THE APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO SHORT TERM LOAD FORECASTING

BY

C. HART POSKAR

A Thesis Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

ISBN 0-315-86137-1

Name C. H. Poskar

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

Engineering - Electronics and Electrical

SUBJECT TERM

0544

SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984

Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

THE APPLICATION OF ARTIFICIAL NEURAL NETWORKS

TO SHORT TERM LOAD FORECASTING

by

C. HART POSKAR

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

ABSTRACT

This work provides an evaluation of the application of Artificial Neural Networks (ANN's) to the problem of Short Term Load Forecasting (STLF) for a large and varied geographical region with extreme weather conditions. Using data supplied by Manitoba Hydro Electric Utility an ANN was optimized for forecasting the Manitoba firm system load for the nest day. Feedforward ANN's using the Backpropagation algorithm were found to be well suited to STLF, combining both weather related and time sequence forecasting. Direct comparison of the ANN using forecasted weather to the present method used by Manitoba Hydro for a month chosen by Manitoba Hydro was performed. The forecasts performed by Manitoba Hydro for that month resulted in an average percent error of 5.8%, with the ANN forecast at 5.6% using 24 hour weather and 5.9% using 4 hour weather. The reliability of the forecasts using ANN's combined with their ability to perform at this level without the aid of an experienced system operator, make ANN's an attractive alternative for STLF. Findings using unsupervised learning algorithms supported the evaluation performed using supervised learning, and are summarized in the Appendix.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. G. Swift and Dr. W. Pedrycz, for their individual visions for this research which I feel have led to a greater whole.

I would also like to thank Dr. B. Kermanshahi, for his help throughout the research, and Peter Czezowski, for his patience with unsupervised learning.

I would also like to acknowledge the help and interest of Manitoba Hydro in this thesis, without which this research could not have been performed. In particular W.D. Buhr for his support of load forecasting, as well as the contributions and help of A. Silk.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The electrical load is the power that an electrical utility needs to supply in order to meet the demands of its customers. It is therefore very important to the utilities to have advance knowledge of their electrical load, so that they may ensure that this load is met and to minimize any interruptions to their service.

Short Term Load Forecasting (STLF) is the general process of forecasting the electrical load of a utility from one minute to one day in advance. This forecast is then utilized for the scheduling of system generation and load distribution. This allows for the advance scheduling of power sales and/or purchases, as well as scheduling of general maintenance on sections of the power system infrastructure.

Historically the need for short term load forecasting has forced its evolution in some form by every electrical utility. Originally STLF was performed solely on the basis of the experience and observations of the system operator. Then, with the advent of the computer, utilities found a tool that could be used to contain massive amounts of data and perform the computations on this data to implement a variety of algorithms which could model the load and extrapolate or make forecasts based on this historical information. The particular model and technique varied to suit the particular needs of each utility.

For Short Term Load Forecasting two general model types have evolved, with many techniques for implementing both of them: time sequence models, and weather dependent models [1–18]. Hybrid models have arisen more recently, which separate the load into weather dependent and weather independent components which are evaluated, then combined to provide the forecast [1–2]. There have been five main techniques used to implement the specific model [3]. These techniques are : Multiple Linear Regression [2,4], Stochastic Time Series [5], General Exponential Smoothing [5], State Space Methods [1], and Knowledge Base Expert Systems [6–7]. The last technique represents a break from the statistical approach of the others, by attempting to perform forecasts by expressing the operators experience and observations in terms of rules.

Recently a sixth technique has been used in the field of Short Term load Forecasting, which also breaks from just improving the model. This technique, Artificial Neural Networks (ANN's), like the Expert Systems, attempts to use experience and observations to perform forecasting [8–18]. However in the case of an ANN the knowledge is supplied by the historical data rather than through the experience of an expert user, and the system learns for itself, representing this knowledge in a modifiable distributed weight matrix rather then a fixed set of coded rules.

Research in the area of the application of Artificial Neural Networks to load forecasting over various levels (system to feeder) and various time ranges (short, medium, and long term) has shown much promise. However the thrust of this research seems to be in the direction of single valued forecasts rather then over a continuous series, using sums and averages with other techniques (as described above) to decrease the average error and standard deviation of both input data and forecasts [8–13]. Furthermore they tend to use large amounts of preprocessing and data selection schemes for the training data, creating systems which may not degrade gracefully if the actual data used is in anyway inaccurate [12–14].

1.1 PURPOSE

Manitoba Hydro (MH) currently employs a composite multiple linear regression (MLR) technique to perform short term load forecasting. This technique has been found to have many inherent shortcomings in both the algorithm and the method of implementation. It is therefore desirable for MH to develop a new STLF system which addresses the current problems, and which is sufficiently adaptable to meet their future needs.

Some of the problems with the MLR technique are :

- the assumption of a linear relationship between load and weather variables,
- the assumption of a uniform relationship over the entire day,
- no tolerance for poor weather forecasts, and
- the reliance on chosing a good "reference"[1] day.

These problems with the present method of STLF at Manitoba Hydro must be addressed by any new technique developed for them. This new technique must be able to take advantage of non–linear relationships, different relationships for different regions of the day, and to have inherent fault tolerance in order to compensate for bad or incorrect data. Furthermore, any new method should try to eliminate as many sources of error as possible, such as the choosing of a reference day, to decrease demands on the valuable time of an experienced system operator. One relatively new technique, (to STLF), that meets these requirements is Artificial Neural Networks. This technique has recently been applied to various forms of load forecasting, and has shown promise when compared with many previous techniques [8–18]

The purpose of this thesis is to examine the use of ANN's for the problem of short term load forecasting, for the Manitoba Hydro Electric Utility.

1. A day, from the past, on which to base the forecast.

1.2 SCOPE

This thesis will look at the applicability of Artificial Neural Networks to perform short term load forecasting, using historical data supplied by Manitoba Hydro Corporation, between January 1989 and May 1992.

A variety of aspects of ANN's will be analyzed in determining a model suitable for Manitoba Hydro. These aspects include the network architecture of the proposed models, and method of training. The focus of this research will be on supervised learning[2], using the Backpropagation (BP) algorithm. The network architecture will look at input and output structure, hidden neurons, feedback and modular nature of ANN's. Finally the training structure for continuous forecasting will be evaluated.

The analysis of results will be based on the daily average percent errors and the daily peak percent errors, both of which are critical to MH.

Final results for each learning method will be compared with actual past forecasts performed by Manitoba Hydro using their present method, for a month chosen by Manitoba Hydro.

1.3 STRUCTURE

Chapter 2 provides a general background of the present method of Short Term Load Forecasting at Manitoba Hydro and of Artificial Neural Networks. This is immediately followed by a general introduction to ANN's and a description of supervised learning.

2. Unsupervised learning will be briefly presented in Appendix C.

In Chapter 3 the application of ANN's to STLF will be investigated using supervised learning. This chapter is divided to examine the systematic evaluation of the application to STLF with respect to the general and specific network architecture, and training structure of the ANN.

A comparison of forecasting using ANN's, to the present method of forecasting employed by MH is presented in Chapter 4.

Chapter 5 will then present the final conclusions and recommendations as determined by this research.

Appendix A shows daily graphs of the comparison results presented in chapter 4.

Appendix B provides the BP algorithm used in this research.

Appendix C gives a brief overview of ANN's implementing unsupervised learning, and some results found using unsupervised learning for STLF.

2. BACKGROUND

2.1 THE PRESENT METHOD

The present method of short term load forecasting employed by Manitoba Hydro is **Multiple Linear Regression** analysis, or **MLR** [19]. MLR is based on the premise that there is a linear relationship between a dependent variable (**F**) and changes between one or more independent variables (ΔA_i). This relationship is illustrated by equation (1) :

$$\mathbf{F} = \mathbf{a}_0 + \mathbf{a}_1 \Delta \mathbf{A}_1 + \mathbf{a}_2 \Delta \mathbf{A}_2 + \mathbf{a}_3 \Delta \mathbf{A}_3 + \ldots \qquad (1)$$

The MLR technique then calculates the coefficients $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$, ..., corresponding to each independent variable. The sum, $\Sigma \mathbf{a}_i \Delta \mathbf{A}_i$, for $i \geq 1$, then becomes the change in the dependent variable due to the independent variables. The constant coefficient, $\mathbf{a}_0$, is the base value of the dependent variable to which any change due to changes of the independent variables is added. These coefficients are calculated so as to minimize the error between the right and left hand sides of equation (1) over a given system of these equations.

For STLF the dependent variable is the load, and the independent variables are weather components. Manitoba Hydro uses three weather components in the MLR analysis. These are temperature (T), wind speed (W), (neglecting wind direction), and sky cover (S).

The basic relationship which is analyzed at Manitoba Hydro is given as :

$$\Delta L = k_T \Delta T + k_W \Delta W + k_S \Delta S \tag{2}$$

where ΔL is the change in load from a given reference day to the forecast day. Similarly ΔT, ΔW, and ΔS are respectively the changes in the temperature, wind speed and sky cover from the given reference day to the forecast day. The coefficients $\mathbf{k_T}$, $\mathbf{k_W}$, and $\mathbf{k_S}$ are calculated using the MLR technique on a set of historical load and weather data. The system of equations consists of one equation for each hour in a given hour set, of each day in a given day set, for each year in a given year set. The changes to the given variables are calculated as shown in equations (3a) through (3d) below. The subscripts denote to which day the variable is referring, **F** for the variable on the forecast day, and **R** for the variable on the reference day. For example, equation (3a) reads, the change in load is equal to the forecast day load minus the reference day load.

$$\Delta L = L_F - L_R \tag{3a}$$

$$\Delta T = T_F - T_R \tag{3b}$$

$$\Delta W = W_F - W_R \tag{3c}$$

$$\Delta S = S_F - S_R \tag{3d}$$

The forecasted load for a particular day is then given by rewriting equation (3a) as:

$$L_F = L_R + \Delta L \tag{4a}$$

Substituting for ΔL using equation (2) gives us :

$$L_F = L_R + k_T \Delta T + k_W \Delta W + k_S \Delta S \tag{4b}$$

This is the same basic equation that the MLR technique solves using historical data. The only difference is that a reference day load is used instead of the constant coefficient or base load, to which the change in load due to weather factors is added.

An example of how a daily load curve might be forecast based on equation (4a) is shown in figure 2.1. In the system employed by MH the change in load is calculated for between four and six individual hours. A minimum of four hours are broken up into two pairs of hours which border the ranges for the am and pm peak loads. These hours are 9:00 and 12:00, for the am peak range, and 17:00 and 22:00 for the pm peak range. The other two hours consist of the am and pm peak hours of the reference day, if they are not already one of the above four. The change in load for the remaining hours between 9:00 and 22:00 are linearly interpolated, and for the hours before 9:00 and after 22:00 the change in load is assumed to be the same as at 9:00 and 22:00, respectively.

The coefficients k_T, k_W, and k_S are actually three sets of coefficients (one for each of temperature, wind speed, and sky cover) which vary throughout the year in regular patterns, and vary from year to year with system load growth (or decay). To remove any anomalous data, the MLR technique requires data over a sufficiently long period of time. However a balance must be struck so as not to average out the effect of system load growth over this time. It was found experimentally that at least three to four years worth of data was

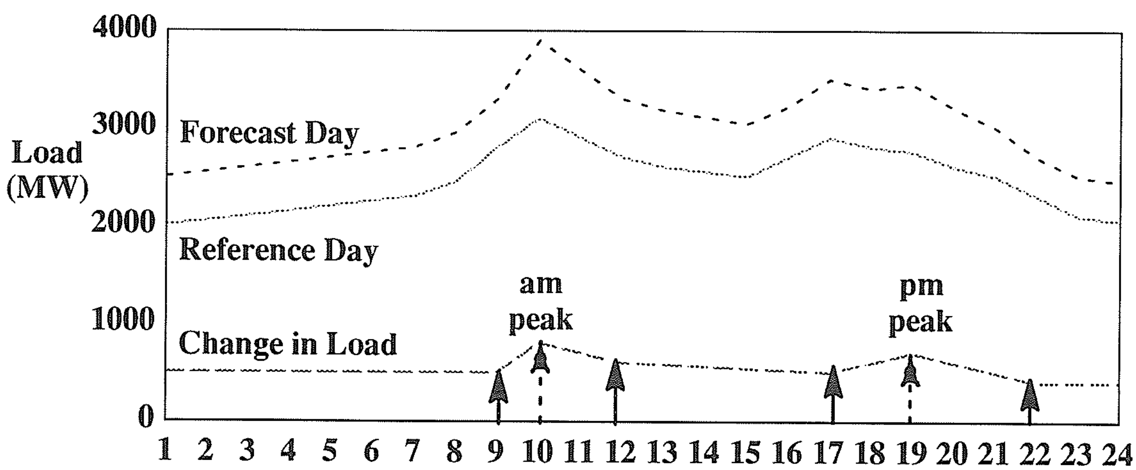


FIGURE 2.1 : Present Method of Daily STLF

required as a minimum, but that when approaching ten years of data the growth factor could not keep up due to the averaging effect. For this reason seven years of data was chosen.

Due to this annual regularity and the computational time required to perform the linear regression, it was decided to calculate only one value for k_T, k_W, and k_S for each month of the year, and to use non–linear smoothing to develop an annual set of points for each coefficient. The system of equations that the MLR solves consists of one equation for the daily peak hour, with the preceding and following two hours, for the weekdays or weekends (but no holidays) of a given month of the year, for each of the last seven years.

Once the coefficients have been developed, the actual operation is quite simple. The forecast day is chosen as the next day. A weather forecast for this day is supplied from an outside source, at a time as close to the forecast day as is still useful to the system operator. This data is then entered into a program along with advance knowledge of changes to industrial load sites (ie. a 200 MW smelter shutting down for a month), and the forecast day. The user is then prompted to choose a reference day from the historical data base. In general a reference day must[3] :

1) be the same day of the week as the forecast day,
2) be from the recent past (usually no more then one year), and
3) have similar weather patterns as the forecast day.

The user is then provided with the 24 hour load forecast, by means of equation (4b) above.

This system has been developed and used for the past several decades. In that time it has been found that while in general performing adequately, there are several areas where improvement could be made.

The area of greatest error using this system is the accuracy of the weather forecast. While the accuracy of the weather forecast can not be improved, a system which would not rely on this accuracy so greatly would be an improvement.

3. These characteristics do not hold for holidays.

A second drawback to this system is that it heavily relies on having expert users. The user is required to choose an appropriate reference day (based on the weather forecast), and must evaluate the load forecast, correcting it or even replacing it with a new one. Choosing the reference day is key to this system, as it is used as the base to which the calculated change in load is added. For making this choice, the simple guidelines often lead to several choices for reference day, but not all (or necessarily any) of them are good ones (a common problem in an environment with many extremes of weather and changes in weather). However the error for the load forecasts can range to over 20% even after the experienced user has made corrections based on their outside knowledge. Therefore a system which could lower the dependency on the choice of reference day and make fewer demands on the system user would be advantageous.

Lastly there are the basic assumptions used to devise and apply this method. The assumption that the relationship between load and weather is linear, that the change in load depends on only one reference day, and that the relationship for each hour is the same for a given day. To investigate these assumptions a nonlinear system could be developed with various input data, and connection schemes. This system could then model both linear and non–linear relationships, and would therefore be an improved system.

2.2 ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) is a parallel distributed system that attempts to model the connectivity and simple biological processing cells (Neurons) of the brain [20].

A parallel distributed system is one with a large number of processing elements capable of working in parallel, and inter–connections between these elements. In this type of system, knowledge is not treated as local representation, as with the coefficients with the MLR process, but rather as distributed representation. Distributed representation means that no single processing element has any consistently unique meaning. Instead it is the pattern distributed over many processing elements which represent the knowledge of the system.

In an ANN the knowledge of the network is determined by the strength of the inter–connections, called synapses, of the neurons in the network. This synapse strength is represented as a real number weight connecting the output of one neuron to the input of another neuron. With the MLR technique each coefficient has a specific meaning, and its contribution to the output is explicitly defined. In an ANN it is the combined effect of the network inputs, and neuron outputs, modified by these synaptic weight values through which they flow, that produces the desired output. No single weight value has any definite meaning.

Parallel distributed systems have certain advantages over other systems. In general the knowledge (or weights) does not need to be explicitly known in advance. Instead the knowledge can be learned by training the network. This adaptability combined with the distributed representation in the network is the property of recognition of similar patterns and generalization to abstract patterns, within the network input space. It is this ability that allows an ANN to compensate and degrade gracefully in performance with bad training data, and unreliable inputs.

Figure 2.2 shows a diagram of the i^{th} Neuron in a feedforward Artificial Neural Network. An ANN consists of **n** basic processing elements called Neurons, (represented by circles in the diagram). These neurons are connected to one another by uni–directional links, (represented as lines). Each link has a real number weight, w_{ij}, associated with it, (represented by filled rectangles on the lines). For all of the feedforward networks we will be using, the neurons will be categorized into one of three types. Those with external inputs called input neurons, those with external outputs called output neurons, and those with no external inputs or outputs called hidden neurons.

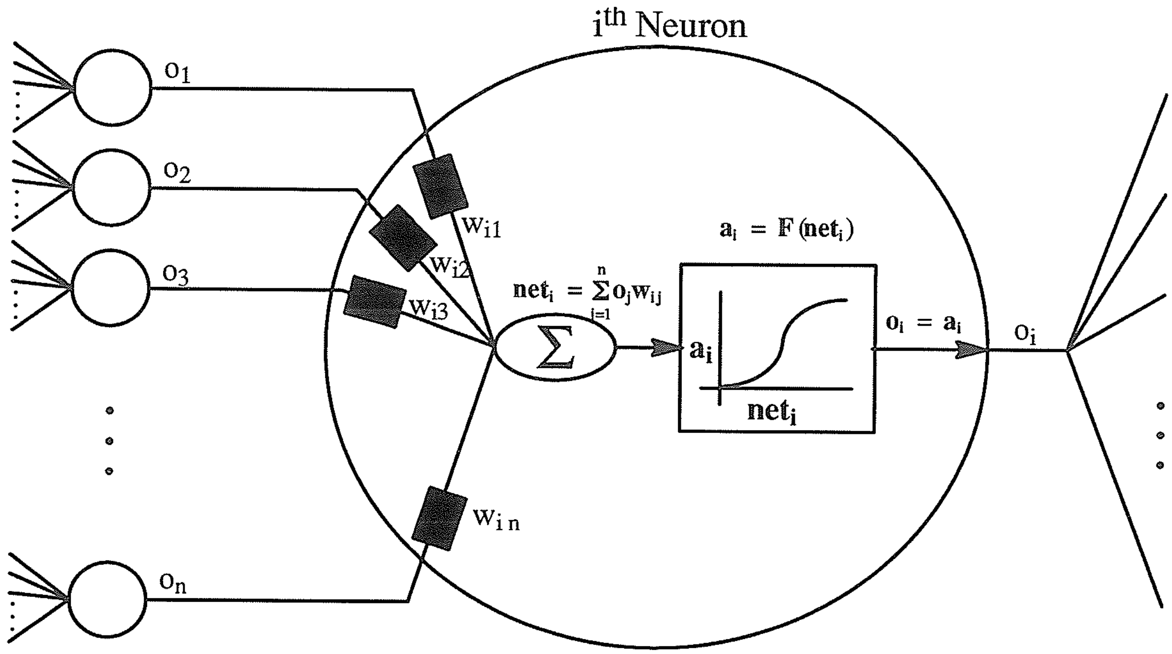


FIGURE 2.2 : A Neuron

In general each neuron in an ANN consists of three basic parts, the net input to the neuron, the state of activation of the neuron, and the neuron output. There may be many classes of input to a neuron which affect the activation of the neuron in different ways. These may be external inputs to the system, or the outputs of other neurons. Each class has its own net input which is passed to the activation function. The activation function combines the various net inputs of the neuron and the present activation of the neuron, to determine the new state of activation of the neuron. The activation of the neuron is then passed to an output function which produces the new neuron output, which can be passed to other neurons or to outside the network.

