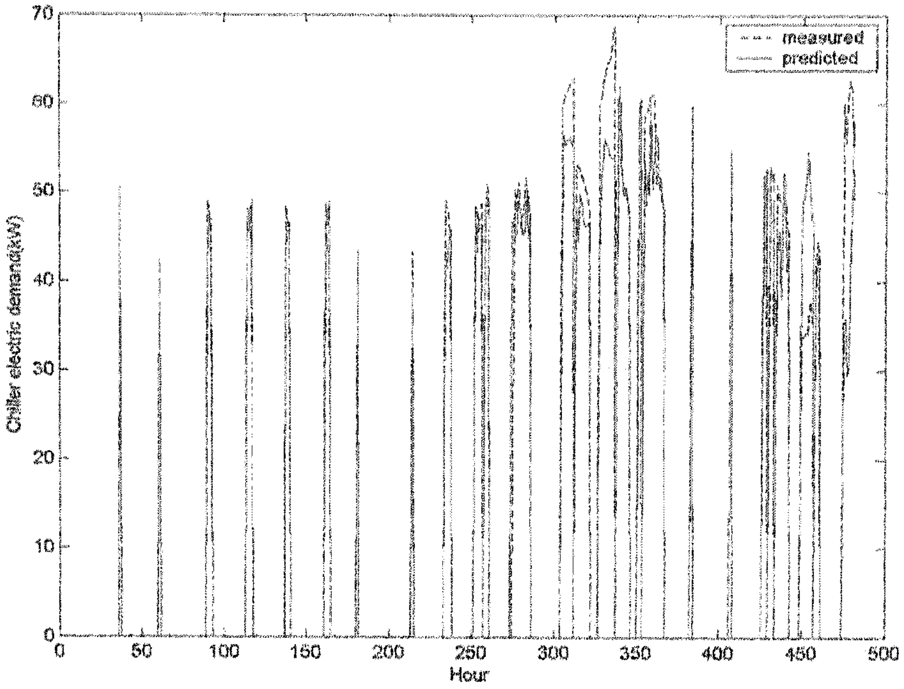


Figure 41. Comparison between the actual chiller energy usage curve and the one predicted by the sliding-window on-line ANN model developed in Experiment 15 of Section 4.2.4.4.

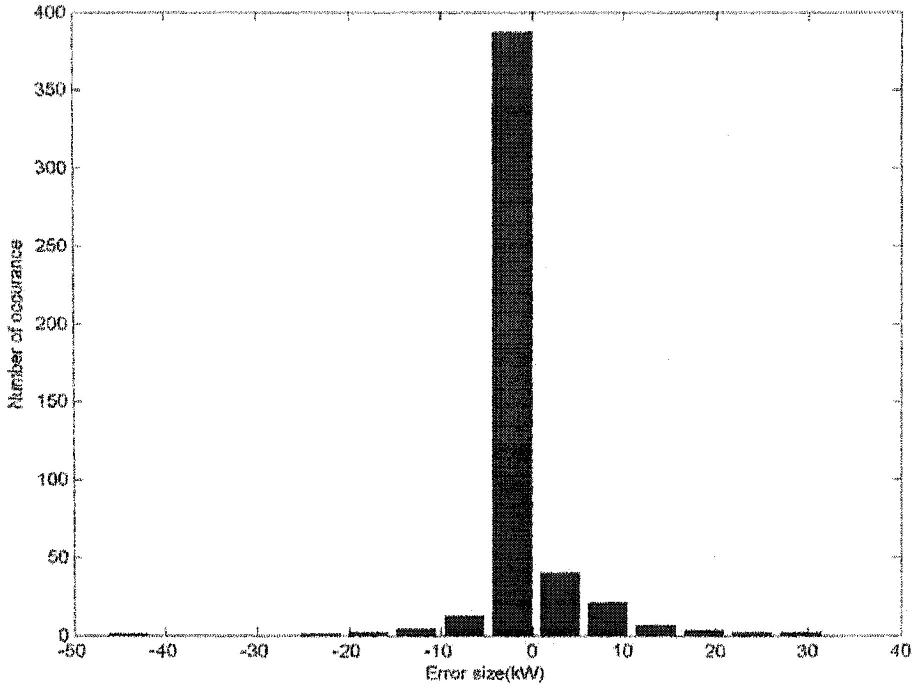


Figure 42. Distribution of the error detected in Experiment 15.

4.2.5.4 Experiment 16 – On-line prediction using SC*i*(t-1) with sliding window training

Variable SC*i*(t-1) is used to decide if the electric demand is needed to be predicted in this experiment. The prediction result is shown in Figure 43. Figure 43 shows that the predicted energy usage matches the measurement fairly well at most time spots. But compared to Experiment 15 presented in this section, the difference between the predicted energy usage and measurement becomes larger. The error ranges from –60 to 60 kW. The CV and RMSE values are also larger than the ones observed in Experiment 15. They are 26% and 12.88kW respectively.

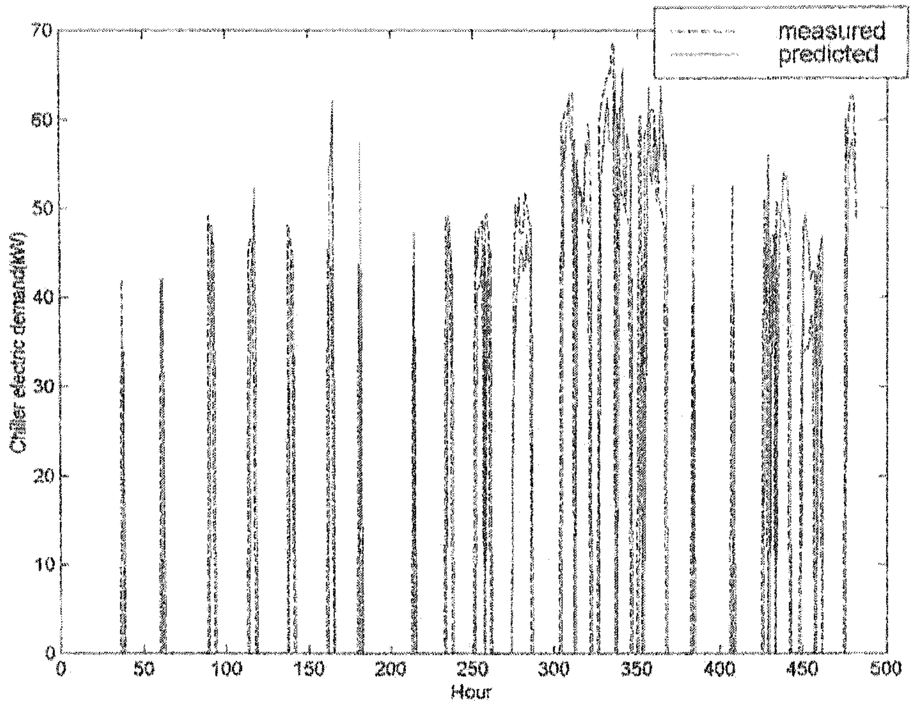


Figure 43. Comparison between the actual chiller energy usage curve and the one predicted by the sliding window on-line ANN model developed in Experiment 16 of Section 4.2.4.4

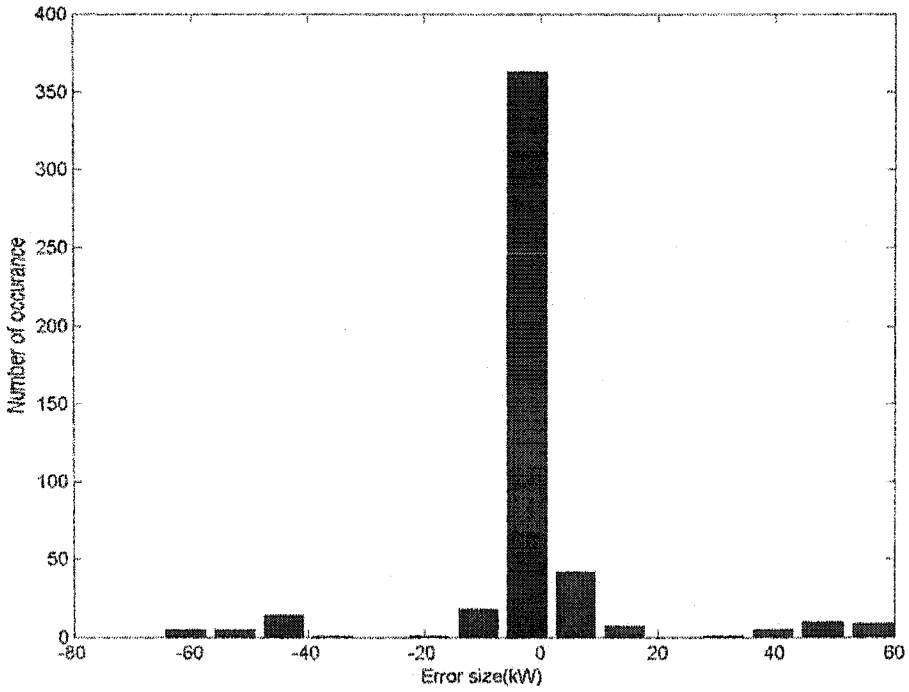


Figure 44. Error distribution in Experiment 16 of Section 4.2.4.4

Table 19. Comparison of the training time for on-line chiller demand predictions for CANMET

Experiment	CV (%)	RMSE (kW)	Training time (seconds)
13	38	2.56	50.4
15	9	4.39	30.9

4.2.6 Summary

Unlike the experiments conducted on simulated data in section 4.1, the experiments carried out in this section involve additional work in terms of data processing. The data set prepared by CANMET are recorded in a Microsoft Access (MA) file which cannot be used directly by the MATLAB ANN code developed in this thesis, additional programs must be written to convert the MA file into a matrix data format that can be easily handled by MATLAB. This appears to be a typical problem in computational energy prediction. The lack of standards in energy data makes it difficult to automate the prediction process. Another difficulty associated with this data set is the completeness and accuracy of the measurements. It is found that there is a massive amount of missing measurements. Some of the measurements are apparently out of the physical range of the quantity measured. The lack of high quality data also makes it difficult to perform experiments and draw sensible conclusions from the experiments.

Attempts are made to predict both the gas energy demand and the chiller electric energy use in the building. Because the boiler and chiller are controlled by two separate systems, the predictions of these quantities are treated separately. That is, one ANN is built to predict the gas energy demand while a different one is use to predict chiller energy demand.

Because the gas demand fluctuates among three levels, conventional ANN prediction models such as the ones developed in Section 4.1 for the Laval building do not work well. Limited success is achieved using a perceptron network developed to classify the measurements of environmental and operational variables into a few clusters.

A higher level of success is achieved with respect to the chiller energy prediction. The key to a successful prediction is to identify the on-off status of the chiller. When the ice-tanks of the building are non-empty, the chiller is completely turned off. Hence the chiller energy demand is zero. No prediction needs to be made. The ANN prediction model is used only when the chiller is turned on. The on/off status of the chiller can, in principle, be predicted by looking at the ice-tank level. However, such type of information is not present in the CANMET data set. No variable can be found to indicate on-off status of the chiller in the next hour. Thus, the best one can do is to use the previous on/off status of the chiller to predict the future on/off status of the chiller. The problem is when the chiller turns off, the system will miss it by one hour since it assumes it is ON based on the previous hour. Only in the following hour will the system finally see it as OFF and then work properly. The prediction of the ON/OFF is always lagged by one hour. Both accumulative and sliding window training produced reasonably satisfactory result. However, due to the limited amount of data, it is difficult to make a fair comparison between these two training schemes other than the fact that the sliding window training scheme is more efficient in terms of CPU time use.

5. CONCLUSION

5.1 Summary

A wide spectrum of computational models for building energy prediction has been examined in this thesis. A comprehensive survey is presented to summarize the pros and cons of each method. The survey presented both the general design philosophy for each prediction model and constructive discussions on the implementation details and the performance using these models. It is evident from the surveyed literature that the artificial neural network (ANN) prediction model is the most attractive model among all models. Its main advantage lies in its flexibility in modeling any nonlinear mapping between a set of independent variables and dependent variables.

A detailed discussion on the theory and practice of ANN models is presented in this thesis. While most of the surveyed literature focused on using a static ANN model to predict energy usage during a fixed period of time, this thesis investigated dynamic ANN models that evolve over time. A dynamic model has the apparent advantage of adapting itself to seasonal or year-to-year change in energy demand patterns. A dynamic model can be combined with an automated data acquisition system to provide online building energy prediction.

A number of computational experiments have been performed in this thesis to demonstrate the effectiveness of a dynamic ANN model. The experiments were carried out on two data sets. The first data set contains simulated data. The second data set contains measurements recorded by sensors installed in the building investigated. It is found that the nonlinear relationship between the environmental/operational measurements and the building energy demand is relatively easy to model by an ANN. This instantaneous mapping is pursued in most of the surveyed literature. However, because the prediction of the energy demand must be made using environmental/operational measurements obtained in previous hours, what should be modeled is a weaker but still nonlinear mapping between time-lagged measurements of the environmental/operational variables and the present building energy demand. To accurately model the impact of the change in environmental/operational variables to building energy demand, it is desirable to include as many previous measurements as possible in the ANN input. However, this could impose a severe computational burden to the ANN training process. It is shown in this thesis that the use of PCA can alleviate this problem by reducing the dimension of the input and removing the redundancy in the data.

Two types of adaptive ANN training schemes have been experimented in this thesis. The sliding window training approach, which keeps the volume of the training data set at a constant level appears to be slightly better than the accumulative training scheme, which simply adds newly available data into the training data set. This is observed in experiments that involve the simulated data (for the Laval Building) and the real data (for the CANMET building).