As can be seen in figure 2.2 the basic neuron which we will be using, consists of simplified versions of the three basic parts described above. The $\mathbf{i}^{th}$ neuron computes a net input equal to the sum of the output of the $\mathbf{j}^{th}$ neuron, $\mathbf{o_j}$, times the weight connecting $\mathbf{o_j}$ to the $\mathbf{i}^{th}$ neuron, $\mathbf{w_{ij}}$, for $1 \leq \mathbf{j} \leq \mathbf{n}$. The net input is then passed to an activation function that bounds the activation value to a predefined range. This function is typically a non–linear and differentiable[4] function called a sigmoid (S–shaped). Finally the output is equal to the activation value of the neuron.

In the special case of the input neurons, the activation value is equal to the external input, and passes this value directly to the neuron output.

There is also a method for updating the weights associated with an ANN, called the learning rule. The learning rule and algorithm which implements the rule is specified for each particular ANN, but they all have their roots in the hypothesis of D.O. Hebb [21] :

> "Any two cells or systems of cells that are repeatedly active at the same time will tend to become 'associated,' so that activity in one facilitates the activity in the other."
>
> "... A growth process accompanying synaptic activity makes the synapse more readily traversed."

This hypothesis that learning occurs by changing the synaptic strength connecting associated cells, and not in the cell itself is the foundation of the hebbian learning rule usually stated:

> When two cells fire (are activated) at the same time, then the strength of the connection between them should be increased.

and in general this rule can be written as [22]:

$$\Delta w_{ij} = \epsilon a_i a_j \qquad (5)$$

where a_i and a_j are the activations of two neurons, i and j, then the change to the weight is proportional to the product of these two activations. The constant of proportionality, ϵ, is called the learning rate, and acts to scale the change to the weights.

Learning is performed by an ANN through the implementation of the learning rule which changes the weight values interconnecting the network. During learning a set of training patterns, called the training set, is presented one pattern at a time to the network.

4. This property will be discussed in section 2.2.2 The Backpropagation Algorithm.

The training set pattern always consists of at least one component, an input vector. After each training pattern input vector is presented to the network, all of the neuron activations are updated, and neuron outputs are produced. Based on the updated neuron outputs, the network weights are updated following the particular learning rule. This update can occur after each pattern, or can be accumulated over the entire training set, and then applied.

The specific ANN's that are to be evaluated in this research are discussed next. For each algorithm we specify what kind of neurons are to be used, the general pattern of connectivity between the neurons, and the specific learning rule to be implemented.

2.2.1 SUPERVISED LEARNING

Supervised learning is a class of learning rules which requires two components for a training set pattern. This pair consists of an input vector and a target vector. The target vector is used during the learning phase for comparison to the network output, produced by the particular input vector of the training pair. When the output vector does not equal the target vector an error is produced, and the weights are adjusted.

The error function is defined using the Least Mean Square (LMS) error :

$$\mathbf{E_p} = \frac{1}{2}\sum_{\mathbf{i}}(\mathbf{t_{pi}} - \mathbf{o_{pi}})^2 \tag{6}$$

$$\mathbf{E} = \sum_{\mathbf{p}} \mathbf{E_p} \tag{7}$$

The error for each pattern $\mathbf{E_p}$ is the sum of the LMS errors as defined in equation (6), where $\mathbf{p}$ indexes the specific training pattern, and $\mathbf{i}$ indexes each element in the target vector ($\mathbf{t_{pi}}$) and each corresponding output neuron ($\mathbf{o_{pi}}$). The total error over all training patterns $\mathbf{E}$, is then the sum of each training pattern error.

The weights are adjusted or updated by a learning rule, which is defined to minimize the error function. There are many known algorithms which minimize multidimensional functions. One class of minimization algorithms which has been found to converge quickly in general and for ANN applications is the conjugate gradient (CG) minimization [23] This technique will be used with the Back Propagation algorithm to minimize the error function.

2.2.2 Back Propagation Algorithm

The Back Propagation (BP) algorithm[5] is so named because during training after propagating the input vector through the network the weights are adjusted by propagating the error backwards through the network. This allows for the use of hidden neurons for which the error function cannot be directly calculated.

In a feed forward network, such as in figure 2.3, there can be a layer of input neurons, a layer of output neurons, and one or more layers of hidden neurons, connected in only one direction. During training only the output neurons have corresponding target vectors from which to directly calculate the error component for each output neuron. For hidden neurons the error must be propagated back from the output at each layer, so as to calculate the error component for each hidden neuron with respect to its inputs. Only then can the weights between a hidden neuron and each of its inputs be updated.

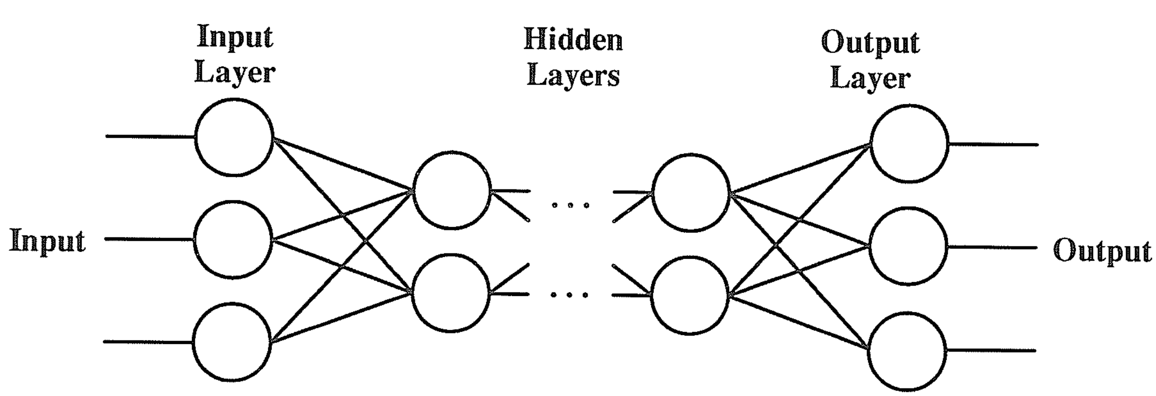


FIGURE 2.3 : A Feed Forward Network

5. The algorithm is presented in Appendix B.

The learning rule for this algorithm, implementing CG minimization, is :

$$\Delta w_{ij} = \lambda h_{ij} \tag{8}$$

where Δw_{ij} is the change in weight connecting neuron j to neuron i, λ is the distance to the new minimum in the conjugate direction $\mathbf{h}$, and h_{ij} is the component of the direction vector corresponding to the component w_{ij} of the weight vector.

The algorithm requires the calculation of the negative gradient of the error function, in order to minimize the error function with respect to the weight space.

When a training pattern, **p**, is presented to the network, the error is calculated for this pattern as in equation (6). The negative gradient of the error function is [22]:

$$-\frac{\partial E_p}{\partial w_{ij}} = \delta_{pi} a_{pj} \tag{9a}$$

where for output neurons :

$$\delta_{pi} = (t_{pi} - o_{pi})\, a'(net_{pi}) \tag{9b}$$

and for hidden neurons :

$$\delta_{pi} = \sum_k \delta_{pk} w_{ki}\, a'(net_{pi}) \tag{9c}$$

where **k** references the neurons in the layer above the layer being evaluated, so that they are already known. This step is the back propagation of the error previously described.

Equations (9b) and (9c) require that the activation function be differentiable to implement the BP algorithm. The activation functions that will be used is a sigmoid called the logistic function, which can be seen in figure 2.4, and is defined by equation (10a) :

$$a(net) = \frac{1}{1 + e^{-net}} \tag{10a}$$

Differentiating this activation function we get :

$$a'(net) = a\ (1 - a) \tag{10b}$$

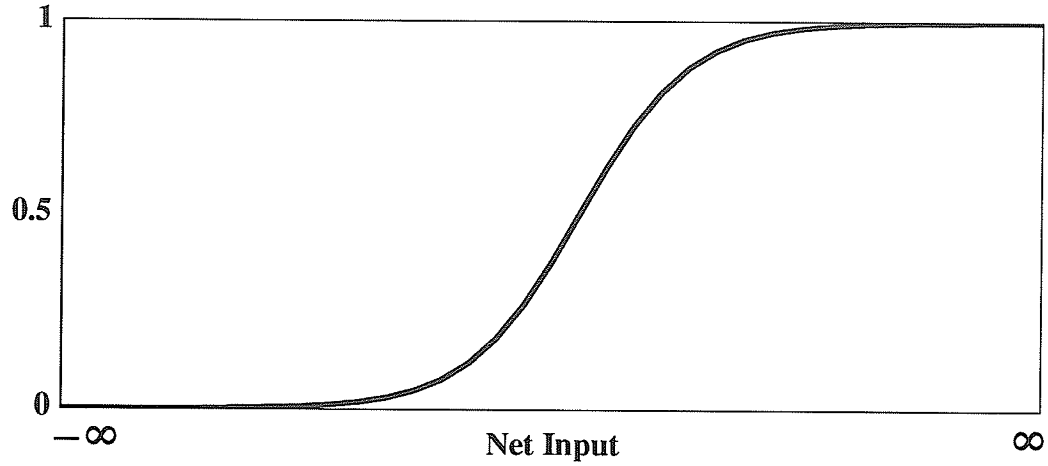


FIGURE 2.4 : The Logistic Activation Function

3. ANALYSIS

3.1 PROCEDURE

A systematic approach is taken to investigate the application of Artificial Neural Networks with supervised learning (via the backpropagation algorithm as described in Chapter 2) to the problem of Short Term Load Forecasting. The approach is broken into five areas, each building on the results for the previous area : General Network Structure, Structure of the Input Layer, Network Modularization, Training, and Structure of the Hidden Layer.

To begin with the general network structure is investigated. In this area the question of hidden layers will be investigated, as well as parallel versus serial output for daily STLF. The results will be compared for each season, independently, and then as a whole.

Then once a general network has been established, the structure of the input layer will be investigated. This area is broken into two groups, the basic input types given to the network, and feedback from previous time steps within the network (which is modeled as input).

Once the input structure for the network has been determined the method of training the network will be evaluated. Specifically how often to train, and how much training data to use will be evaluated, but not training parameters (such as the learning rate).

The next area to be evaluated is the modularization of the network into sub–networks. This will be done in two parts : the days of the week, and holidays. For the days of the week, the separation, or grouping of one or more days into a single network will be determined. Also holidays will be included to evaluate whether they should or could be incorporated, and if not what to do with them.

After determining the previous aspects of the network, the hidden layer(s) will be reevaluated (if any have been used).

In the first two areas the testing data is randomly chosen from the data set, then starting with section 4, on training the network, the test data will represent forecasts for some time in the future of all data from the corresponding training set.

3.2 GENERAL NETWORK STRUCTURE

The first question to be asked is can an Artificial Neural Network learn to perform the desired application, STLF, and if so what is the most suitable form for the network to take.

To answer this we began with small training sets containing only a sample of seasonal data, and data from weekdays (Monday to Friday). The data sets would be comprised of information on weather, date/time, and load data. Five networks were then designed to be trained and tested with this data for each season. The first four have parallel input and output (24 hours at a time) and evaluate the performance of the hidden layer(s). The use of hidden layers in an ANN can allow for feature extraction, with each hidden neuron representing one feature, which may or may not be obvious when evaluating the problem. In general this allows an ANN to perform better pattern recognition, which is basically the application of load forecasting (extracting the weather–load pattern). The fifth network implements serial (hourly) input and output also using a fixed hidden layer.

The networks (1A–1E) are :

1A. A feedforward network with no hidden layer and 24 output neurons,

1B. A feedforward network with 1 hidden layer of 20[6] neurons and 24 output neurons,

1C. A feedforward network with 2 hidden layers of 20 neurons each (total of 40 hidden neurons) and 24 output neurons,

1D. A feedforward network with 2 hidden layers of 10 neurons each (total of 20 hidden neurons) and 24 output neurons, and

1E. A feedforward network with 1 hidden layer of 20 neurons and 1 output neuron.

Each of these networks will be trained using the same data. The format of the data is presented in table 3.1 for each of the two representations.

	Data Type		Parallel	# Neurons	Serial	# Neurons
Input	Day of the Month	Binary		31		31
Input	Day of the Week	Binary		5		5
Input	Hour	Continuous	Not Used	N/A	An Hour	1
Input	Temperature	Continuous	24 Hours	24	At Hour	1
Input	Wind Speed	Continuous	24 Hours	24	At Hour	1
Input	Sky Cover	Continuous	24 Hours	24	At Hour	1
Output	Load	Continuous	24 Hours	24	At Hour	1

Table 3.1 : The Input and Output Data for the Training and Testing Sets

The main difference is that the serial input (network 1E) requires the knowledge of the hour for which the data applies, while the position of the neuron in the parallel networks (1A–1D) defines the hour that it represents. Each data type is represented as either binary

6. The initial number of hidden neurons (20) is based on work done with Dr. Kermanshahi [18].

(0,1) or continuous [0,1], with only five binary neurons for the day of the week as only weekdays are being considered.

To begin with each network is trained and tested using data chosen from only one season at a time, for which the training and test results are shown in tables 3.2 and 3.3 respectively. The results presented are the average percent error (Avg.) and the daily peak percent error (Peak). These are calculated as follows :

$$\textbf{Percent Error} = \frac{|\textbf{Actual Load} - \textbf{Forecast Load}|}{\textbf{Actual Load}} \times 100\% \tag{16}$$

$$\textbf{Avg.} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{24} \textbf{Percent Error}(i,j)}{n \times 24} \tag{17}$$

$$\textbf{Peak} = \sum_{i=1}^{n} \textbf{Percent Error}(i, \textbf{Peak Load Hour}) \tag{18}$$

where **n** is the number of days in the respective set, and **i, j** reference the day and the hour of the day in question (only one hour being important to the peak calculation).

Network	Winter		Spring		Summer		Fall	
	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
1A	3.6	3.5	3.6	3.0	5.2	4.3	3.9	3.1
1B	2.9	2.8	3.4	2.9	4.1	3.6	3.7	2.9
1C	2.9	2.6	3.6	3.0	4.0	3.6	3.8	2.9
1D	2.9	2.6	3.6	3.0	4.1	3.6	3.8	2.9
1E	3.9	5.6	4.6	5.1	4.6	2.4	4.9	3.6

Table 3.2 : Training Results for Seasonal Training

For each of the networks, the training results were quite good. While the error results for networks 1A and 1E being consistently higher on average, a few of their peak forecasts performed quite well, notably network 1E in the summer. The results for network 1A were perhaps the most surprising, that it did so well without a hidden layer.

Network	Winter		Spring		Summer		Fall	
	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
1A	5.1	2.7	7.6	2.1	5.1	4.8	5.1	2.0
1B	4.3	2.1	6.3	4.3	4.8	4.7	4.8	3.0
1C	4.2	2.1	6.6	4.5	4.9	5.2	5.3	3.2
1D	4.1	2.6	6.6	4.5	4.9	5.2	5.3	3.3
1E	5.1	5.8	8.3	6.8	5.6	4.9	6.1	5.9

Table 3.3 : Test Results for Seasonal Training

The testing results were good with an average close to or below five percent[7], with the exception of network 1E, the network with serial input. Also the spring season had higher average errors for the testing data, then the other seasons. This was expected as spring and fall usually are the hardest to forecast, due to the variation in the weather from day to day. However the fall errors were more similar to the summer season then to the spring season, not as expected.

Network 1A again performed better than expected. It is consistent with 5.1 percent error for the test data (except for the spring season 7.6), and gives excellent peak forecasts. This indicates that for peak forecasting at the seasonal level, the STLF application may not need a hidden layer. However that still leaves the problem of the spring results, and the other features associated with forecasting the daily load.

Networks 1B, 1C, and 1D all performed similarly in both training and testing. However their response to the spring data was also quite poor. This leads to the possibility that seasonal training does not provide sufficient data for the times of change between winter and summer, or that the training set is to small, and the network is memorizing (using each hidden neuron to recognize a particular training pair) instead of generalizing.

Network 1E performed the worst throughout the four seasons, and did not perform the peak forecasting as well as with the training data.

7. Five percent average error is the benchmark for good forecasting at MH.

The next step is to retrain and test these networks with data from all four seasons at one time to try and improve the results (particularly for the Spring). Table 3.4 shows a comparison of the previous testing results, averaged over the four seasons, with the test results for each network trained and tested with the composite annual data set.

Network	With Seasonal Training		With Annual Training	
	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
1A	5.7	2.9	7.3	5.2
1B	5.1	3.5	4.9	3.6
1C	5.3	3.8	5.0	4.1
1D	5.2	3.9	5.0	4.1
1E	6.3	5.9	5.9	6.2

Table 3.4 : Comparison of Test Results Between Seasonal and Annual Training

It can be seen that with annual training, there is a drop in the average error of all of the networks with hidden layer(s), 1B–1D. This reduction in error was mostly in the data corresponding to the spring season.

Network 1A, with no hidden layer had a dramatic increase in error, mostly from the winter and summer seasonal data. This can be explained as an averaging effect over the network with no hidden layer to identify more than one feature, as the peak load is no longer consistent over all of the data as it was with the individual seasonal data. The result is the averaging of high and low peak values, indicated by the dramatic increase in the average peak error.

Networks 1B, 1C, and 1D again had very consistent results, the best of which was network 1B. These results reinforce the idea that a network with more then one hidden layer can be replaced by a network with a single hidden layer of the same total number of hidden neurons in it as in the original network [24]. Furthermore, it indicates that 20 hidden neurons is sufficient (though not necessarily optimal). The improvement in the results can be attributed to the larger composite training set, which now is large enough so that we need not fear memorization of the training data, and for each season the network has knowledge of 'unseasonable' weather / load relationships.

Network 1E shows the most improvement, for similar reasons stated above, it still performs quite poorly (even worse) for the peak hours. Due to this poor peak performance, and continued higher average errors, the serial network configuration does not warrant further investigation at this time.

The general network structure will be that of network 1B, a feedforward ANN with a single hidden layer of 20 neurons, and parallel input and output (24 output neurons). The specific input configuration will be determined in the next section. The training set will be comprised of the composite data set (representative of all seasons), and will be investigated further in section 3.4.

3.3 STRUCTURE OF INPUT LAYER

The structure of the input layer will now be investigated, to determine both what type of data should be used, and how it should be represented. Furthermore the use of feedback data as input to the network will be tried for appropriate types of data.

3.3.1 INPUT TYPES

The basic network configuration that was chosen from section 3.2 is summarized in table 3.5, and will be considered as a base case and used for comparison to other input configurations.

	Data Type		# Neurons
Input	Day of the Month	Binary	31
Input	Day of the Week	Binary	5
Input	Temperature	Continuous	24
Input	Wind Speed	Continuous	24
Input	Sky Cover	Continuous	24
Hidden	N/A	N/A	20
Output	Load	Continuous	24

Table 3.5 : Basic Network Configuration

Table 3.6 (on the next page) lists each variation[8] on the base case, labeled network 2A, that will be considered. Each variation is made one at a time, so that their individual effect may be evaluated. The variations include removing each input type, except temperature whose relationship with load is most prominent, as well as changing the manner of representation of the temperature and wind speed inputs. Also two additional inputs are evaluated, the month of the year (binary), and the hourly windchill (continuous). The windchill is calculated from the temperature and wind speed at a given hour in the manner used by Environment Canada, who currently supply the weather forecasts to MH. The calculation is :

$$\mathbf{WindChill} = 1.1626 \times \left[10.45 + \left(10 \times \sqrt{\mathrm{W}}\right) - \mathrm{W}\right] \times \left[33 - \mathrm{T}\right] \qquad (19)$$

where T is temperature in degrees Celsius, and W is wind speed in meters per second.

From cases 2B, 2D and 2N the training and testing results indicate that a network without the day of the month, the month of the year and the sky cover inputs perform as well or better than the network which include them.

From case 2C the training data indicates that the network could be trained just as well without the day of the week, however the test performance is slightly worse without it.

8. All variations in this section are to the input layer only.

Case	Data Type	# Neurons	Training		Testing	
			Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
2A	Base Case	N/A	3.6	2.9	4.9	4.6
2B	No Day of the Month	0	3.6	3.1	4.6	3.9
2C	No Day of the Week	0	3.6	2.8	5.2	4.7
2D	Add Month of the Year	12	3.6	3.0	4.9	4.5
2E	Degree Days Temperature	48	3.3	2.9	4.3	4.3
2F	Maximum Temperature	1	4.0	3.0	8.3	5.9
2G	Maximum Temperature and Temperature Range	1 1	4.1	3.2	8.1	6.0
2H	Mean Temperature	1	4.1	3.0	8.1	5.8
2I	Mean Temperature and Temperature Range	1 1	4.1	3.2	8.1	6.0
2J	No Wind Speed	0	3.7	3.0	5.2	4.9
2K	Maximum Wind Speed	1	3.7	3.0	5.1	4.9
2L	Add Wind Chill	24	3.4	2.8	4.0	4.5
2M	Add Wind Chill and No Wind Speed	24 0	3.4	2.9	3.8	4.2
2N	No Sky Cover	0	3.5	3.1	4.2	4.4

Table 3.6 : Network Input Configurations and Results for Input Types

From cases 2E, 2F, 2G, 2H, and 2I again both the training and the testing data confirm that for the presentation of temperature input, the full set of daily temperature values, with each hours data represented by 1 (base case 2A) or more (case 2E) neurons is appropriate. When single values representing an hourly temperature (with and without the temperature range as well) the average errors nearly double, with a corresponding increase to the average peak hour error. The manner of presentation in case 2E was to split temperature across two inputs representing 'degree days'[9]. One input represents the temperature for heating days, ($< 20\,^{\cdot}\mathrm{C}$), and one for cooling days ($\geq 20\,^{\cdot}\mathrm{C}$), for each hour of the day.

The training and testing results for cases 2J and 2K indicate that the wind speed has some affect and that as with temperature, hourly presentation is preferred. However as with

9. A reference to the division of days into heating or cooling days employed by MH.

the cases 2C and 2D the differences are not very large. This leads us to investigate other methods of presentation of the wind speed data. One such input is wind chill which combines both wind speed and temperature (this has the added advantage of reinforcing the temperature input). Cases 2L and 2M show that replacing the wind speed input with the corresponding hourly windchill input is the preferred manner of incorporating wind speed data.

3.3.2 FEEDBACK

Feedback is a connection made from a neuron in the network back into the network, so that neurons' output at time t becomes some input at time t+1. The time increments we are using are days, with each input from a given day presented in parallel and the days output made in parallel. Therefore we will use the term feedback to refer to inputs from days previous to the forecast day.

It makes sense at this point to feedback weather data, but not date information, as the feedback of date data would always be the same, (ie. Tuesday always precedes Wednesday). Furthermore, from the previous section it was found that the inputs with the most affect on the network were temperature and wind chill. Therefore these will be attempted as feedback. The other, more commonly recognized, feedback will be from the output, the load. However only one of the hourly loads will be used, in an attempt to give the network a starting point as to the load in the recent past.

Table 3.7 shows the results for training and testing for each case of feedback. Networks 2O–2P show that one or two days previous temperature data improves the network performance the most. Increasing the number of previous days temperature further (2P–2Q) still gives better performance than the base case, but has increasing error.

Case	Data Type	# Neurons	Training		Testing	
			Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
2A	Base Case	N/A	3.6	2.9	4.9	4.6
2O	Temperature 1 past day	48	3.3	2.7	3.7	3.9
2P	Temperature 2 past days	96	3.3	2.8	3.7	3.8
2Q	Temperature 3 past days	144	3.3	2.8	4.1	3.9
2R	Temperature 4 past days	192	3.3	2.8	4.5	4.2
2S	Wind Chill 1 past day	24	3.4	2.7	3.8	4.2
2T	Load from past hour 24	1	3.5	2.9	4.7	4.5
2U	Load from past hour 1	1	3.6	2.9	4.7	4.3
2V	Load from past AM peak	1	3.6	2.9	4.6	4.3

Table 3.7 : Network Input Configurations and Results for Feedback

Network 2S indicates that a windchill feedback may also improve the network. This result may be a reflection of the temperature component of windchill, and if so would probably react as networks 2P–2Q when combined with temperature feedback.

The load feedback was tried for three different hours, 1:00, 24:00, and the AM peak hour. Hour 24 was used to provide continuity to hour 1 of the forecast day, however at forecast time this value would not be known and present another source of error by using the forecast value rather then an actual value. Hour 1 was then tried to see if it could convey the same information. Finally the AM peak was tried as it would seem to be the most appropriate, generally being the most recent known hour at the time of forecast, and giving a consistent value to indicate recent load patterns. From networks 2T – 2V both the training and testing results indicate that load feedback does enhance the network, and that each choice of hour for the feedback provides roughly equivalent results.

3.3.3 COMPOSITION OF RESULTS

From section 3.3.1, it was found that the types of input which lent themselves to this application were temperature (in degree day form), wind chill, and possibly the day of the week.

Composites of these input types will now be considered to determine the best input set. Furthermore due to the inherent relationship between temperature and load, temperature will always be considered.

Table 3.8 presents the summary for the four cases considered :

Case	Input Data Type	Training		Testing	
		Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
2–1	Temperature alone	3.7	3.1	3.6	4.3
2–2	Temperature, and Day of the Week	3.7	3.0	4.2	4.0
2–3	Temperature, and Wind Chill	3.5	2.6	3.4	3.3
2–4	Temperature, Wind Chill, and Day of the Week	3.7	2.7	3.8	3.8

Table 3.8 : Results for Composite Input Variations

The average error for the training set for each of the cases were similar, with the cases using windchill having better results for the daily peak hours. For the test set, network 2–3 outperformed the other networks both overall and for peak hours. Networks 2–2 and 2–4 both using the day of the week input had the highest average errors, but better peak performance then network 2–1 without it. It would seem though when comparing networks 2–3 and 2–4 that day of the week, if included, may possibly become to relied on by the network. That is, conflicting weather and day data will result in chosing the day over the weather for the load forecast, when the weather should win out in influencing the final load.

From section 3.3.2 the feedback to be evaluated with this composite input will be two days previous temperature, the previous days windchill, and the previous days AM peak load. The two days temperature feedback was chosen over the one day feedback due to the observed general performance increase with increased temperature data, and slightly better performance of that network during peak hours. Temperature is again always considered, and combined with each other combination of feedbacks.

Table 3.9 presents the summary for the four cases considered :

Case	Feedback Data Type	Training		Testing	
		Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
2–5	Temperature alone	3.3	2.7	3.4	2.5
2–6	Temperature, and Load	3.3	2.6	3.7	2.8
2–7	Temperature, and Wind Chill	3.3	2.8	3.6	2.7
2–8	Temperature, Wind Chill, and Load	3.3	2.6	3.6	2.7

Table 3.9 : Results for Composite Feedback Variations

All of the networks with feedback have nearly identical training results, better than without feedback. Network 2–5 has the best testing results, with an average error equaling that of network 2–3 without feedback, but with an even better average peak error.

Therefore the input structure of network 2–5 will be chosen, and the modified network structure is shown in table 3.10.

	Data Type		# Neurons
Input	Temperature of the Forecast Day	Continuous	48
Input	Temperature 1 Day before Forecast Day	Continuous	48
Input	Temperature 2 Days before Forecast Day	Continuous	48
Input	Wind Chill of the Forecast Day	Continuous	24
Hidden	N/A	N/A	20
Output	Load	Continuous	24

Table 3.10 : Modified Network Configuration

3.4 TRAINING

With the basic network configuration determined a method of training the network must be devised for the actual application of load forecasting.

For this application the network is to be trained with data from some period before the forecasting period. Testing of the network is then performing forecasting by using input data from the forecasting period, rather then randomly from the entire period. The amount of training data to be used needs to be determined, as well as how often to update the training, and whether or not to use the previously learned weights as an initial weight set when updating the training.

The training data for all cases evaluated in this section will again be limited to weekday data with no holidays[10]. The training set will be chosen from data for upto one year before the forecast date, since that has proven sufficient so far and will have the most similar magnitude to the current (forecast date) system peaks. Each case will be trained so as to perform forecasting on each month of the year for the year 1990.

Four cases will be investigated:

3A Training with a seasonal subset as previously used in sections 1 and 2, updated as new seasonal information becomes available with new random weight[11] sets to start each update,

3B Training with the complete previous years data, updated each month with new random weight sets to start each update,

3C Training with the complete previous years data, updated each month starting each update with the previous months weight set,

3D Training with the complete previous years data, updated each week with new random weight sets to start each update.

10. Holidays are defined in 3.5.2.

11. For cases A, B, and D all initial random weight matrices are the same. For Case C the first random weight set is the same used by the other cases.

The forecasting results are tabulated in table 3.11 for each month, and the year as a whole. The training results are not presented, as they were consistent through each case, with only the results from case A (the seasonal subset) being on average 0.1 % higher average percent error.

	Case A		Case B		Case C		Case D	
	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
January	4.0	3.7	4.4	5.6	4.4	5.6	3.8	5.0
February	22.9	20.7	3.2	3.8	3.3	4.1	3.1	3.8
March	16.5	14.3	3.3	3.4	3.2	3.1	3.4	3.1
April	3.5	3.1	2.8	2.0	2.9	2.0	2.9	2.2
May	7.7	5.9	3.9	4.3	4.0	4.3	3.6	3.6
June	5.5	5.5	4.6	5.6	4.6	5.7	4.4	5.4
July	7.3	4.2	8.5	4.3	8.5	4.3	7.4	4.3
August	11.5	6.4	3.1	3.5	3.1	3.6	3.3	3.3
September	10.0	7.7	3.4	2.7	3.4	2.7	3.3	2.3
October	5.0	6.2	4.3	5.7	4.5	5.4	4.3	4.8
November	4.3	5.2	4.4	5.6	4.3	5.6	4.2	5.4
December	19.4	19.6	5.3	7.8	6.0	8.4	5.2	7.2
Annual	9.7	8.4	4.3	4.5	4.4	4.6	4.1	4.2

Table 3.11 : Forecasting Results for Different Training Structures

For case A the results show that training with a seasonal subset is not adequate for forecasting. This indicates that the seasonal subset is not a good enough representation of the problem.

For cases B and C we see much improved, and far more consistent results. The average error in all months except for July are acceptable. While case C, which used the previous months weight set for initial weight values, did require on average 10 to 15 percent fewer iterations, it did not perform as well as case B, starting from one random weight set.

Case D performed the best overall, but still very close to cases B and C, except for July, where it performed considerably better than these cases (although still not good enough). This then indicates that the weekly training is superior to the monthly training, not so much when the forecast weather / loads follow a predictable pattern, but instead when the pattern is more difficult, as with July.

As well as the July data, the average peak error for December was quite high for each of the cases. This was mainly due to higher peak errors in the days preceding Christmas, and the week between Boxing Day and New Years. These days tend to act as pseudo holidays, with similar daily load curves to normal, but lower daily peaks (by 300–500 MW).

For ease of comparison, the remainder of this chapter will continue to use the monthly training and forecasting, as it is the comparison of different cases, and not actual forecasting which is being compared.

3.5 NETWORK MODULARIZATION

There is a question as to the modular aspect of an ANN, for a particular application. To make an ANN modular is to break up a network into several isolated sub–networks, each of which have the same general architecture, but serve a different purpose.

There are several manners in which a network for load forecasting could be distributed across modular networks. Seasonal networks are one manner of modularization, ie. one sub–network for each season brought together as one network with only the appropriate sub–network active at the same time. This would be similar to combining one of the individual networks trained with only seasonal data from section 3.2. However it was determined that an annual network performed better than the individual seasonal networks.

Furthermore the seasonal information is maintained through the periodic updates as new data becomes available.

In section 3.3.1 the composite results indicated that we could remove the day of the week input, but the individual results indicated that this input had some value. Furthermore, until this point only weekdays have been considered, not weekends. This allows for further evaluation of how to represent data from different days, and groups of days of the week, and the possible exploitation of network modularization.

Other data that has yet to be evaluated are holidays. This again could lead to increased modularization in the network, with a special network for holidays, or perhaps holidays should be included with non–holidays, using an additional input to represent such days.

3.5.1 DAYS OF THE WEEK

There are two general ways to pass information regarding the day of the week to an ANN.

The first method is to directly supply the information to the ANN, in the form of an input. This method has already tried and rejected in section 3.3.3.

The second method is to present the information indirectly. This is done by breaking up the network into several modular networks operating independently, each representing one or more days of the week. Then to forecast for a particular day, non–zero data would be supplied to the network representing that day.

Table 3.12 shows the results (on a seasonal basis) for each network that was tried. These include seven daily networks for each day of the week as a separate network (Monday through Sunday), a network that just combines Tuesday, Wednesday and Thursday data

(TWT), two networks that combine Monday–Friday (weekday) and Saturday–Sunday (weekend), and finally a network for the entire week (weekly). These results are brought together in table 3.13 for comparison of average composite results on a weekly basis.

	Winter		Spring		Summer		Fall	
	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
Monday	3.1	4.5	3.1	1.8	7.9	4.8	5.0	7.1
Tuesday	4.3	4.8	4.9	2.9	7.7	5.5	4.9	6.8
Wednesday	4.8	3.4	4.6	4.0	13.4	8.1	5.3	7.2
Thursday	6.5	8.1	5.5	3.8	11.4	7.0	3.6	5.7
Friday	3.7	3.4	4.9	3.4	8.2	4.0	5.9	7.5
Saturday	3.0	4.1	3.8	2.2	7.4	5.8	5.2	5.7
Sunday	3.0	3.9	3.9	5.6	7.3	4.6	5.5	9.0
TWT	4.6	5.9	3.7	3.1	9.0	5.5	4.5	6.0
Weekday	4.4	5.6	2.8	2.0	8.5	4.3	4.3	5.7
Weekend	2.9	3.6	3.9	3.7	7.3	6.6	4.6	6.9
Weekly	5.1	6.1	6.8	8.9	9.3	8.9	8.3	11.7

Table 3.12 : Forecasting Results for Parallel Day of the Week Networks

There are seven ways of combining this data for weekly results :

Daily, S, S – the average of Monday – Friday, Saturday and Sunday networks,

TWT, S, S – the average of Monday, TWT, Friday through Sunday networks,

Weekday, S, S – the average of the Weekday, Saturday, and Sunday networks,

Daily, Weekend – the average of the Monday – Friday, and Weekend networks,

Weekday, Weekend – the average of the Weekday and Weekend networks,

TWT, Weekend – the average of the Monday, TWT, Friday, and Weekend networks,

Weekly – the average of the weekly network.

	Daily, S, S (%)	TWT, S, S (%)	Weekday, S, S (%)	Daily, Weekend (%)	TWT, Weekend (%)	Weekday, Weekend (%)	Weekly (%)
Winter	4.1	3.9	4.0	3.9	3.9	3.9	5.1
Spring	4.3	3.8	3.1	3.8	3.8	3.1	6.8
Summer	9.0	8.6	8.2	8.6	8.6	8.1	9.3
Fall	5.1	5.0	4.6	4.8	4.8	4.4	8.3
Annual	5.6	5.3	4.9	5.3	5.3	4.8	7.4

Table 3.13 : Composite Results for Average Weekly Performance

The best results were obtained when training the network for weekdays and weekends separately. The weekdays were best trained as a group including data from Monday through Friday, with no day of the week indicator. The weekends produced similar results when trained with both Saturday and Sunday, as when trained separately. The peak hour forecast may be improved upon by separate network, to avoid averaging.

3.5.2 HOLIDAYS

As with the days of the week, holidays can also be represented directly, or indirectly. When represented directly they have an additional input to indicate that the day in question is a holiday. Indirectly, they may be trained in a separate network, with only holiday data. Finally a combination of the two would result in holidays, and non–holidays in modular networks, but with training of the holiday networks including non–holiday data, and an input indicating holidays.

For the holidays three cases were evaluated :

4A The network having a binary holiday neuron, with training data incorporating holidays treated as either weekday or weekend depending on actual occurrence of holiday.

4B The network having a binary holiday neuron, with training data incorporating holidays treated as weekends only no matter when the occurrence of the holiday.

4C The basic network trained exclusively with holiday data.