5.2 Contributions

The most up-to-date literature survey is presented in the thesis. Several different prediction models are presented and summarized in detail. The advantages and disadvantages of these models are analyzed and compared to provide comprehensive overview on methodology for building energy prediction problems.

Dynamic ANN models for energy prediction problems have been examined in detail while most researchers only focus on static model. The models are proved to be effective by experiments performed using both simulated data and real measurements.

While most of the surveyed literature focus only on using ANN to model instantaneous mapping between the environmental/operational variables and building energy demand for static prediction models, this thesis used ANN to dynamically model the nonlinear mapping between time-lagged measurements and the present building energy demand. This method is more practical because of the difficulty that the instantaneous measurements will not be available until the end of the hour. As many previous measurements as possible are included as ANN input variables.

Principle Component Analysis (PCA) is used in the thesis to reduce the input dimension and remove the redundancy in the data.

Two ways to update the dynamic ANN model have been presented in this thesis. The updating is accomplished through retraining the network periodically. Two approaches for network retraining, accumulative training and sliding window training are developed. The experiments shown in this thesis demonstrated the effectiveness of these two online training schemes.

5.3 Future Research Direction

It is conceivable that a dynamic ANN trained with a sliding window can outperform an accumulatively trained ANN both in accuracy and efficiency. However, a fair comparison requires a sufficiently large volume of data, which is not available to this author at this time. When a large volume data becomes available, an optimal window size needs to be determined. Additional research is needed to investigate how to determine an optimal window size.

It is suggested in this thesis that the prediction of gas energy usage requires the use of an ANN that is capable of classifying the environmental and operational data into several discrete groups. A simple perceptron is developed in this thesis with limited success. Additional research is required to develop a more sophisticated classification network to improve the accuracy of gas energy prediction.

Another future research direction is to combine the ANN prediction models developed in this thesis with an automated data acquisition software package to produce a truly on-line building energy prediction system. To make this system portable to a larger number of commercial buildings, one must also standardize the format in which the measured energy data is stored.

Energy predictions for holidays are not fully addressed in this thesis. In the experiments for Laval building, energy demand at non-working time is not predicted because energy demand at this time is zero. For the energy prediction of CANMET Lab, only one ANN is used to predict the energy for both holidays and working days. In the future, two separate ANN models can be used to make energy predictions for holidays and working days respectively.

Table 20. CV and RMSE values obtained from experiments for Laval Building

Experiments for Laval Building	**CV (%)**	**RMSE (kW)**
Static prediction		
1-Modeling the non-linear mapping between E(t) and the temperatures	4%	6.10
2-Using time-lagged measurements as inputs	16%	25.46
3-Using additional time-lagged measurements as inputs and PCA to reduce the dimension of the input	7%	11.01
On-line prediction		
4-Accumulative training with time-lagged temperature measurement as input	15%	28.26
5-Accumulative training with time-lagged chiller energy usage as inputs	17%	28.92
6-Sliding window training using temperature data collected in previous hours	15%	27.73
7-Sliding window training using temperature and energy measured in previous hours	16%	27.78

Table 21. CV and RMSE values obtained from experiments for CANMET Center

Experiments for CANMET Center	**CV (%)**	**RMSE (kW)**
Static prediction		
Gas demand prediction		
8-Modeling the non-linear mapping between G(t) and other variables	62	134.2
9-Gas energy prediction using measurements recorded in previous hours	53	111.6
10-Classification of the input	50	121.8
Chiller electric demand		
11-Modeling the nonlinear mapping between the chiller energy usage E(t) and other temperature and operational measurements	23	3.73
12-Chiller energy prediction with previous hour measurements	26	4.28
On-line chiller demand prediction		
13 - Prediction using SCi(t) using accumulative trained ANN	38	7.56
14- Prediction using SCi(t-1) using accumulative trained ANN	253	13.29
15 -Prediction using SCi(t) using sliding window training	9	4.39
16- Prediction using SCi(t-1) using sliding window training	26	12.88

REFERENCES

Alves. A.P. and Moulin L., 2000 "Confidence Intervals for Neural Network Based Short-Term Load Forecasting", IEEE Transaction on Power System, Vol.15, No.4, pp. 1191-1196.

Anstett, M. and Kreider J.F., 1993 "Application Of Neural Networking Models To Predict Energy Use" *ASHERE Transaction*, Vol. 99, pp. 505-517.

Bakirtzis,A.G.,Theochairs,J.B, Kiartzis,S.J., Satsios,K.J., 1995, "Short-Term Load Forecasting Using Fuzzy Neural Networks"", IEEE Transaction on Power Systems, Vol. 10, No.3, pp.1518-1524.

Boonyatikarn, S., 1982 "Impact of Building Envelope on Energy Demand and Energy Design Guidelines." *Proceedings of the ASHERE/DOE Conference on Thermal Performance of the Exterior Envelope of Buildings II*, pp. 469.

BreekWeg M.R.B., Gruber P. and Ahmed O., 2000a "Development of a Generalized Neural Network Model to Detect Faults in Building Energy Performance-Part I" ASHERE Transaction, Vol.106, Pt2, pp.61-73.

BreekWeg M.R.B., Gruber P. and Ahmed O., 2000b "Development of a Generalized Neural Network Model to Detect Faults in Building Energy Performance-Part II" ASHERE Transaction, Vol.106, Pt2, pp.74-93.

Charytoniuk W and Chen M.S, 2000 " Very Short- Term Load Forecasting Using Artificial Neural Networks", IEEE Transactions on Power Systems, Vol. 15, No.1 pp.263-268.

Curtiss, P.S., Kreider, J.F. and Brandemuehl, M.J., 1993 "Adaptive Control of HVAC Processing Using Predictive Neural Networks," *ASHERE Transactions*, Vol.99, Pt.1, pp.496-504.

Curtiss, P.S., Kreider, J.F. and Brandemuehl, M.J., 1994 "Energy Management in Central HVAC Plants Using Neural Networks," *ASHERE Transactions*, Vol.100 Pt.1, pp.476-493.

Curtiss, P.S., Dodier, R., Claridge, D.E., and Kreider, J.F., 1995 "Building Energy Use Prediction and System Identification Using Recurrent Neural Networks". *Journal of Solar Energy Engineering*, Vol.117, pp.161-166.

Curtiss, P.S., Shavit, G. and Kreider, J.F. 1996 "Neural Networks Applied to Buildings-A Tutorial and Case Studies in Prediction and Adaptive Control ", *ASHERE* Transaction, Vol. 102, Pt.1, pp.1141-1146.

Curtiss, P.S., Massie, D.D. and Kreider, J.F., 1998a "Predicting Central Plant HVAC Equipment Performance Using Neural Networks- Laboratory System Test Results". *ASHERE Transactions,* Vol.104, Pt1B, pp.221-228.