At present holidays are treated as weekends by MH for STLF, regardless of when they occur during the week. For that reason network 4B is included, to see whether the ANN holiday forecasts are better as weekends, or as they occur in the week (network 4A).

Table 3.14 shows the results for those days chosen as holidays by MH. It should be noted that no distinction is made between statutory and non–statutory holidays, or between holidays which vary in date from year to year (maintaining the same day of the week) and those which occur on the same date each year.

Holiday	Date	Case 4A		Case 4B		Case 4C	
		Avg. (%)	Peak (%)	Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
New Years	Jan. 1, 1990	3.6	0.4	5.3	9.3	4.6	.6
Good Friday	Apr. 13, 1990	2.9	0.3	4.8	2.2	8.1	7.6
Easter	Apr. 15, 1990	9.9	9.0	5.5	5.2	8.9	10.9
Easter Monday	Apr. 16, 1990	9.9	13.8	11.0	15.2	5.8	7.7
Victoria Day	May 21, 1990	4.5	2.8	6.1	2.7	8.6	6.3
Canada Day	Jul. 1, 1990	7.6	1.0	6.1	3.0	9.2	4.3
August	Aug. 6, 1990	3.2	8.4	15.5	11.3	12.4	5.0
Labor Day	Sep. 3, 1990	3.5	0.3	4.9	5.6	5.4	7.9
Thanksgiving	Oct. 8, 1990	3.9	2.2	3.2	1.3	4.1	3.5
Remembrance Day	Nov. 11, 1990	3.6	5.1	4.9	1.7	3.1	5.6
Christmas	Dec. 25, 1990	2.9	6.0	2.6	8.7	8.3	6.8
Boxing Day	Dec. 25, 1990	1.1	0.8	2.1	0.2	14.3	17.2
Average		4.7	4.2	6.0	5.5	7.7	7.0

Table 3.14 : Forecast Results for each Holiday

The results clearly indicate that the networks for case 4A, performs the best for forecasting holidays. These are the weekday and weekend networks, modified to include a holiday neuron, and trained with both holiday and non–holiday data.

Now the question remains of whether modular networks for holiday and non–holiday forecasting is required. Table 3.15 shows the average error and average peak error for each months' weekday and weekend network for which there is a holiday. In each case it is specified whether the network is weekday or weekend, and in the case of April both are used. For proper comparison the average for the networks trained with holiday data is only over the days of the month which were not holidays, rather than over the entire forecast period.

The results indicate that the average error is consistent over both cases, however the average peak error results show that there is a difference over the distribution of the average error. The networks trained with holiday data show average peak errors which are consistently greater than or equal to the networks trained without holiday data, (with the exception of May 4.2% < 4.3%). In particular the difference in the November peaks is unacceptable.

Month	Network	With Holidays		Without Holidays	
		Avg. (%)	Peak (%)	Avg. (%)	Peak (%)
January	Week Day	4.4	5.6	4.4	5.6
April	Week Day	3.0	2.4	2.9	2.0
April	Week End	4.2	3.8	4.0	3.7
May	Week Day	4.0	4.2	4.0	4.3
July	Week End	7.3	6.5	7.3	6.4
August	Week Day	3.2	3.6	3.1	3.6
September	Week Day	3.4	3.1	3.4	2.7
October	Week Day	4.6	6.1	4.5	5.4
November	Week End	4.0	8.5	4.0	3.8
December	Week Day	5.8	8.8	6.0	8.4

Table 3.15 : Forecast Results for Non–Holidays Trained With & Without Holidays

For this reason the holiday and non–holiday networks will be implemented as separate modular networks.

3.6 STRUCTURE OF HIDDEN LAYER

With the network structure in place, as to basic configuration, input types, training method and parallel structures, the original decision to use twenty hidden neurons in the hidden layer will now be analyzed. The weekday and weekend networks were trained and used to forecast for each season of the year (represented now by January, April, July, and October). This was repeated ten times with different initial random weight sets each time. The results were then averaged and normalized. The results are graphed on the following page in figures 3.1 and 3.2 for weekdays and weekends respectively.

Only the non–holiday networks were used, since the holiday networks performed quite similarly to the the non–holiday networks in previous cases.

From these graphs an approximate curve is fit to the plotted data. The data, and the curve, indicate that for both weekday and weekend networks at least four or five hidden neurons are required to produce satisfactory results. Furthermore from the approximate curve it is evident that minimum average error occurs with approximately ten hidden neurons for each network. Therefore ten hidden neurons will be used for the final network configuration.

These results are interesting, since before any actual analysis was performed, simple observation of an annual set of daily load curves yielded at least four recognizable features. These were :

– double AM and PM peak with a valley between them for Summer daily load curves,

– a broad peak continuing between the AM and PM peaks for Winter daily load curves,

– the system peak over Winter loads, and

– the system minimum peak in Spring and Fall.

This reinforces the idea of hidden neurons acting as feature extractors, and that ANN's can extract features that are not readily apparent.

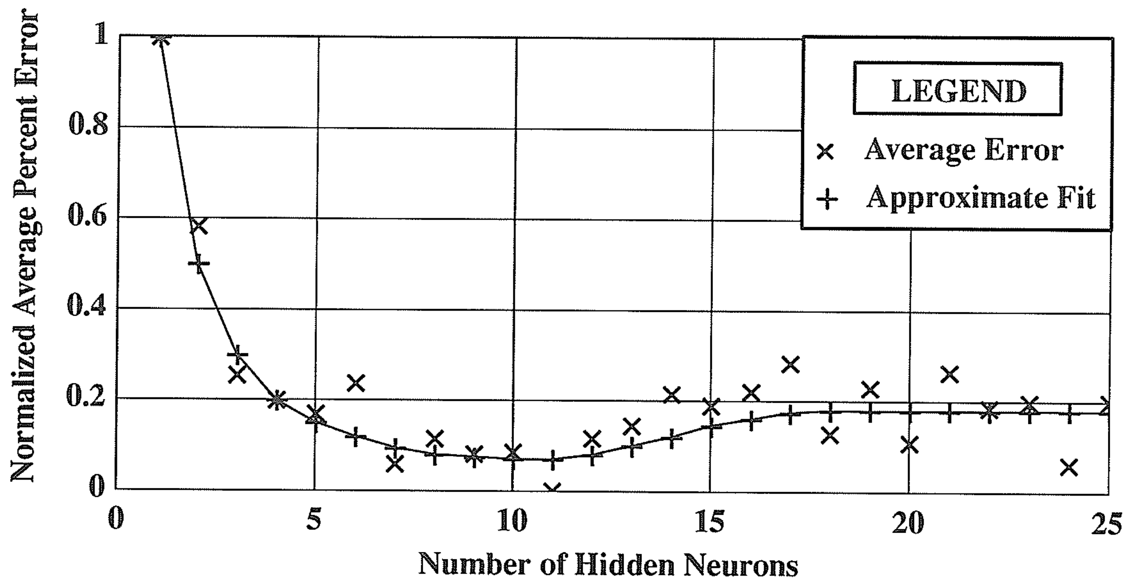


FIGURE 3.1 : Graph of Hidden Neurons vs. Average Error for the Weekday Networks

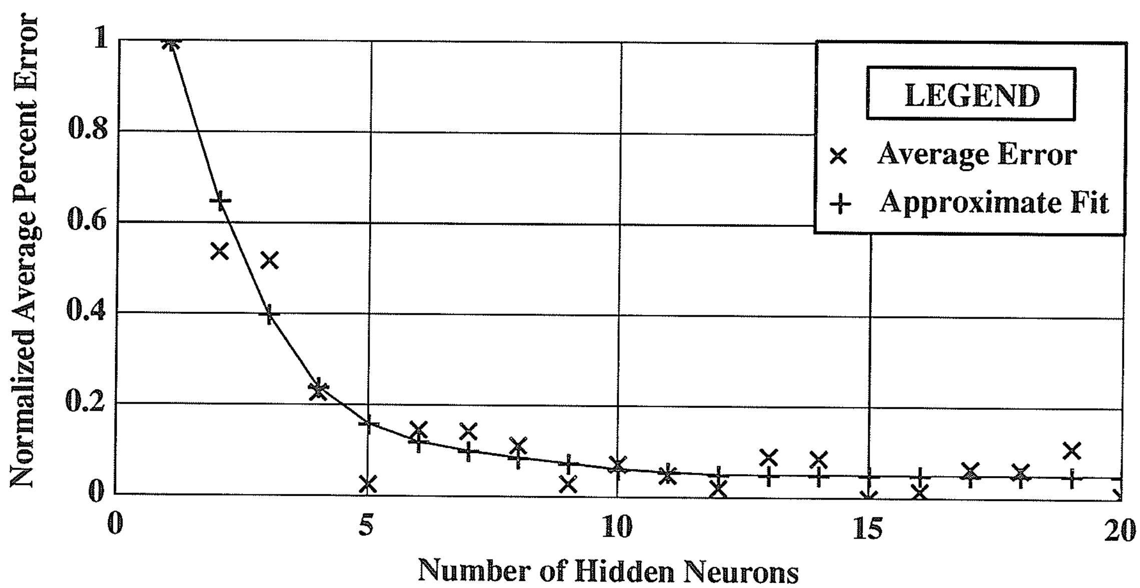


FIGURE 3.2 : Graph of Hidden Neurons vs. Average Error for the Weekend Networks

3.7 SUMMARY

It was quickly determined in section 3.2 that Artificial Neural Networks using supervised learning and feedforward networks is applicable to the problem of short term load forecasting. Furthermore it was apparent that a network featuring parallel input and output for daily load forecasting was the most suitable. Performance for this network was enhanced with a single hidden layer, and that additional hidden layers performed similarly, with no performance improvement in exchange for their added computational burden.

From the basic network structure illustrated in table 3.5, the input structure was evaluated, both adding and subtracting input types[12], as well as the manner in which these types were presented to the network, resulting in the updated network structure of table 3.10.

With the network input determined, the method of training (that is how often, and with what range of data), to perform actual load forecasting was investigated. Training with complete annual data was found to perform much better than training with composite seasonal data, and more frequent retraining was found to give only modest improvements when forecasts were good, but considerable improvement when forecasts were not so good. Also by restarting each training update from a random initial weight set, while taking a little longer to train, found better minimizations for each new forecast period, than starting from the previous final weight set.

The modularization of the network was then evaluated. It was found that rather than one network with the given configuration for all days of the year, that four networks of similar configuration would make up the load forecasting network. These networks are one for weekdays, one for weekends, one for holidays on either weekdays or weekends. The networks for holidays differ only with the addition of an extra input neuron to indicate whether the input is a holiday or not, and that their training data includes past holiday data.

12. This includes feedback, modeled as inputs.

Finally the hidden layer was examined to verify the initial choice of twenty hidden neurons. It was found that the performance could be maintained with a minimum of four to five hidden neurons. However the average (over all seasons, and a variety of initial random weight sets) indicateda need for ten or eleven hidden neurons.

This leads to the final configurations[13] shown in table 3.16 for the non–holiday networks and table 3.17 for the holiday networks. The training for each network can be performed at one week intervals, for forecasting of the following week. The weekday networks may be trained during the weekend before they are to be used, and similarly the weekend networks may be trained during the week (Monday to Friday) before they are to be used. This allows for plenty of time for training the networks in advance of their use, while allowing for frequent updates as indicated.

	Data Type		# Neurons
Input	Temperature of the Forecast Day	Continuous	48
Input	Temperature 1 Day before Forecast Day	Continuous	48
Input	Temperature 2 Days before Forecast Day	Continuous	48
Input	Wind Chill of the Forecast Day	Continuous	24
Hidden	N/A	N/A	10
Output	Load	Continuous	24

Table 3.16 : Final Network Configuration For the Non–Holiday Networks

	Data Type		# Neurons
Input	Temperature of the Forecast Day	Continuous	48
Input	Temperature 1 Day before Forecast Day	Continuous	48
Input	Temperature 2 Days before Forecast Day	Continuous	48
Input	Wind Chill of the Forecast Day	Continuous	24
Input	Holiday	Binary	1
Hidden	N/A	N/A	10
Output	Load	Continuous	24

Table 3.17 : Final Network Configuration For the Holiday Networks

13. These networks will be used for the comparisons to be performed in Chapter 5.

4. COMPARISON OF METHODS

For a comparison of the methods used, one month was chosen to be forecast for Manitoba Hydro from the recent past. The month chosen by MH is May, 1992. This month contains one holiday, Victoria Day.

4.1 MANITOBA HYDRO FORECASTS

As discussed in chapter 2, there is much work that goes into making these forecasts. This involves developing coefficients, chosing a reference day, and then adjusting the forecast. The results for the actual forecasts used by Manitoba Hydro for May 1992, are shown in table 4.1. As well as the overall average percent error (Avg) and the peak hour average percent error (Peak), the results include the overall absolute average load error (Abs Avg), and the peak hour absolute average load error (Abs Peak). The absolute average errors in Mega–Watts, (MW), are given to reflect the magnitude of the errors, and to illustrate the occurrence of seemingly better average forecasts (based on percent error alone). That is, for the same absolute error for two load forecasts (where the actual loads are different), the percent error will be smaller for the forecast corresponding to the larger actual load.

The results are broken up as with the modularity of the ANN's, displaying averages separately for week days, week ends, and holidays, where the holiday (falling on a weekday) is not included in the results for week days. Then the averages for the entire month are shown for overall performance.

	Avg. (%)	Peak (%)	Abs Avg (MW)	Abs Peak (MW)
Week Days	**4.8**	**4.1**	**98**	**95**
Week Ends	**5.9**	**5.2**	**100**	**99**
Holidays	**8.1**	**6.5**	**158**	**148**
Average	**5.2**	**4.5**	**99**	**98**

Table 4.1 : Manitoba Hydro Forecasting Results for May 1992

The chosen month had better than average forecasts for Manitoba Hydro, reflected in the need to adjust only 12[14] of the 31 days after forecasting with the MLR technique. The results of the actual forecasts made, using the reference days chosen by the system user but without adjustment, are shown in table 4.2.

	Avg. (%)	Peak (%)	Abs Avg (MW)	Abs Peak (MW)
Week Days	**5.5**	**4.5**	**108**	**104**
Week Ends	**6.3**	**6.8**	**105**	**130**
Holidays	**7.4**	**6.2**	**145**	**141**
Average	**5.8**	**5.3**	**108**	**114**

Table 4.2 : MLR Forecasting Results for May 1992

It can be seen that the adjustments made by the system user resulted in a small decrease of average percent error of approximately 0.6% overall and 0.8% for peak hours. However that still required the choice of a good reference day.

14. Note that 2 other days had no reference day associated with them (an indication that the MLR forecast was not used at all), but were treated as though they were forecast by the MLR technique.

For just one day, Manitoba Hydro supplied a forecast using a reference day not chosen by an expert user, but by following there guidelines presented in chapter 2. The forecast was made for May 1, 1992, and the results are presented in table 4.3.

The reference day chosen using the guidelines was the Friday from one week before the forecast day, while the expert user chose a reference day of more than one year previous to the forecast day, and from a different day of the week.

Choice of Reference Day	Avg (%)	Peak (%)	Abs Avg (MW)	Abs Peak (MW)
by Expert User	4.6	1.7	77	37
by guidelines	14.2	17.4	260	372

Table 4.3 : Comparison of Choice of Reference Day for May 1, 1992

The results clearly indicate just how important the choice of the reference day is, when a poor choice results in an overall increase of 9.6% error, and a peak error increase 15.7% . This is the risk when an arbitrary reference day is chosen which fits the criteria, the MLR forecast becomes unreliable. This necessitates the presence of an expert to choose an appropriate reference day, with the assumption that the weather forecast is accurate.

4.2 ARTIFICIAL NEURAL NETWORK FORECASTS

Forecasting performed with the ANN uses the method described in Chapter 3. The month of May 1992 was forecast, first with 24 hour weather data, and then with 4 hour weather data (as supplied for the MLR forecast).

The results for the forecast using 24 hour weather data is displayed in table 4.4.

	Avg. (%)	Peak (%)	Abs Avg (MW)	Abs Peak (MW)
Week Days	5.2	4.4	99	99
Week Ends	6.3	6.0	110	120
Holidays	7.1	6.9	126	131
Average	5.6	5.0	103	107

Table 4.4 : ANN Forecasting Results for May 1992 with 24 Hour Weather

The results are very close to those obtained using the MLR technique, with the system users reference day.

At present MH does not receive 24 hour weather forecasts, but instead weather forecast for 4 hours of the day are used[15]. These four hours are 9:00, 12:00, 17:00, and 22:00. These hours correspond to the AM and PM peak ranges as described in Chapter 2. Forecasting for May 1992 was then repeated using these four hours, extrapolating to 24 hour inputs using linear interpolation (as described in Chapter 2 for extrapolating 24 loads). The results for this forecast are presented in table 4.5.

	Avg. (%)	Peak (%)	Abs Avg (MW)	Abs Peak (MW)
Week Days	5.5	4.7	104	105
Week Ends	6.5	6.8	115	148
Holidays	8.5	9.1	145	173
Average	5.9	5.5	108	124

Table 4.5 : ANN Forecasting Results for May 1992 with 4 Hour Weather

There is a small (0.3%) increase in the overall error using 4 hour weather, however it still performs at near to the same level as the ANN with 24 hour input. This increase comes mainly from the weekends and holidays, and particularly for the peak hour, as opposed to the overall average. This would imply that there is a relationship between the daily weather, and the individual (hourly) load forecasts. One reason for the decline in performance could

15. Note that forecast not actual weather is being used.

be the method for extrapolation is not a good enough method, perhaps the use of a 4 to 24 ANN for the weather extrapolation would work better. However the most effective remedy would be to obtain 24 hour weather forecasts, rather than the representative 4 hour forecasts, and eliminate the need for extrapolation.

For each of the ANN's the results for forecasting of May 1, 1992, are presented in table 4.6 for comparison to the forecasts by Manitoba Hydro with good and arbitrary reference days.

Choice of Reference Day	Avg. (%)	Peak (%)	Abs Avg (MW)	Abs Peak (MW)
24 hour ANN	3.3	1.0	56	22
4 hour ANN	3.9	1.1	67	23

Table 4.6 : Comparison of Choice of Reference Day for May 1, 1992

While for the particular day, the ANN forecast outperformed the MLR forecast, what is most important to note is the comparison to the MLR forecast with an arbitrary reference day. The ANN does not require a reference day, and therefore its' results are more consistent, not being effected by a poor reference day. Furthermore the ANN does not require the system operator to spend valuable time in determining an appropriate reference day.

4.3 SUMMARY

The month of May, 1992, was chosen by Manitoba Hydro for a comparison of forecasting techniques. This month was well forecast by the MLR technique, as fewer than half the forecast days needed to be adjusted by the system operator.

The results[16] indicated that when the system operator chose a suitable reference day that the MLR forecasts were approximately equivalent to the forecasts made using the Artificial Neural Networks. The ANN using 24 hour weather information forecast on average 0.3% better than the ANN using 4 hour (extrapolated to 24 hour) weather data.

Figure 4.1 shows a graph of the actual daily load curve for May 1992. Also graphed are three forecasts,

– the MLR forecast using an arbitrary reference day following the MH guidelines for choosing a reference day,

– the MLR forecast using the reference day chosen by the system operator, and

– the ANN forecast.

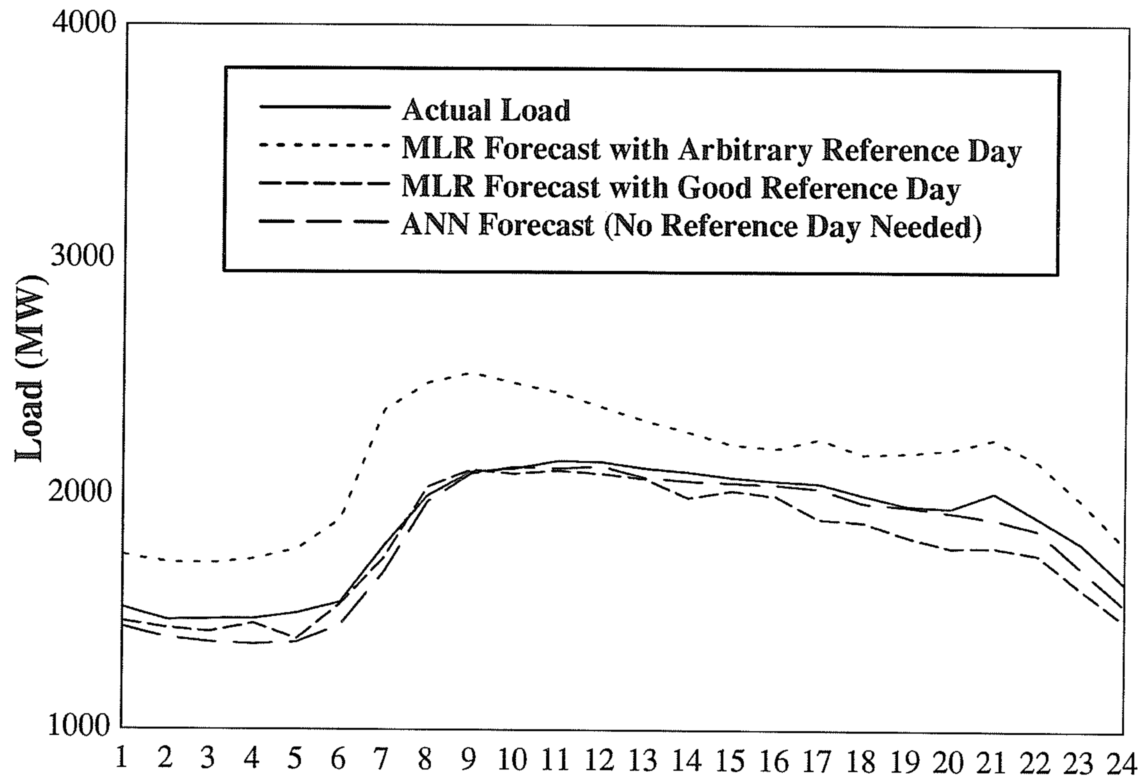


FIGURE 4.1 : Comparison of Daily Load Forecasts for May 1, 1992

This graph reinforces the importance of choosing a good reference day, with errors as high as 32% for individual hours (7:00), an arbitrary reference day provides unreliable forecasts

16. Daily load curves are graphed in Appendix A.

using MLR. This is one of the main advantages of using an ANN to forecast STLF, as it does not rely on a reference day.

A further advantage of the ANN for STLF is depicted by the decline in performance from the forecasts using 24 weather data to the forecasts using 4 hour weather data. This illustrated the reliance of a single hours load on not just the weather at the corresponding hour, but on the daily weather. This relationship is overlooked by the MLR technique.

5. CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSIONS

In this investigation it was found that Artificial Neural Networks are well suited to the application of short term load forecasting.

It was found that ANN's have the ability to incorporate both weather/load patterns and time sequence patterns to perform STLF. It was found that the best forecasts are performed when weather data is provided directly (as numerical input), and time sequence data indirectly (position of input or output indicates hour or day for particular data).

It was also determined that for the three basic types of weather input, that only temperature has a direct affect, with performance improvements by neglecting sky cover, and using wind chill (a function of wind speed and temperature), rather then using the wind speed.

While numerical data representing day of the week information was found to not offer any benefit, it was found that separating weekday and weekend data into modular networks gave the better performance overall. Furthermore for the daily peak forecasts of weekends, it is worth investigating two separate networks, for Saturdays and Sundays, (but this may require more than a single years training data to be sufficient).

This modularization was also used to forecast holidays, but with the inclusion of a holiday input, to differentiate holidays from non–holidays during training.

For Manitoba Hydro the optimal feedforward networks with supervised learning that were determined are described in tables 3.16 and 3.17. Under actual forecasting conditions for the month of May 1992, it was found that the ANN forecast (with average error of 5.9%, or 108 MW) performed nearly as well[17] as the experienced user (with average error of 5.2%, or 99 MW), and very close to the MLR (with average error of 5.8%, or 108 MW).

Even though in direct comparison the performance of the ANN's were not much different from the forecasting employed by MH[18], they have the advantage of being easily automated, adaptable, consistent, and perform this well with no adjustment or input from the system user, (such as choosing a good reference day).

5.2 RECOMMENDATIONS

For the application of ANN's to STLF using supervised learning, two main network structures were evaluated. These are parallel input and output, and serial input and output. The serial network did not perform very well in comparison to the parallel network, however the use of serial load with unsupervised learning[19], parallel input was found to work. Therefore one area of supervised learning that should be investigated, is the network structure of parallel input and serial output.

17. Note that all ANN forecasts are performed without benefit of industrial load data.
18. See daily load curves Appendix A.
19. See Appendix C.

For training the ANN, the backpropagation algorithm with conjugate gradient minimization was used. This algorithm was used so as to train a network with hidden units, in a reasonable amount of time. Training is performed to a certain point, to balance how well the training set is learned, with the ability to generalize. Both the learning algorithm, and the point to which the network should be trained, for optimal forecasting, should be further investigated.

Forecasting for the holidays should also be investigated further. Holidays might be classified into more then one type of holiday input, for example statutory and festive. Also the use of a continuous neuron for holiday input rather than a binary neuron may allow for better forecasts of days which are pseudo holidays. These might be in–service days for schools, festival days in individual towns and regions, or the days preceding and following holidays, (in particular the week between Christmas and New Years).

Other recommendations are to extend the weather forecast data from 4 to 24 hours, and to investigate other weather data types. Other weather inputs that may be helpful to short term load forecasting are humidity, precipitation (perhaps type of precipitation), and wind direction.

REFERENCES

[1] J.H. Park, Y.M. Park K.Y. Lee, "Composite Modeling for Adaptive Short–Term Load Forecasting," IEEE Transactions on Power Systems, Vol. 6, No. 2, pp. 450–456, May 1991.

[2] A.D. Papalexopoulos and T.C. Hesterberg, "A Regression–Based Approach to Short–Term System Load Forecasting," IEEE Transactions on Power Systems, Vol. 5, No. 4, pp.1535–1544, November 1990.

[3] I. Moghram and S. Rahman, "Analysis and Evaluation of five Short–Term Load Forecasting Techniques," IEEE Transactions on Power Systems, Vol. 4, No. 4, pp.1484–1491, October 1990.

[4] N. Hubele and C. Cheng, "Identification of Seasonal Short–Term Load Forecasting Models Using Statistical Decision Functions," IEEE Power Engineering Society, Summer Meeting, 1989.

[5] P. Bolzern and G. Fronza, "Role of Weather Inputs in Short–Term Forecasting of Electric Load," Electric Power and Energy Systems, Vol. 8, No. 1, pp. 42–46, January 1986.

[6] S. Rahman and R. Bhatnagar, "An Expert System Based Algorithm for Short Term Load Forecast," IEEE Transactions on Power Systems, Vol. 3, No. 2, pp. 392–399, May 1988.

[7] K. Ho and Y. Hsu, C. Chen, T. Lee, C. Liang, T. Lai and K. Chen, "Short Term Load Forecasting of Taiwan Power System Using A Knowledge–Based Expert System," IEEE Transactions on Power Systems, Vol. 5, No. 4, pp. 1214–1220, November 1990.

[8] S. Chen, D.C. Yu, A.R. Moghaddamjo, "Weather Sensitive Short–Term Load Forecasting Using Nonfully Connected Artificial Neural Networks," IEEE Power Engineering Society Summer Meeting, Paper No. 91 SM 499–9 PWRS, July 1991.

[9] D.C. Park, M.A. El–Sharkawi, R.J. Marks II, L.E. Atlas and M.J. Damborg, "Electric Load Forecasting Using an Artificial Neural Network," IEEE Transactions on Power Systems, Vol. 6, No. 2, pp. 442–448, May 1991.

[10] C.N. Lu, H.T. Wu and S. Vemuri, "Neural Network Based Short Term Load Forecasting," IEEE Power Engineering Society Winter Meeting, Paper No. 92 WM 125–5 PWRS, January 1991.

[11] T.M. Peng, N.F. Hubele and G.G. Karady, "Conceptual Approach to the Application of Neural Network for Short–Term Load Forecasting," Proc. IEEE International Symposium on Circuits and Systems, Vol. 4, pp. 2942–2945, 1990.

[12] T.S. Dillon, S. Sestito and S. Leung, "Short Term Load Forecasting Using an Adaptive Neural Network," Electrical Power and Energy Systems, Vol. 13, No. 4, pp. 186–192, August 1991.

[13] T.M. Peng, N.F. Hubele and G.G. Karady, "Advancement in the Application of Neural Networks for Short–Term Load Forecasting," Transactions on Power Systems, Vol. 7, No. 1, pp. 250–257, February 1992.

[14] W.M. Grady, L.A. Groce, T.M. Huebner, Q.C. Lu and M.M. Crawford, "Enhancement, Implementation, and Performance of an Adaptive Short– Term Load Forecasting Algorithm," Transactions on Power Systems, Vol. 6, No. 4, pp. 1404–1410, November 1991.

[15] D.C. Park, M.A. El–Sharkawi and R.J. Marks II, "An Adaptively Trained Neural Network," IEEE Transactions on Neural Networks, Vol. 2, No. 3, pp. 334–345, May 1991.

[16] K. Ho, Y. Hsu and C. Yang, "Short Term Load Forecasting Using A Multilayer Neural Network with an Adaptive Learning Algorithm," Transactions on Power Systems, Vol. 7, No. 1, pp. 141–149, February 1992.

[17] K.Y. Lee and J.H. Park, "Short–Term load Forecasting Using an Artificial Neural Network," Transactions on Power Systems, Vol. 7, No. 1, pp. 124–130, February 1992.

[18] B.S. Kermanshahi, C.H. Poskar, G. Swift, P. McLaren, W. Pedrycz, W. Buhr and A. Silk, "Artificial Neural Networks for Forecasting Daily Loads of A Canadian Electric Utility," Proc. of the 2nd Int. Forum on App. of Neural Networks to Power Systems, Yokohama, Japan, pp. 302–307, April 1993.

[19] B. Abraham and J. Ledolter, Statistical Methods for Forecasting, New York, NY: John Wiley & Sons, Inc., 1983.

[20] D.E. Rumelhart, J.L. McClelland and the PDP Research Group, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations, Cambridge, MA: The MIT Press, 1986.

[21] D.O. Hebb, Organization of Behavior, New York, NY: John Wiley & Sons, Inc., pp. 60, 70, 1949.

[22] J.L. McClelland and D.E. Rumelhart, Explorations in Parallel Distributed Processing, Cambridge, MA: The MIT Press, 1988.

[23] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes in C, Cambridge, MA: Cambridge University Press, 1988.

[24] D.E. Rumelhart, J.L. McClelland and the PDP Research Group, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 2: Psychological and Biological Models, Cambridge, MA: The MIT Press, 1986.

[25] T.S. Dillon, K. Morsztyn and K. Phua, "Short Term Load Forecasting Using Adaptive Pattern Recognition and Self Organising Techniques," Proc. of the 5th PSCC, pp. 1–16, 1975.

[26] T. Kohonen, Self Organization and Associative Memory Third Edition, Berlin: Springer–Verlag, 1989.

[27] J.C. Bezdek, E.C. Tsao and N.R. Pal, "Fuzzy Kohonen Clustering Networks," Proc. of the 92 IEEE International Conferance on Fuzzy Systems, San Diego, California, pp.1035–1043, 1992

APPENDIX A :