Curtiss, P.S., Kreider, J.F., Assawamartunlue, K. and Jeannette.,E., 1998b "Experimental Results of a Predictive Neural Networks HVAC Controller". *ASHERE Transactions*, Vol.104, Pt2, pp.192-197.

Dhar, A., Reddy, T.A., and Claridge D. E., 1998 "Modeling Hourly Energy Use in Commercial Buildings With Fourier Series Functional Form". *ASME Journal of Solar Energy Engineering,* Vol. 120, No.3, pp. 217-223.

Dhar, A., Reddy, T.A., and Claridge D.E., 1999a "A Fourier Series Model to Predict Hourly Heating and Cooling Energy Use in Commercial Buildings with Outdoor Temperature as the Only Weather Variable". *ASME Journal of Solar Energy Engineering*, Vol 121, No.1, pp.47-53.

Dhar, A., Reddy, T.A., and Claridge D. E., 1999b "Generalization of the Fourier Series Approach to Model Hourly Energy Use in Commercial Buildings". *ASME Journal of Solar Energy Engineering,* No.1, Vol.121, pp.54-62.

Djukanovic M. R. S, Babic B., Sobajic D.J., and Pao Y-H 1995, "a neural-based short term load forecasting using moving window procedure" Electric Power &Energy System Vol. 17, No.6, pp.391-397.

Drezga I. and Rahman S., 1999 "SHORT-TERM LOAD FORECASTING WITH LOCAL ANN PREDICTORS", IEEE Transactions on Power Systyems, Vol.14, No.3, pp.844-850.

Dodier R. 1995, "Statistical Properties of Neural Networks, with Application to Building Energy Prediction" Thesis.Department of Civil, Environmental and Architectural Engineering, Portland University

Fausett, L., 1994 "Fundamentals of Neural Networks: Architecture, Algorithms, and Applications", Prentice Hall, Englewood Cliffs.

Fels M., 1986 "Special Issues Devoted to Measuring Energy Savings: The Scorekeeping Approach," *Energy and Buildings*, Vol. 9, Nos 1 and 2.

Feuston, B., and Thurtell, J., 1994 "Generalized Non-linear Regression with Ensemble of Neural Nets. *ASHERE Transaction.*, Vol.100 (2)

Fletcher R. "Practical Methods of Optimization" Chichester NY 1987

Haberl J.S. and Thamilseran S., 1996 " The Great Energy Predictor Shootout II-: Measuring Retrofit Savings-Overview and Discussion of Results" *ASHERE Transactions*, Vol.102, pp.419-435.

Harvey, A.C., 1992 " Time Series Models" Philip Allan Publishers Limited

Hetcht-Nielson, R, 1989 " Theory of the backpropagation neural network," *Proc. IEEE Int'l Conf. on Neural Networks,* Vol.1, pp.593-605.

Hong-Tzer Yang, Chao- Ming Huang and Ching-Lien Huang, 1996 " Identification of ARMAX model for Short Term Load Forecasting: An Evolutionary Programming Approach" *IEEE Transaction on Power Systems*, Vol.11, No.1, pp.403-408.

Yoo H. and Pimmel R. L., 2000 "Short Term Load Forecasting Using a Self-Supervised Adaptive Neural Network" *IEEE Transactions on Power System*, Vol.14, No.3, pp.779-784.

Johnson R. A. and D. W. Wichern, 2002. "Applied Multivariate Statistical Analysis", 5th edition, Prentice Hall, NJ, 2002. page 426 - 476.

Katipamula, S., Reddy, T.A. and Claridge D. E., 1998 "Multivariate Regression Modeling". ASME *Journal of Solar Energy Engineering*, Vol. 120, pp.177-184.

Kawashima, M., Dorgan., C. E., and Mitchell., J. 1994 "Artificial Neural Network Backpropagation Model With Three-Phase Annealing Developed For the Building Energy Predictor Shootout " *ASHERE Transaction,* Vol.100, Pt2, pp.1096-1103.

Kawashima, M., Dorgan., C. E., and Mitchell., J. 1995 "Hourly Thermal Load Prediction For The Next 24 Hours By ARIMA, EWMA, LR, and an Artificial Neural Network" *ASHERE Transaction* Vol.101, Pt1, pp.186-200.

Khotanzad, A., Hwang,R.C., Abaye, Maratukulam,A.D.,1995 "An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities" IEEE Transactions on Power System, Vol. 10, No.3 August, pp.1716- 1722

Kim, K., Youn, H. and Yong C.K., 2000. "Short-Term Load Forecasting for Special Days in Anomalous Load Conditions Using Neural Networks and Fuzzy Inference Method". *IEEE Transactions on Power Systems*. Vol. 15, No. 2, pp. 559-565.

Kimbara, A., Kurosu, S., Endo, R., Kamimura, K., Matsuba, T. and Yamada, A, 1995 " On-line Prediction for Load Profile of an Air-Conditioning System". *ASHERE Transaction*, Vol. 101, Pt.2, pp.198-207.

Kissock, J.K., Reddy, T.A. and Claridge D. E., 1992 "Measuring Retrofit Savings for the Texas Loans STAR Program: Preliminary Methodology and Results". *Proceedings of the ASME/JSES/KSES/International Solar Energy Conference*, Maui, Hawaii, pp.299-308.

Kissock, J.K., Reddy, T.A., Fletcher, D., and Claridge D. E., 1993 "The Effect of Short data Periods on The Annual Prediction Accuracy of Temperatures-Dependent Regression Models of Commercial Buildings Energy Use". Proceedings of the 1993

ASME/JSES/SED *International Solar Energy Conference*, Washington, D.C., April, pp.455-463.

Kissock, J.K., Reddy, T.A. and Claridge D. E., 1998 "Ambient-Temperature Regression Analysis for Estimating Retrofit Savings in Commercial Buildings". *ASME Journal of Solar Energy Engineering*, No.3, Vol. 120, pp.168-176.

Kreider, J.F. and Haberl, J.S., 1994 "Predicting Hourly Building Energy Use- The Great Energy Predictor Shootout- Overview and Discussion of Results". *ASHERE Transaction*, Vol.100, Pt2. pp.1104-1118.

Kreider, J.F., Claridge D.E., Curtiss., P, Dodier, R. and Krarti, M., 1995 "Building Energy Use Prediction and System Identification Using Recurrent Neural Networks". *ASME Journal of Solar Energy Engineering*, Vol 117, pp.161-166.

Lee K.Y., Cha Y.T. and Park J.H., 1992 "Short-Term Load Forecasting Using Artificial Neural Networks", IEEE Transaction on Power Systems, Vol. 7, No.1, pp.124-131.

Liu, K, Subbarayan.S and Shoult, R.R., Manry, M.T., KWan, C., Lewis, F.L. and Naccarino, J, 1996 "Comparison of Very Short-Term Load Forecasting Techniques" *IEEE Transactions on Power System*, Vol. 11 No.2, pp.877-882.

Lu,C.N., Wu,H.T., and Vemuri S., 1993 "Neural Network Based Short Term Load Forecasting", IEEE Transaction on Power Systems, Vol. 8, No.1, pp.336-342.

MacArthur, J.W., Mathur, A. and Zhao. J. 1989 " On-Line Recursive Estimation for Load Profile Prediction" *ASHERE Transaction*, Vol. 95, Pt.1, pp.621-628.

Mackey, D. J. 1992 "A Practical Bayesian Framework for Backpropagation Networks."Neural Computation , Vol.4, pp. 448-472.

Mackey, D. J. 1994 "Baysesian Nonlinear Modeling for the Energy Prediction Competition." *ASHERE Transactions*, Vol.100, Pt 2, pp. 1053-1062.

Mohammed, O., Park, D., Merchant,R., Dinh T., Tong C., Azeem A, Farah, J., Drake,C.,1995, "Practice experience with an adaptive neural networks short-term load forecasting system", IEEE Transaction on Power Systems, Vol. 10, No.1, pp.254-265.

Mori, H. and Kobayashi, H., 1996 "Optimal Fuzzy Inference for Short-Term Load Forecasting". *IEEE Transactions on Power Systems*, Vol.11, No.1, pp.390-396.

Ohlsson. M, Peterson C., Pi, H., Rognvaldsson T.,1994 "Predicting System loads with Artificial Neural Networks." *ASHERE Transactions*, Vol.100, Pt 2.

Ott. L, 1988 "An Introduction to Statistical Methods and Data Analysis." PWS-KENT Publ. Co., Boston.

Piras, A.B., Buchenel, Y. Jaccard, A. Germond. K. Imhof, 1996 "Heterogeneous Artificial Neural Network for Short Term Electrical Load Forecasting", *IEEE Transaction on Power Systems,* Vol. 11, No.1, pp.397-402.

Ruch, D. and Claridge D. E., 1992 "A Four – Parameter Change-Point Model for Predicting Energy Demand in Commercial Buildings" ASME *Journal of Solar Energy Engineering*, No.2, Vol.114, pp.77-83.

Schalkoff, R., 1997 "Artificial Neural Networks" MIT Press and McGraw-Hill Co., Inc.

Seem, J.E. and Braun, J.E.1991 "Adaptive Methods for Real-Time Forecasting of Building Electrical Demand" *ASHERE Transaction*, Vol.97, Pt1, pp.710-721.

Sforna, M. and Proverbio F 1995 " A neural network operateor short-term and on-line load forecasting environment" Electric Power System Research . Vol. 33 pp139-149.

Sullivan, R.T. and Nozaki, S.A., 1984 "Multiple Regression Techniques Applied to Fenestration Effects on Commercial Building Energy Performance," *ASHERE Transactions*, Vol.90, Part 1A, pp.116-123

Tobi, T., and Hanafusa, T. 1991. "A Practical Application of Fuzzy Control for an Air-Conditioning System" *International Journal of Approximate Reasoning*, Vol. 5, No. 3, pp.331-348.

Wasserman, P.D. 1989 "Neural Computing Theory and Practice" Van Nostrand Reinhold Co., New York

Wald, A. 1943 "Test of statistical hypotheses concerning several parameters when the number of observations is large" *Tran. Am. Math. Soc.* Vol. 54. pp.426-482

Yoo H. and Pimmel R.L ., 1999 "Short-Term Load Forecasting Using a Self-Supervised Neural Network ".IEEE Transactions on Power Systems, Vol.14, No.2, pp.779-784.

APPENDIX A-MAIN CRITERIA FOR EVALUATION OF ENERGY PREDICTION PERFORMANCE

A number of criteria are used in the energy prediction literature for evaluating the accuracy of energy prediction. They include coefficient of variation (CV), robust coefficient of variation (RCV), mean bias error (MBE), root mean square error (RMSE), coefficient of determination (R^2), expected error percentage (EEP) and mean absolute percentage error (MAPE)· Following variables are used in these criteria:

$y_{data,t}$ represents measured data at time t or data value of the dependent variable corresponding to particular set of the independent variables,

$y_{pred,t}$ - represents predicted data at time t,

$y_{data,max}$ - represents maximum measured data,

$\bar{y}_{data}$ - represents mean value of the measured data, and

n - is the number of data points in the data set, some times in regression models n can be replaced by n-m where

m – is the total number of regression parameters in the model

These criteria are defined as follows.

1 Coefficient of Variation (CV):

$$CV = \frac{\sqrt{\frac{\sum_{t=1}^{n}(y_{pred,t} - y_{data,t})^2}{n}}}{\left|\bar{y}_{data}\right|},$$

2 Robust Coefficient of Variation (RCV):

$$\mathrm{RCV} = \frac{\sqrt{\dfrac{\sum_{t=1}^{n}(y_{pred,t} - y_{data,t})^2 \times 90\%best}{n}}}{y_{data,95\%} - y_{data,5\%}},$$

3 Mean Bias Errors (MBE):

$$\mathrm{MBE} = \frac{\dfrac{\sum_{t=1}^{n}(y_{pred,t} - y_{data,t})}{n}}{\left|\bar{y}_{data}\right|},$$

4 Root Mean Square Errors (RMSE) or Standard Deviation (Std):

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(y_{pred,t} - y_{data,t})^2}{n}},$$

5 Coefficient of Determination (R^2):

$$\mathrm{R}^2 = \left(1 - \frac{\sum_{t=1}^{n}(y_{pred,t} - y_{data,t})^2}{\sum_{t=1}^{n}(y_{pred,t})^2}\right),$$

6 Expected Error Percentage (EEP):

$$\mathrm{EEP} = \frac{\sqrt{\dfrac{\sum_{t=1}^{n}(y_{pred,t} - y_{data,t})^2}{n}}}{\left|y_{data,\max}\right|},$$

7 Mean Absolute Percentage Errors (MAPE)

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \frac{\left|y_{data,i} - y_{pred,i}\right|}{y_{data,i}},$$