DAILY PLOTS OF LOAD FORECAST AND ERROR

DATA FOR MAY 1991

FIGURE A1 : Forecast and Error for May 1, 1992

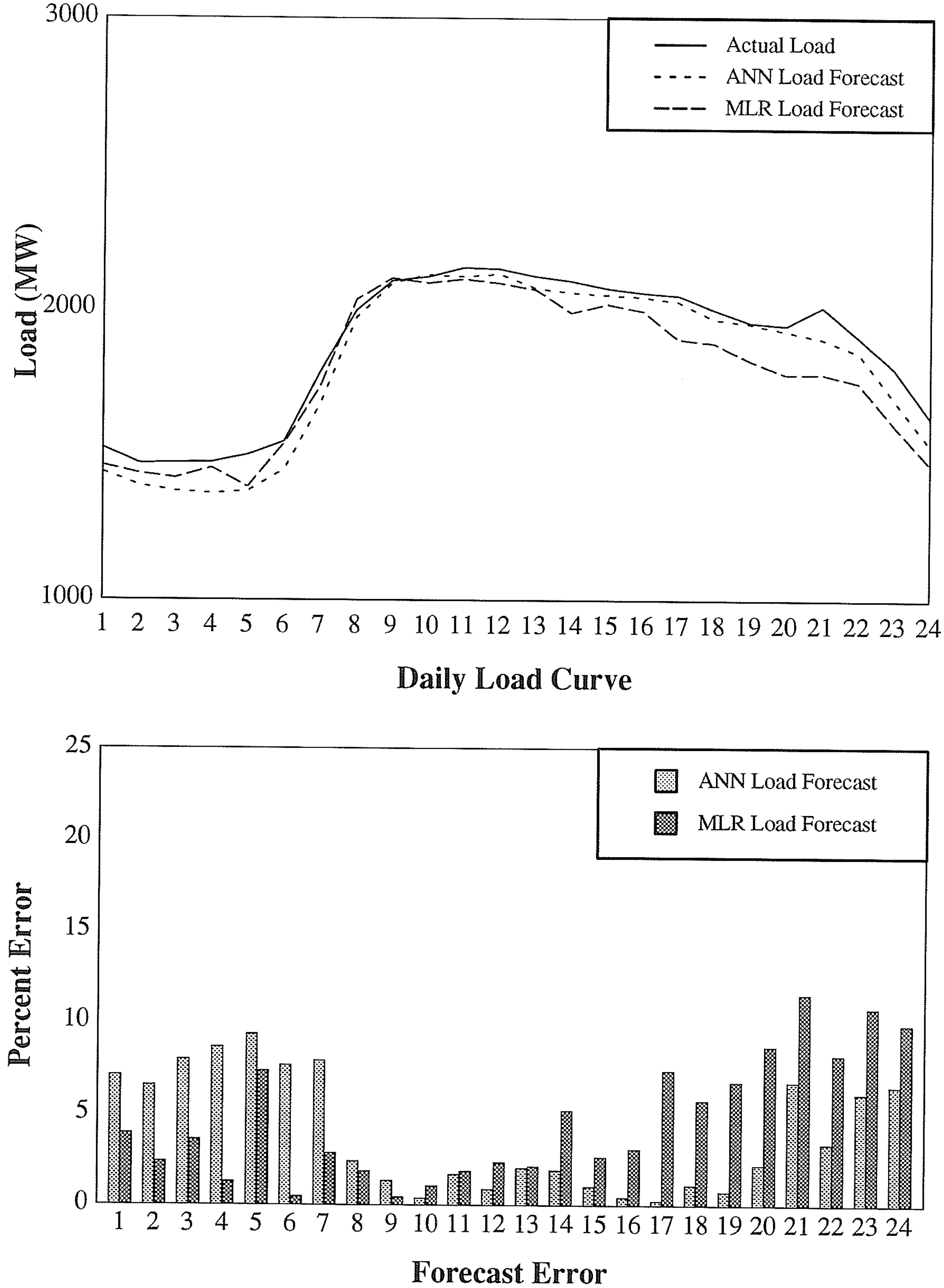


FIGURE A2 : Forecast and Error for May 2, 1992

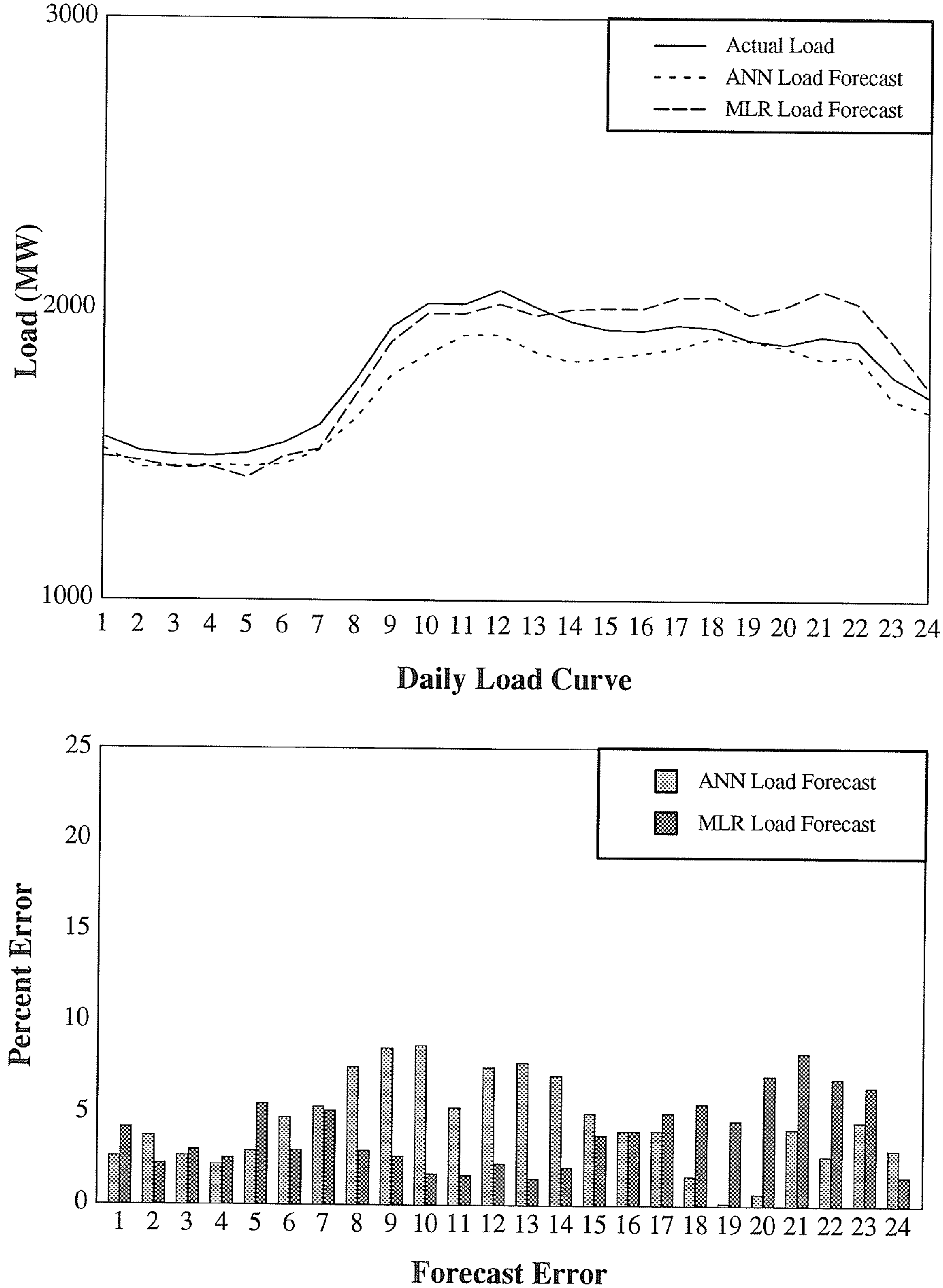
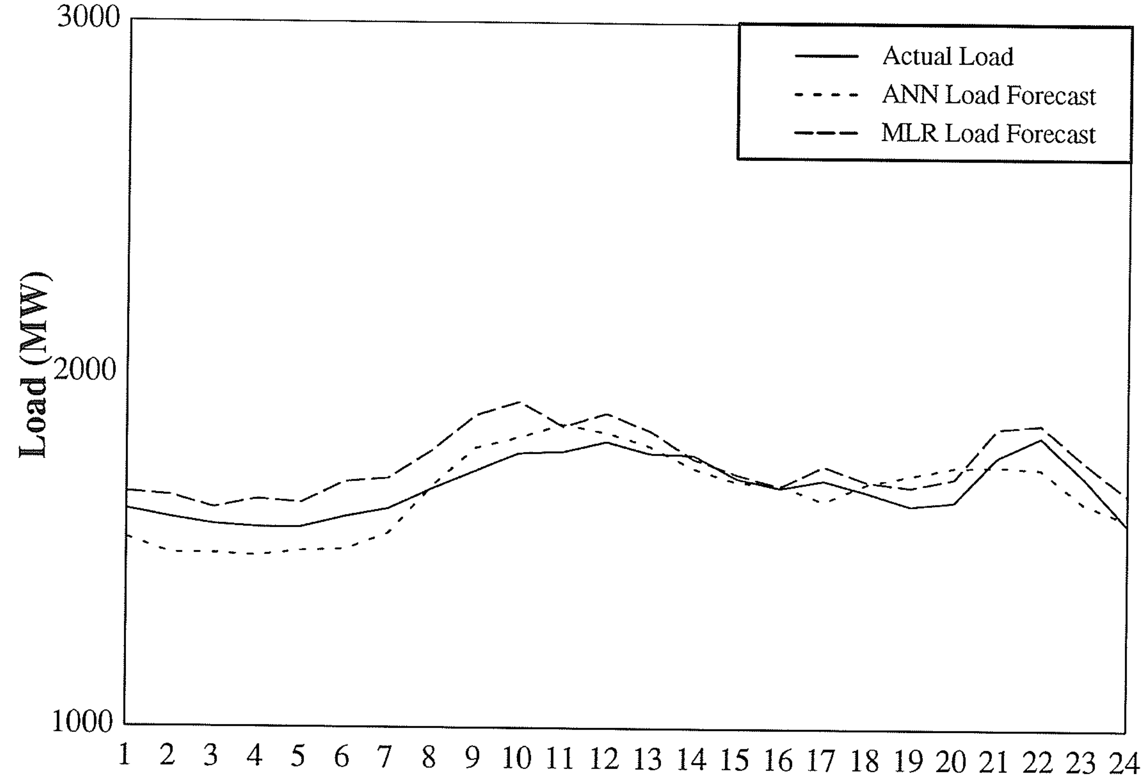
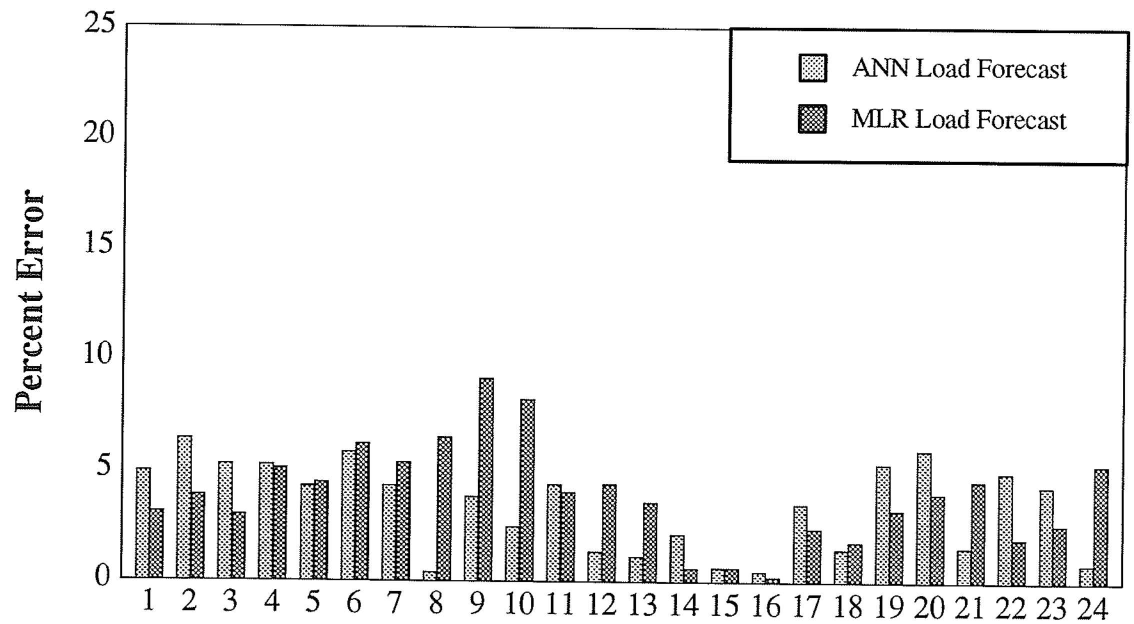


FIGURE A3 : Forecast and Error for May 3, 1992

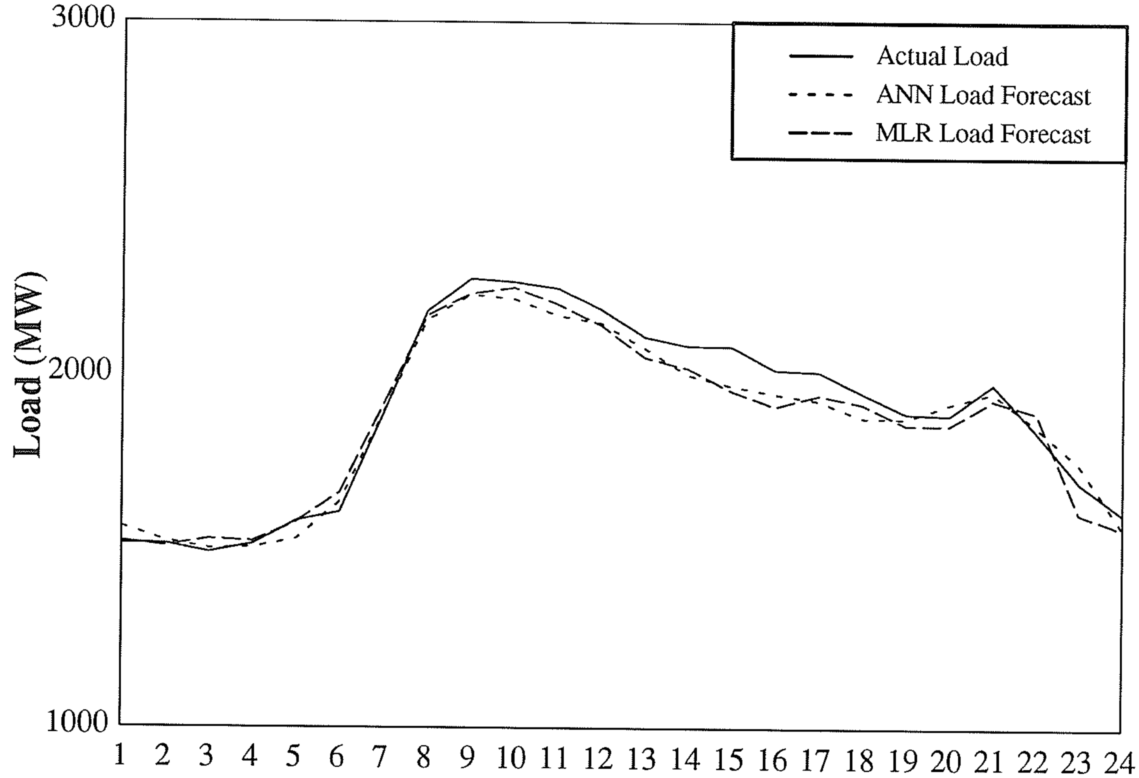


Daily Load Curve

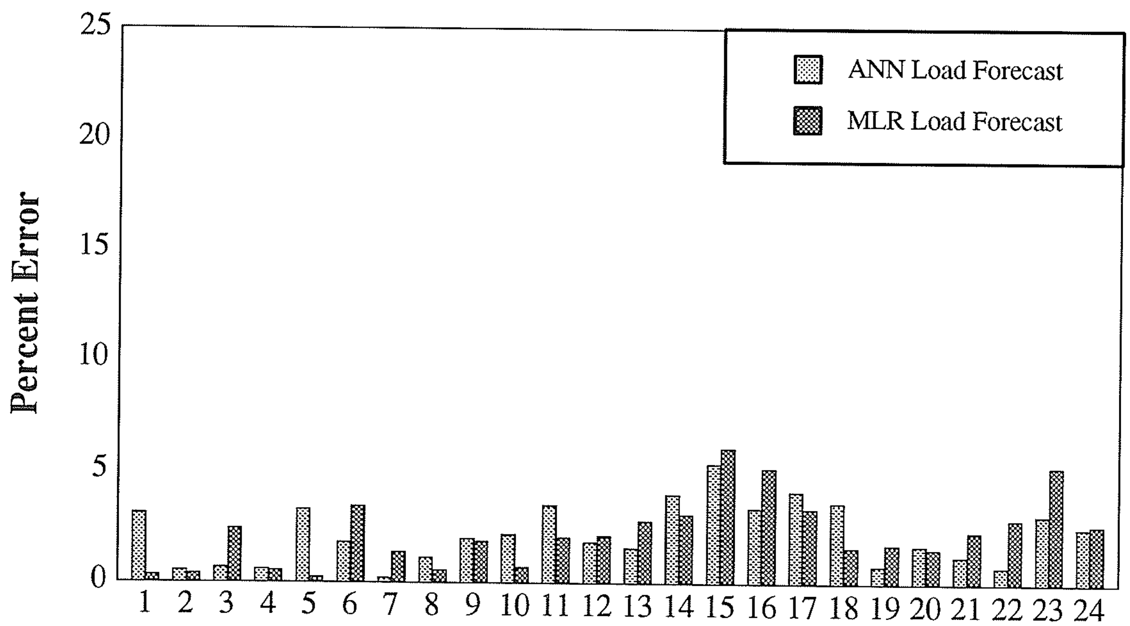


Forecast Error

FIGURE A4 : Forecast and Error for May 4, 1992

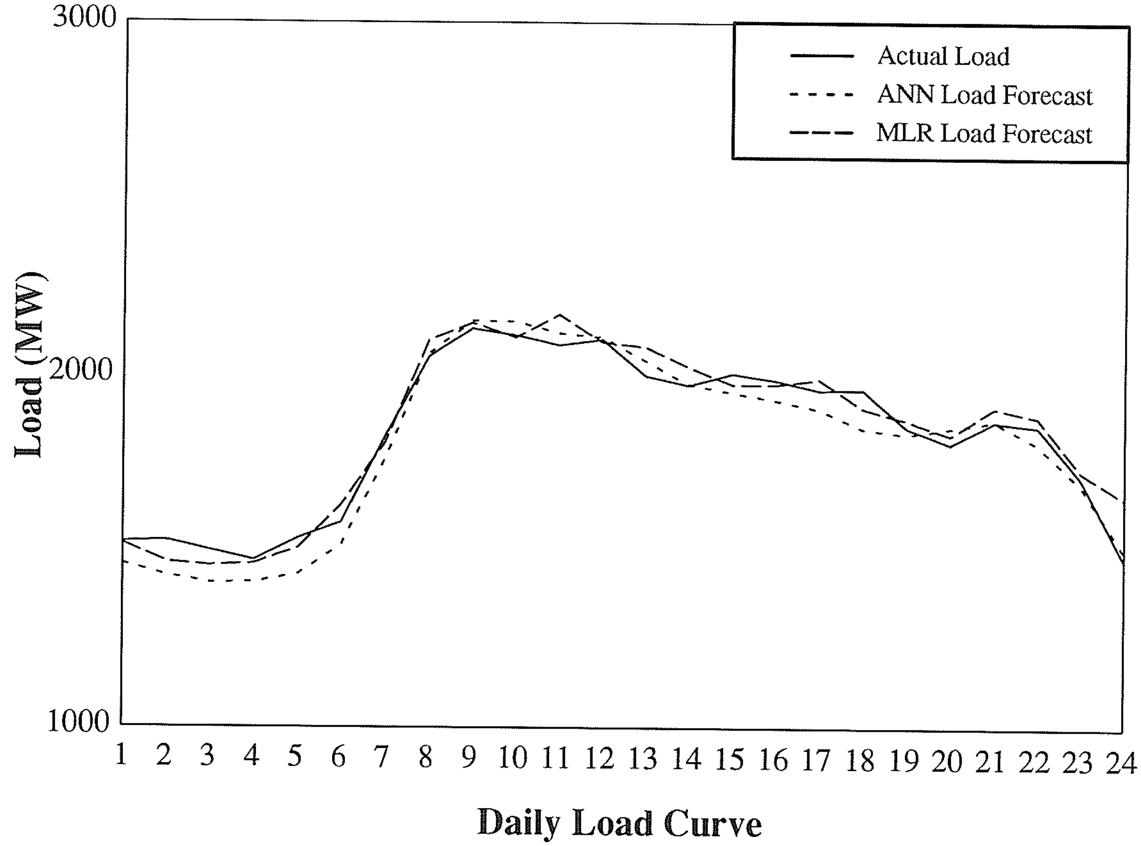


Daily Load Curve



Forecast Error

FIGURE A5 : Forecast and Error for May 5, 1992



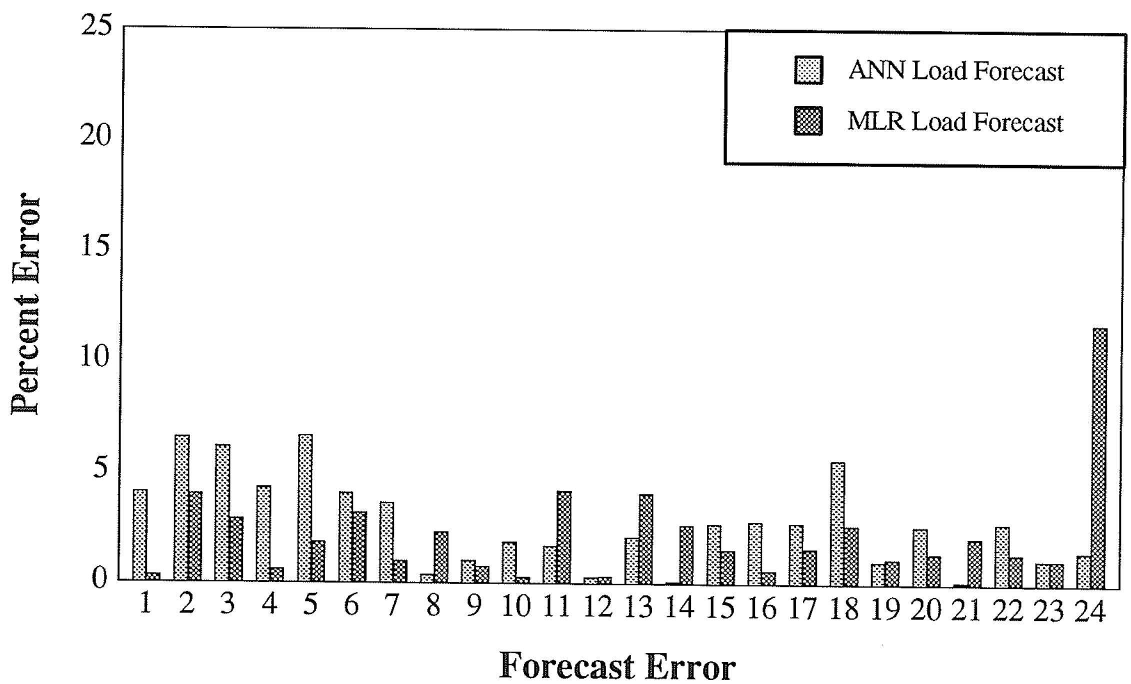
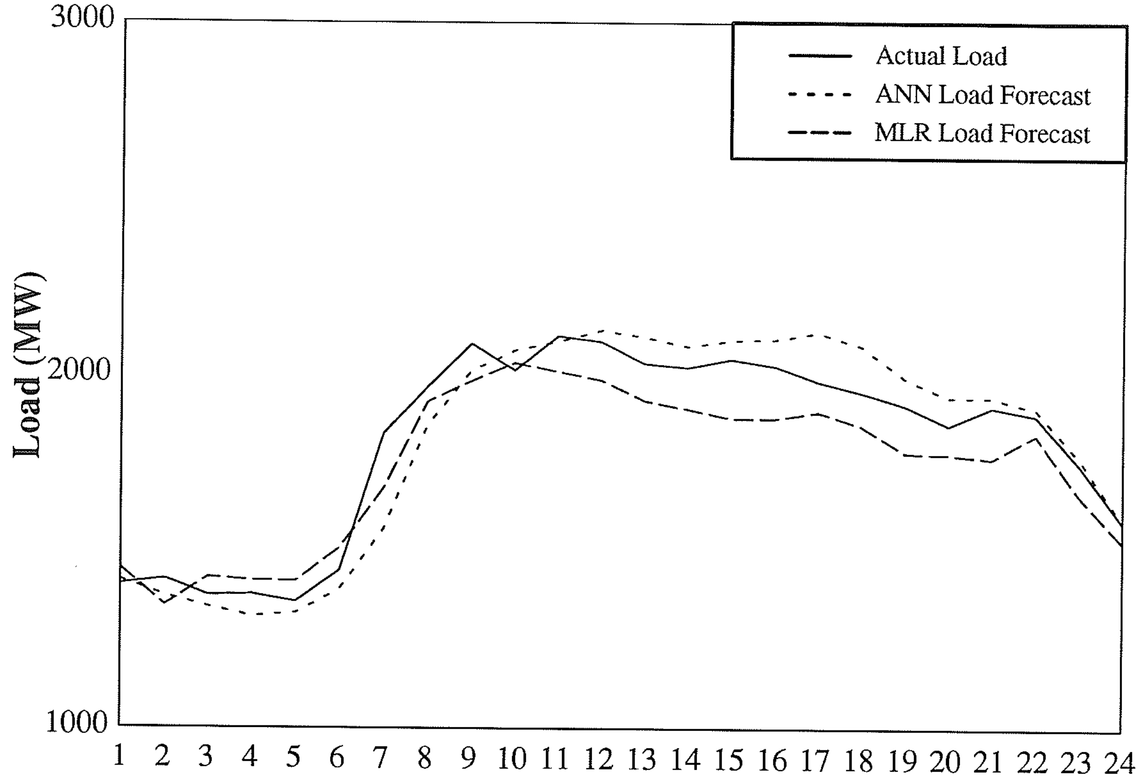
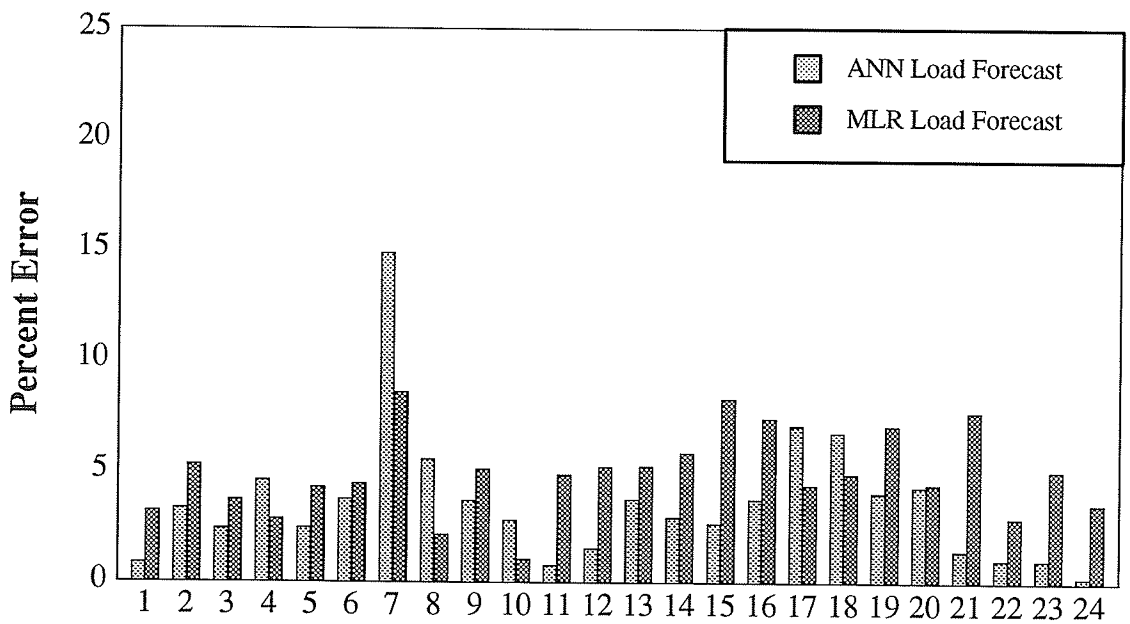


FIGURE A6 : Forecast and Error for May 6, 1992

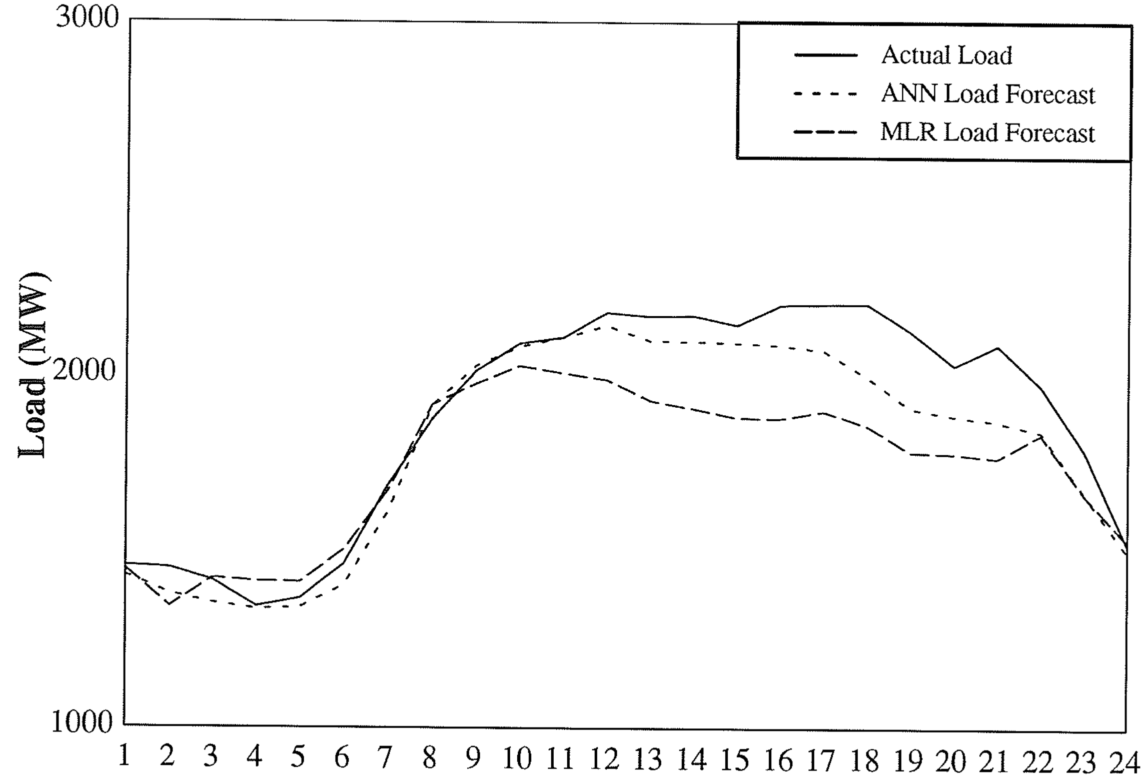


Daily Load Curve

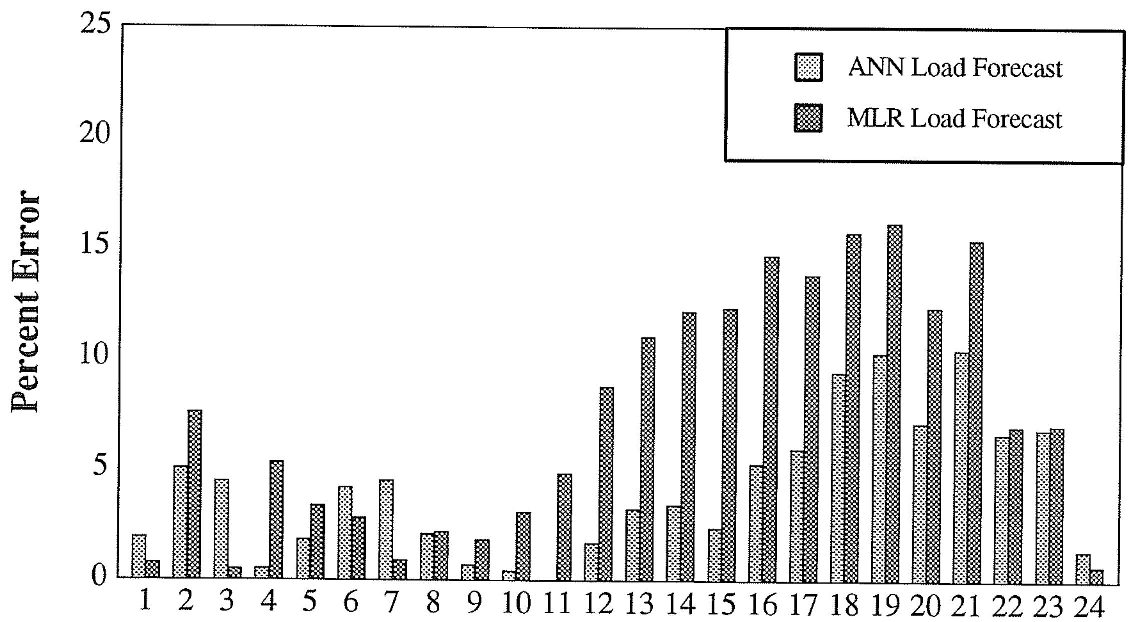


Forecast Error

FIGURE A7 : Forecast and Error for May 7, 1992

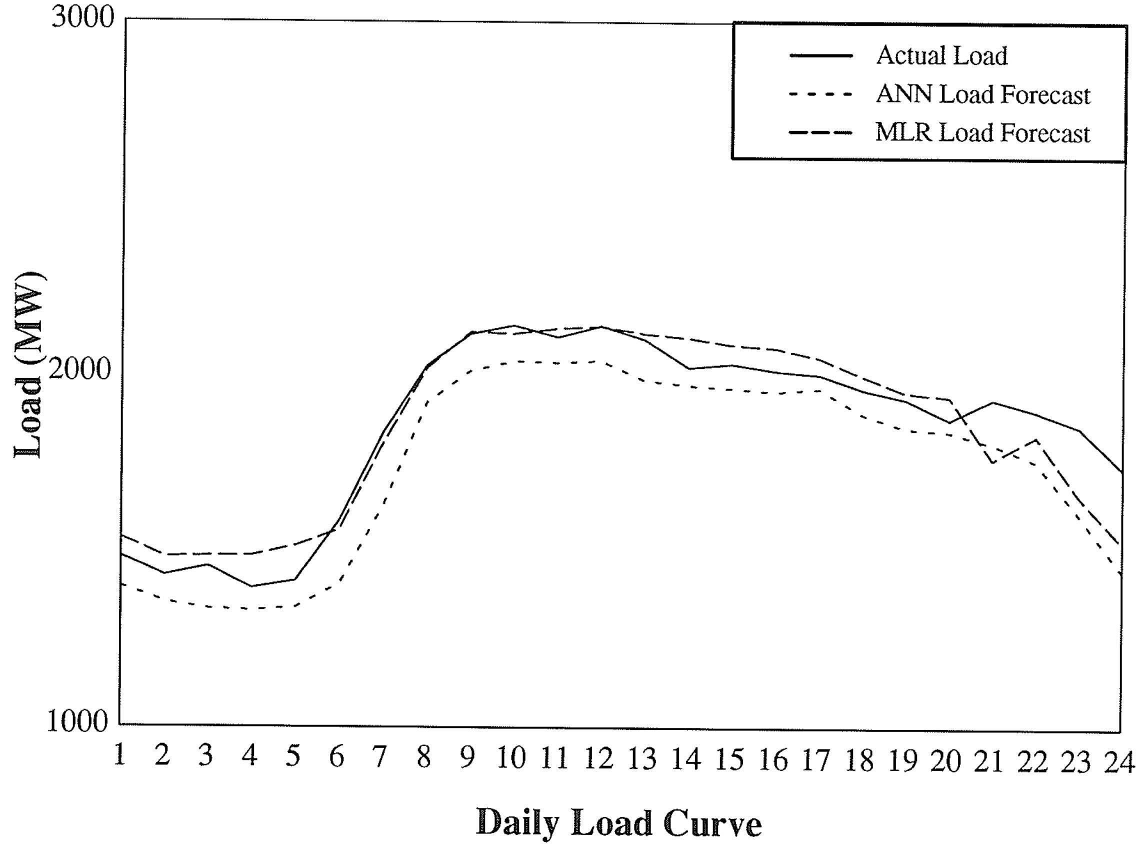


Daily Load Curve



Forecast Error

FIGURE A8 : Forecast and Error for May 8, 1992



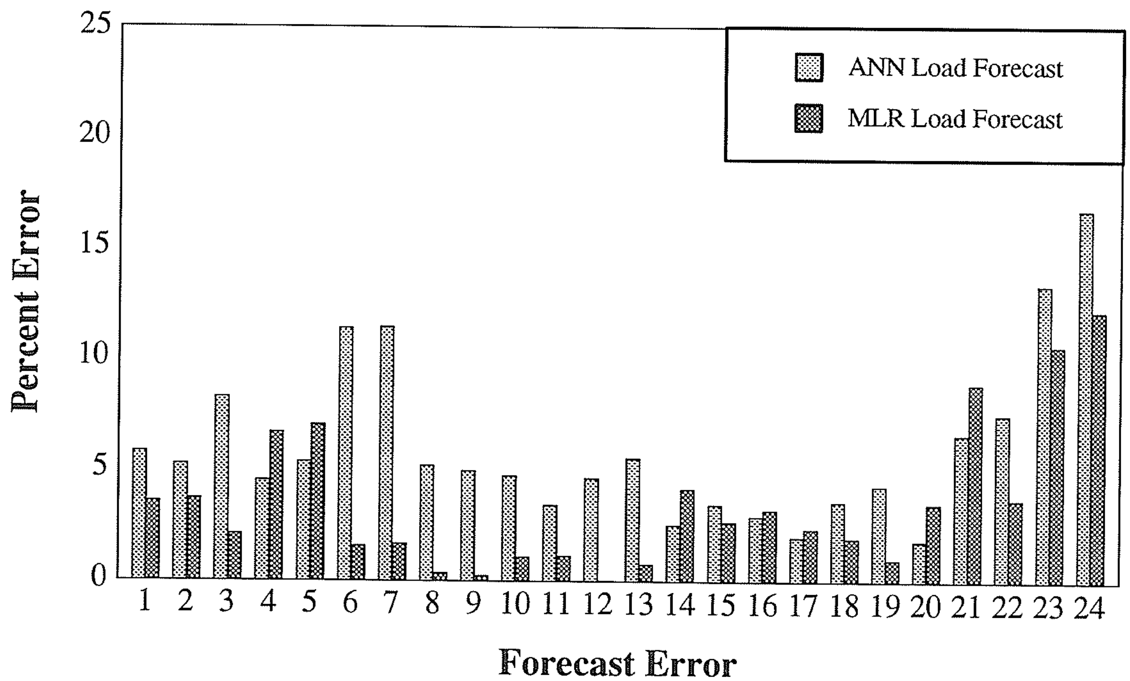
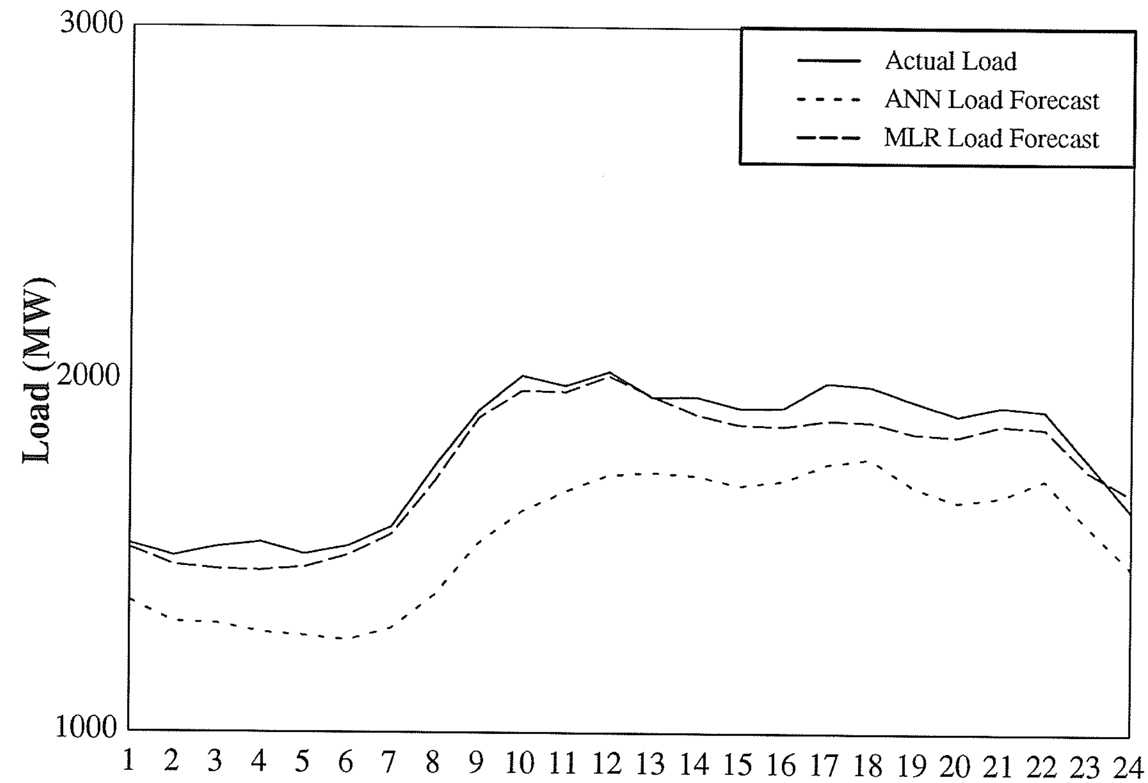
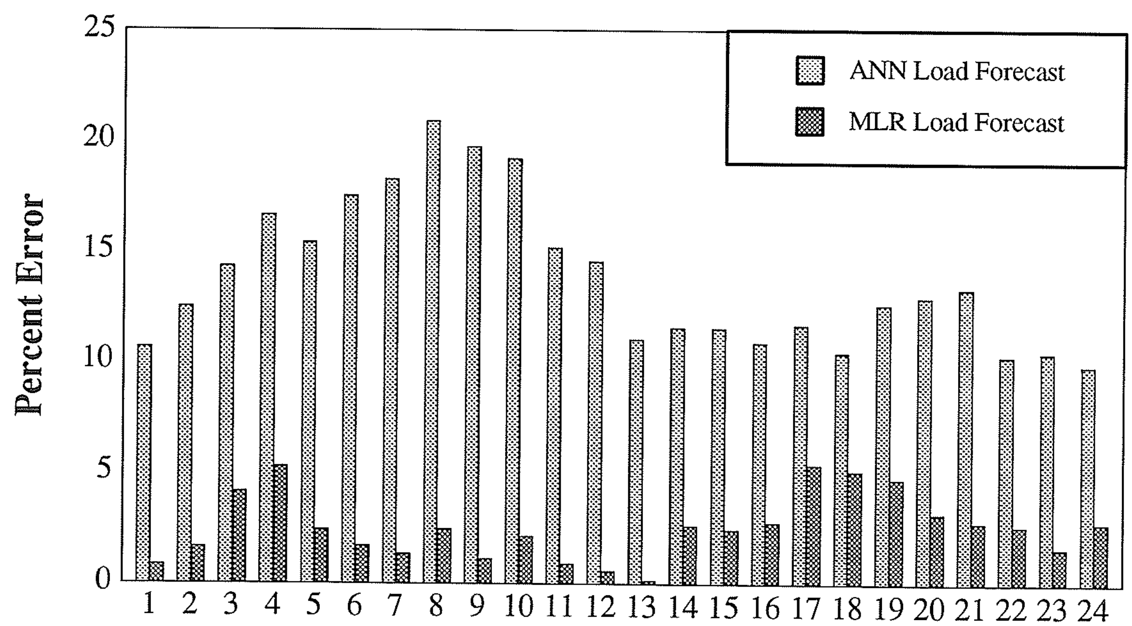