APPENDIX B - PART OF MATLAB CODES SELECTED FROM THE EXPERIMENTS

1.Code for Experiment 1- Modeling the nonlinear mapping between Electric Demand *E*(t) and the temperatures

```
%% static electric demand prediction without time delayed inputs%%

clear;
format short e;
load delay0.mtx
[nrows,ncols] = size(delay0);
%
% === reorganize the data, seperate the input from the output ==
ndays    = 84;
ntrndys = ndays -5*4;
nhrspdy = 12;

ntrnhrs = ntrndys*nhrspdy;

% input variables H(t), Td(t), Tw(t), Tl(t)

matin  = delay0(1:ntrnhrs,1:4);

%number of columns
ninput = size(matin,2);
%
% output
%
matout = delay0(1:ntrnhrs,5);
%
% normalize the data
%
[dinn, minp, maxp, doutn, mint, maxt ] = premnmx(matin', matout');
pr = minmax(dinn);
%
% define the range of training data

  pr=[-1 1
      -1 1
      -1 1
      -1 1];
  %
  % create a new network only in the first iteration;
  %
  net = newff(pr, [ninput 2*ninput+1 1], {'tansig', 'tansig',
'purelin'}, 'trainlm');
  net.trainParam.lr     = 0.01;
  net.trainParam.epochs = 50;
  net.trainParam.goal   = 1.0e-4;
  %
  % train the network
  %
  ht=cputime;[net,tr]=train(net, dinn, doutn);cputime-ht % calculate
the training time
```

```
xlabel('epochs');
ylabel('MSE');
%
% see how ANN performs on the training data;
%
yn = sim(net, dinn);
%
% We must postprocess the ANN output to compare with the
% original target
%
y = postmnmx(yn, mint, maxt);
t = postmnmx(doutn, mint, maxt);
%
% Plot both the target and the ANN output to see how they differ.
%
figure(2);
clf;
plot(t,'--');
hold;
plot(y,'r');
legend('target','ann-output');
xlabel('Hour');
ylabel('Electric demand(kW)');

% Plot the error curve in a different Figure.
%
figure(3);
clf;
plot(t-y);
xlabel('Hour');
ylabel('Error size(kW)');
display('ANN training completed. Hit any key to continue...');
pause;
%
% Now use the trained ANN to make prediction on test data.
%
totalhrs = ndays*nhrspdy;
p1 = delay0(ntrnhrs+1:totalhrs,1:4); % input
%
% the target is the electric usage at time t
t1 = delay0(ntrnhrs+1:totalhrs,5);
[tinn, minp, maxp, toutn, mint, maxt ] = premnmx(p1', t1');
y1n = sim(net, tinn);
y1  = postmnmx(y1n, mint, maxt)';
err = t1-y1;
%
figure(1);
clf;
plot(t1);
hold;
plot(y1,'r');
legend('measured','predicted');
xlabel ('Hour');
ylabel ('Electric demand (kW)');

%
figure(2);
```

```
  clf;
  plot(err);
  xlabel('Hour');
  ylabel('Error size(kW)');
  %
  figure(3);
  clf;
  [hbar,xh]=hist(err,15);
  bar(xh,hbar);
  xlabel('Error size (kW)');
  ylabel('Number of error occurance');
  cv   = norm(err)/sqrt(length(y1))/mean(t1);
  rmse = norm(err)/sqrt(length(y1));
  fprintf(' cv = %11.3e, rmse = %11.3e\n', cv, rmse);
```

2.Code for Experiment 4- Accumulative training with time-lagged temperature measurements as input

```
% electric demand prediction using accumulative ANN training%
%
clear;
format short e;
load delayk.mtx;
lag = 6;
[nrows,ncols] = size(delayk);

% time lagged input H(t),Td(t-k),Tw(t-k),Tl(t-1)
matin = delayk(:,1:14);
% output
%
matout = delayk(:,15);
%
nhrs      = 12;
nday      = 20;
nbase     = nhrs*nday; % number of data sets used to train the initial
ANN
interval = 12;         % The ANN is trained every interval hours
nsteps = 20;           % number of days to be predicted

ninput0 = 0;
for j = 1:nsteps
  %
  % preprocess training data
 [pn, meanp, stdp, tn, meant, stdt]= prestd(matin', matout');
 [ptrans, transMat]=prepca(pn,0.01);
  pr = minmax(ptrans);
  ninput  = size(ptrans,1);
  fprintf('j = %d, ninput = %d\n', j, ninput);
  % create a new network only in the first iteration;
  %
  if (j==1 | ninput0 ~= ninput )
     net = newff(pr, [ninput 2*ninput+1 1], {'tansig', 'tansig',
'purelin' },'trainlm');
     net.trainParam.lr      = 0.01;
```

```
      net.trainParam.epochs = 25;
      net.trainParam.goal    = 1.0e-4;
    end;
  %
  % train the network, compute the training time
  %
  ht=cputime;[net,tr]=train(net, ptrans, tn);cputime-ht
  %
  % see how ANN performs on the training data;
  %
  yn = sim(net, ptrans);
  %
  % We must postprocess the ANN output to compare with the
  % original target
  y = poststd(yn, meant, stdt);
  t = matout(1:nbase)';

  % Plot both the target and the ANN output to see how they differ.

  figure(1);
  clf;
  plot(t);
  hold;
  plot(y,'r');
  legend('target','ann-output');
  %
  % Plot the error curve in a different Figure.
  figure(2);
  clf;
  plot(t-y);
  fprintf('ANN training completed. Hit any key to continue...\n');
  %pause;

  % Now use the trained ANN to make prediction on test data.

  p1 = matin(nbase+1:nbase+interval,:)';
  t1 = matout(nbase+1:nbase+interval)';
  %
  % the target is the electric usage at time t

  p1n = trastd(p1, meanp, stdp);
  t1n = trastd(t1, meant, stdt);
  p1ntrans = trapca(p1n, transMat);
  y1n = sim(net, p1ntrans);
  y1  = poststd(y1n, meant, stdt);

  err(interval*(j-1)+1:interval*j) = t1-y1;
  mesr(interval*(j-1)+1:interval*j) = t1;
  pred(interval*(j-1)+1:interval*j) = y1;
  %
  % accumulate training data
  nbase = nbase + interval;
  ninput0 = ninput;

end;

figure(1) ;
```

```
clf;
plot(mesr,'--');
hold;
plot(pred,'r');
legend('measured','predicted');
xlabel('Hour');
ylabel('Electric demand(kW)');

figure(2);
clf;
[hbar, xh]=hist(err,15);
bar(xh, hbar);
xlabel('Error size(kW)');
ylabel('Number of error occurance');
cv=norm(pred-mesr)/sqrt(length(mesr))/mean(mesr);
rmse= norm(pred-mesr)/sqrt(length(mesr));
fprintf(' cv = %11.3e, rmse = %11.3e\n', cv, rmse);
```

3.Code for Experiment 15- -On-line prediction using SC*i*(t) with sliding window

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% initial chiller prediction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [pred, mesr, cv, rmse] = csldpca(cutoff, trnalg, nepochs);

% electric demand prediction using sliding window ANN training
cutoff=0.01;
trnalg=2;
nepochs=25;

format short e;
%
% load ice bank status indicator.  When ice bank has
% ice left, the chiller is turned off.
%
load icestat.txt
%
% load working hour measurements.  Part of the data will
% be used to train the first ANN which predicts the
% variations of the chiller usage with respect to
% the environmental and operational variables.
%
load chilwork.txt
%
% load the entire meaurement data.  Part of the data will
% be used to test the ANN.
%
%
load chillall.txt

[nrows,ncols] = size(chillall);
%
% === reorganize the data, seperate the input from the output ==
%== remove the zero data from the original dataset
```

```
nzdata=chillall;
izer=find(abs(nzdata(:,5))<=0.1);
nzdata(izer,:)=[];
[mnz, nnz] = size(nzdata);
lag=6;
%
% Take every other sample from nzdata to form the training data set.
%
trdata = nzdata(1:1:mnz,:);
[mtr,ntr] = size(trdata);

interval = 48; % The ANN is trained every two days due to the %
%incomplete data%
tbeg=1;
tend=mtr-123;

% training data from August,2002 to May , 2003
pbeg=3468;
pend=4127;
nbase=pbeg;

% input variables from previous hour
matin = [trdata(tbeg+lag-1:tend-1,4:6)...
         trdata(tbeg+lag-1:tend-1,18:20)];
for j = 2:lag
   matin =[matin trdata(tbeg+lag-j:tend-j,4:6)...
            trdata(tbeg+lag-j:tend-j,18:20)];
end;

ninput = size(matin,2);
matout =trdata(tbeg+lag:tend,5);
% output
% normalize the data
%
nsteps = 10;
ht1=cputime;
ninput0=0;

for i= 1:nsteps

[pn, meanp, stdp, tn, meant, stdt] = prestd(matin', matout');
[ptrans, transMat] = prepca(pn, 0.01);
ninput = size(ptrans,1);
fprintf('ninput = %d\n', ninput);
fprintf('step = %d\n',i);

 pr = minmax(ptrans);
  %
  % create a new network only in the first iteration;
  % if the number of inputs in the following iterations is the same,
the network is kept unchanged;
  % otherwise it is reinitialized

  if (j==1 | ninput ~= ninput0)
     if ( trnalg == 1)
        net1 = newff(pr, [ninput 2*ninput+1 1], {'tansig', 'tansig',
'purelin' },'trainlm');
```

```
     else
        net = newff(pr, [ninput 2*ninput+1 1], {'tansig', 'tansig',
'purelin'},'traingdm');
     end;
 end;
     net1.trainParam.lr      = 0.01;
     net1.trainParam.epochs = nepochs;
     net1.trainParam.goal    = 1.0e-4;
  %
  % train the network
  %
  [net1,tr]=train(net1, ptrans, tn);
  xlabel('epochs');
  ylabel('MSE');
  %
  % see how ANN performs on the training data;
  %
  yn = sim(net1,ptrans);
  %
  % We must postprocess the ANN output to compare with the
  % original target
  %
  y = poststd(yn, meant, stdt);
  t = matout';
  %
  % Plot both the target and the ANN output to see how they differ.
  %
  figure(1);
  clf;
  plot(t,'--');
  hold;
  plot(y,'r');
  legend('target','ann-output');
  %
  display('ANN training completed. Hit any key to continue...');
  pause;
  %
  % Now use the trained ANN to make prediction on test data.
  %
  p1= [chillall(nbase+lag-1:nbase+interval+lag-1,4:6)...
        chillall(nbase+lag-1:nbase+interval+lag-1,18:20)];

 for j = 2:lag
   p1 =[p1 chillall(nbase+lag-j:nbase+interval+lag-j,4:6)...
           chillall(nbase+lag-j:nbase+interval+lag-j,18:20)];
 end;

 % target of testing data
 t1 = chillall(nbase+lag:nbase+interval+lag,5);

  % the target is the electric demand at time t
  %
  p1n = trastd(p1', meanp, stdp);
  t1n = trastd(t1', meant, stdt);
  p1ntrans = trapca(p1n, transMat);
  y1n = sim(net1, p1ntrans);
  y1 = poststd(y1n, meant, stdt)';
  err1 = t1-y1;
  %
  % accumulate training data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %indicator prediction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  t2 = icestat(nbase+lag:nbase+interval+lag,1); % chiller on-off status
at time t
  y2 = t2;
  %
  fout=y1.*y2; %using the output from the first network times chiller
status value

  mesr(interval*(i-1)+1:interval*i+1) = t1;
  pred(interval*(i-1)+1:interval*i+1) = fout;

  newipt=find(abs(t1)<=0.1);
  p1(newipt,:)=[];
  t1(newipt,:)=[];
  if (p1~=0)
     matin=p1;
     matout=t1;
  else
     matin=matin;
     matout=matout;
  end;

  %Plot the predicted electric demand and actual demand
  figure(2);
  clf;
  plot(mesr,'--');
  hold;
  plot(pred,'r');
  legend('measured','predicted');
  xlabel('Hour');
  ylabel('Chiller electric demand(kW)');
  %
  err=mesr-pred;

  figure(3);
  clf;
  plot(err);
  xlabel('Hour');
  ylabel('Error size(kW)');
  %
  figure(4);
  clf;
  [hbar,xh]=hist(err,15);
  bar(xh,hbar);
  xlabel('Error size(kW)');
  ylabel('Number of occurance');
  cv   = norm(err)/sqrt(length(err))/mean(t1);
  rmse = norm(err)/sqrt(length(err));
```

```
fprintf('cv = %11.3e, rmse = %11.3e\n', cv, rmse);

nbase = nbase + interval;
ninput0=ninput;

end;
ht2=cputime;
httotal=ht2-ht1
```

APPENDIX C – METHODS OF CONVERTING ORIGINAL DATA OBTAINED FROM CANMET TO MATLAB FORMAT

To make use of the data in the MATLAB ANN prediction program, we converted EnergyData.txt into a matrix format that can be easily manipulated in MATLAB. Two matrix files are created as the result of this conversion process. One contains measurements related only to chiller energy prediction, and the other contains measurements related only to boiler energy prediction. Each matrix consists of multiple rows and columns. The first four columns list the month, the day, the year and the hour at which the measurements are made. Column 5 represents either the chiller energy demand (kW) or the boiler gas demand (kW). The next columns correspond to other independent variables extracted from the original data. Under this representation, each matrix row contains the values of various variables measured at a specific hour. An example of the matrix file corresponding to heater gas usage is shown below, where

1st column: month

2nd column: day

3rd column: year

4th column: hour

5th column: kw_gaz

6th column: tsch1

7th column: tsch2

8th column: tta_eau_rcp

9th column: ttr_eau_rcp

10th column: ad_chaudiere1

11th column: ad_chaudiere2

12th column: gv_heat_central

13th column: modul_chaud1

14th column: modul_chaud2

15th column: fete_m2_m3

16th column: ws_systeme_m3

17th column: text_12

```
9  4 2002  9    0.000e+000    2.500e+001  2.610e+001  2.566e+001  2.833e+001  0  0  0.000e+000  0.000e+000  0.000e+000  0  0  0.000e+000
9  4 2002 10    0.000e+000    2.510e+001  2.553e+001  2.586e+001  2.753e+001  0  0  0.000e+000  0.000e+000  0.000e+000  0  0  0.000e+000
9  4 2002 11    0.000e+000    2.525e+001  2.584e+001  2.575e+001  2.615e+001  0  0  0.000e+000  0.000e+000  0.000e+000  0  0  0.000e+000
9  4 2002 12    0.000e+000    2.520e+001  2.560e+001  2.566e+001  2.560e+001  0  0  0.000e+000  0.000e+000  0.000e+000  0  0  0.000e+000
9  4 2002 13    0.000e+000    2.506e+001  2.523e+001  2.546e+001  2.573e+001  0  0  0.000e+000  0.000e+000  0.000e+000  0  0  0.000e+000
9  4 2002 14    0.000e+000    2.510e+001  2.520e+001  2.540e+001  2.565e+001  0  0  0.000e+000  0.000e+000  0.000e+000  0  0  0.000e+000
9  4 2002 15    0.000e+000    2.510e+001  2.520e+001  2.529e+001  2.560e+001  0  0  0.000e+000  0.000e+000  0.000e+000  0  0  0.000e+000
```

To change the original data into the format described above, we took the following steps:

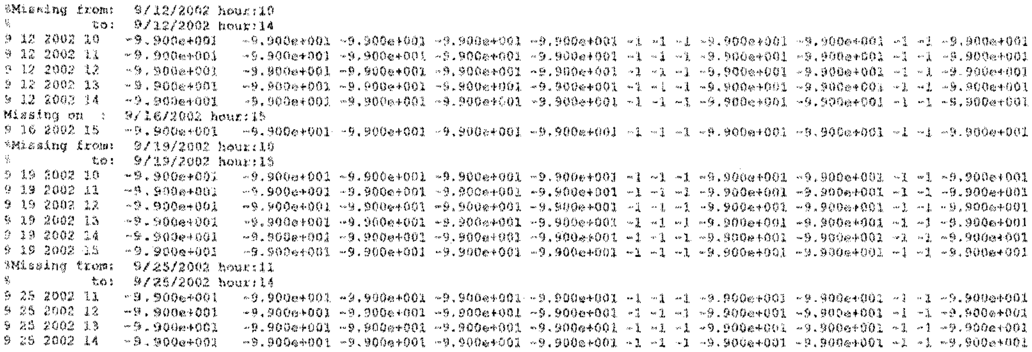
First we create a table called *gaztable* that contains 6696 rows and 17 columns, where 6696 is the total number of hours between 12:00 pm 6/21/2002 and 12:00 am 3/27/2003. Each row of this table is used to store data measured at a particular hour. The data entries of the table are initialized to –99.0 if they are floating point numbers and –1 if they are integer values. This initial value indicates that no value was read from the database yet. The first column of the table lists the month in which the data is measured, the second column lists the day, the third column lists the year, and the fourth column lists the hour.

We then read the EnergyData.txt file line by line. Each line is parsed so that the month, date, year, hour, variable name and its quantity are separated.

Based on the month, date, year and hour at which the variable is measured we can easily calculate the row index associated with the measured data just retrieved from EnergyData.txt.

Once the row index *i* is obtained, we can simply store the measured quantity to the appropriate column of the *i*-th row in *gaztable*.

Once this conversion process is completed, it is discovered that *gaztable* contains many – 1.0 and –1, the original initialized values indicating that the original data set is incomplete. As a result, a piece of code is added to go through the *gaztable* and print out the beginning and ending hour of the period during which the values of `KW_GAZ` are missing. A separate file called gazpatch.txt is created to store only rows of *gaztable* that correspond to missing data. The file looks like the following:



```
%Missing from:  9/12/2002 hour:10
%           to:  9/12/2002 hour:14
9 12 2002 10   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 12 2002 11   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 12 2002 12   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 12 2002 13   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 12 2002 14   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
Missing on   :  9/16/2002 hour:15
9 16 2002 15   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
%Missing from:  9/19/2002 hour:10
%           to:  9/19/2002 hour:15
9 19 2002 10   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 19 2002 11   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 19 2002 12   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 19 2002 13   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 19 2002 14   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 19 2002 15   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
%Missing from:  9/25/2002 hour:11
%           to:  9/25/2002 hour:14
9 25 2002 11   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 25 2002 12   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 25 2002 13   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
9 25 2002 14   -9.900e+001   -9.900e+001 -9.900e+001 -9.900e+001 -9.900e+001 -1 -1 -1 -9.900e+001 -9.900e+001 -1 -1 -9.900e+001
```

Using gazpatch.txt as the starting point, we can then interpolate the missing data manually in gazpatch.txt. Once the interpolated data has been created in gazpatch.txt, an additional code is used to read both the original data file EnergyData.txt and the interpolated data file gazpatch.txt to create a final matrix data file gaztable.txt.

A similar strategy is used to create a data file for chiller energy demand.

Figure. 1 shows the diagram of the data processing for gas usage data.

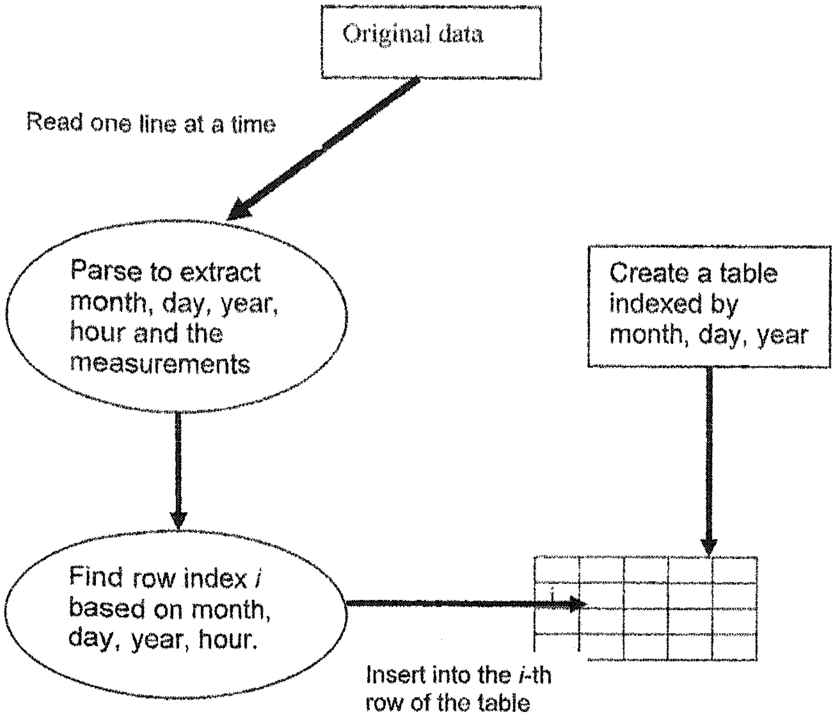

Original data
Read one line at a time
Parse to extract month, day, year, hour and the measurements
Create a table indexed by month, day, year
Find row index *i* based on month, day, year, hour.
i
Insert into the *i*-th row of the table

Figure.1 Data processing diagram for gas usage