FIGURE A9 : Forecast and Error for May 9, 1992



Daily Load Curve



Forecast Error

FIGURE A10 : Forecast and Error for May 10, 1992

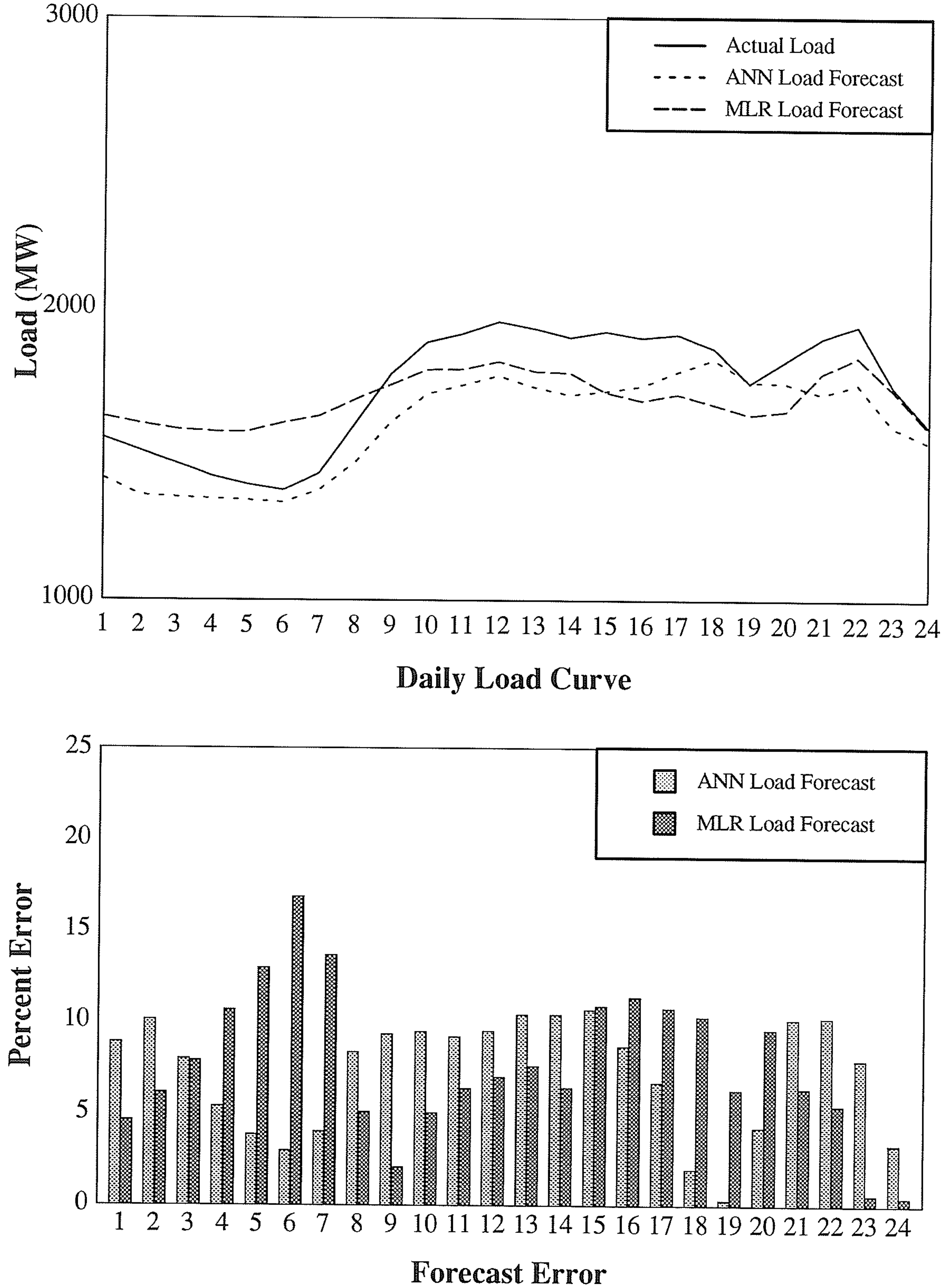
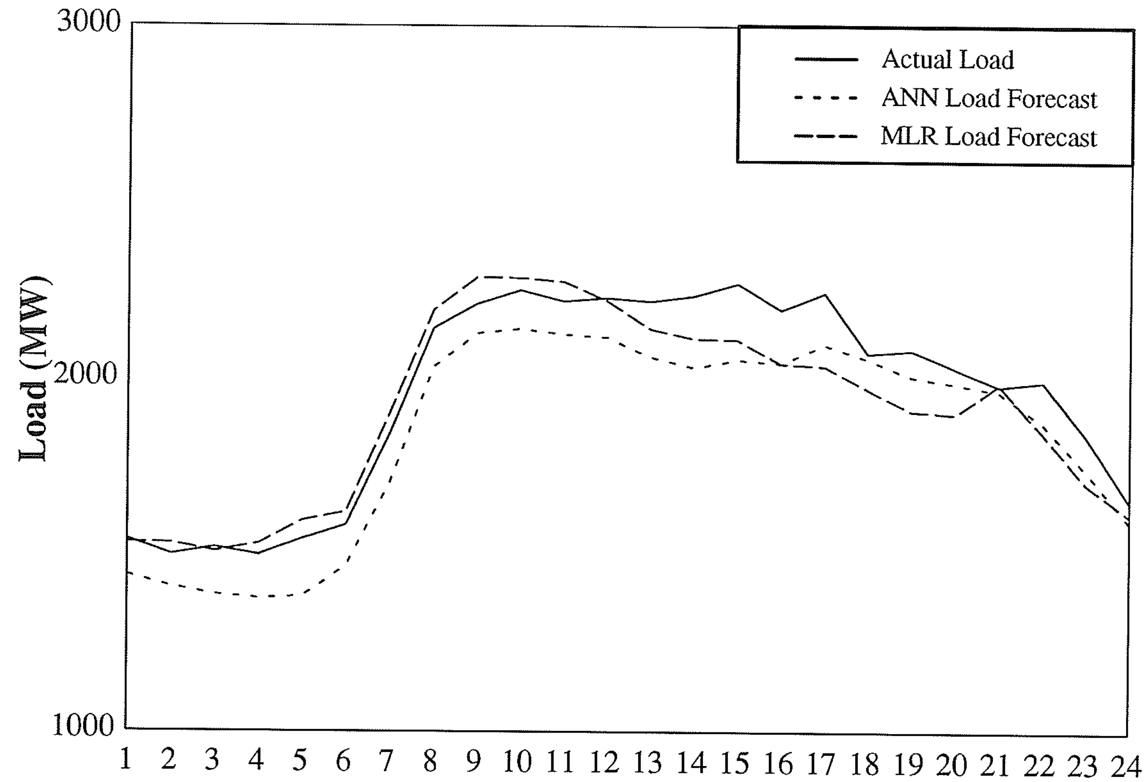
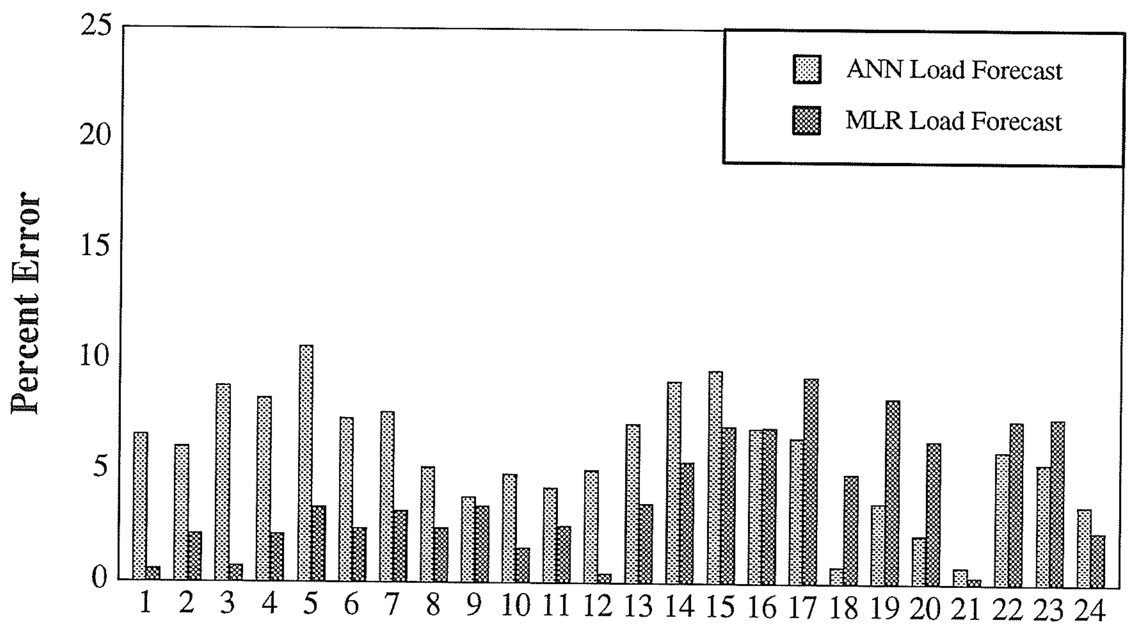


FIGURE A11 : Forecast and Error for May 11, 1992

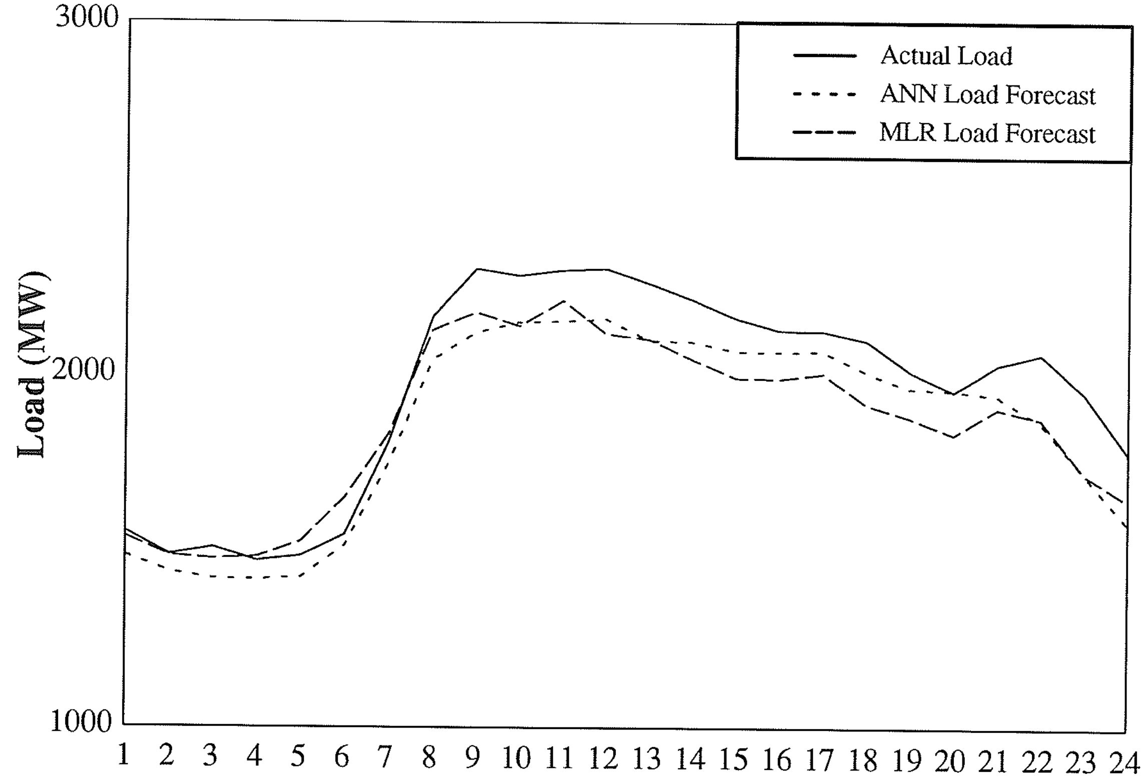


Daily Load Curve

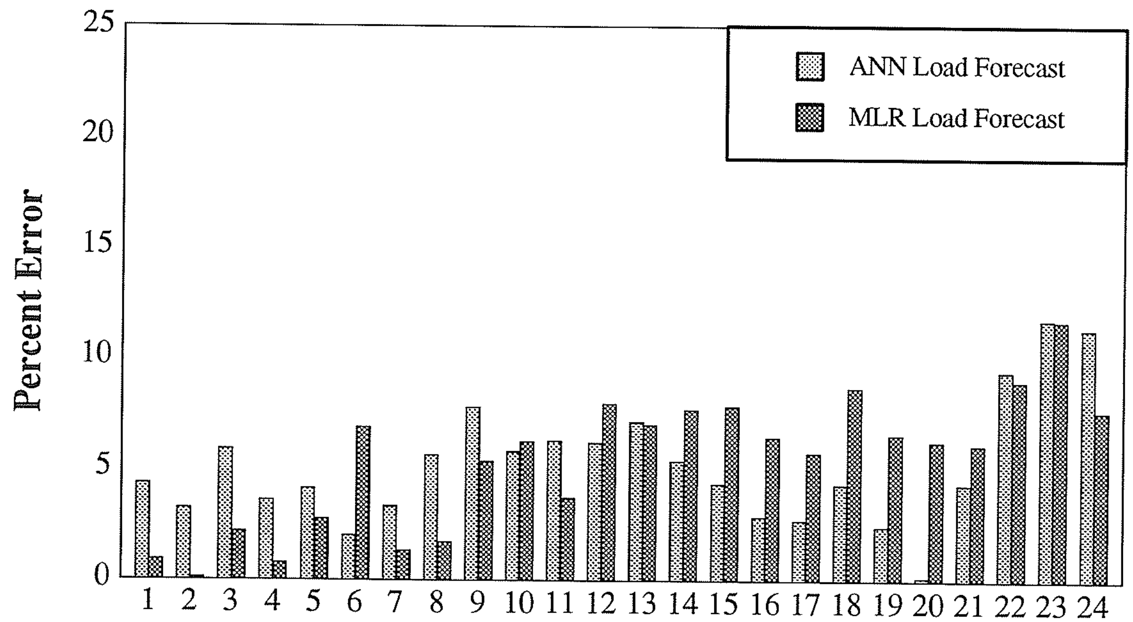


Forecast Error

FIGURE A12 : Forecast and Error for May 12, 1992

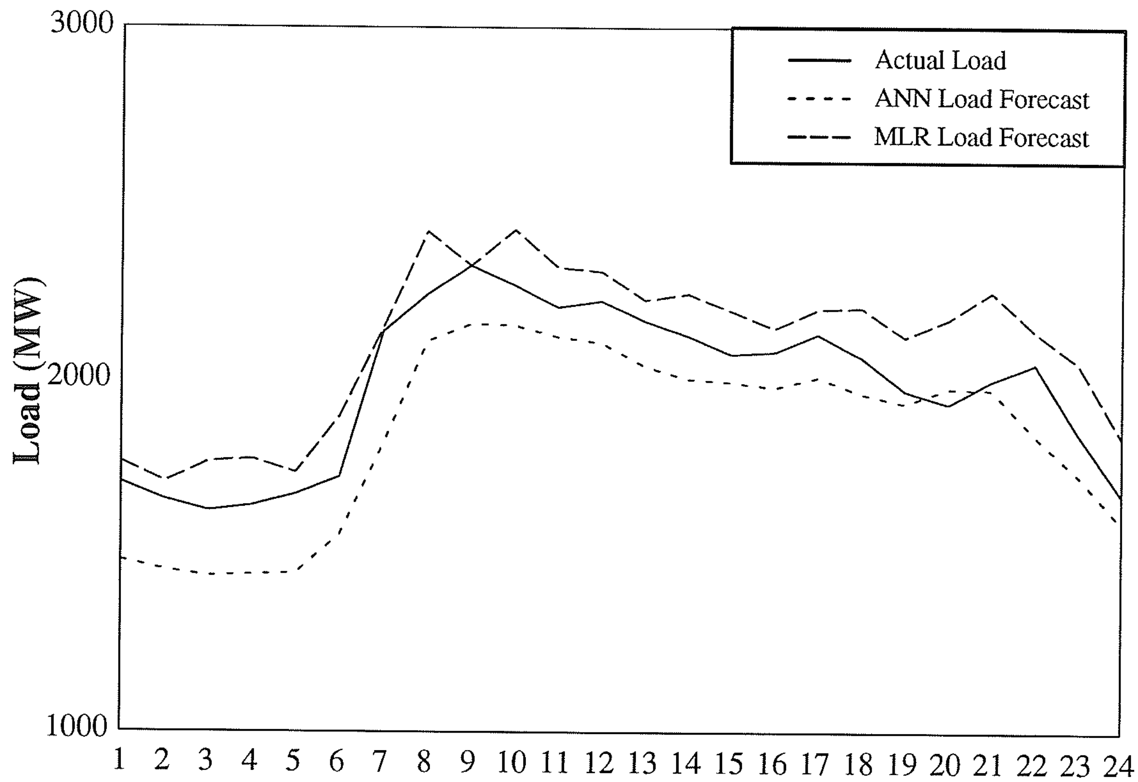


Daily Load Curve

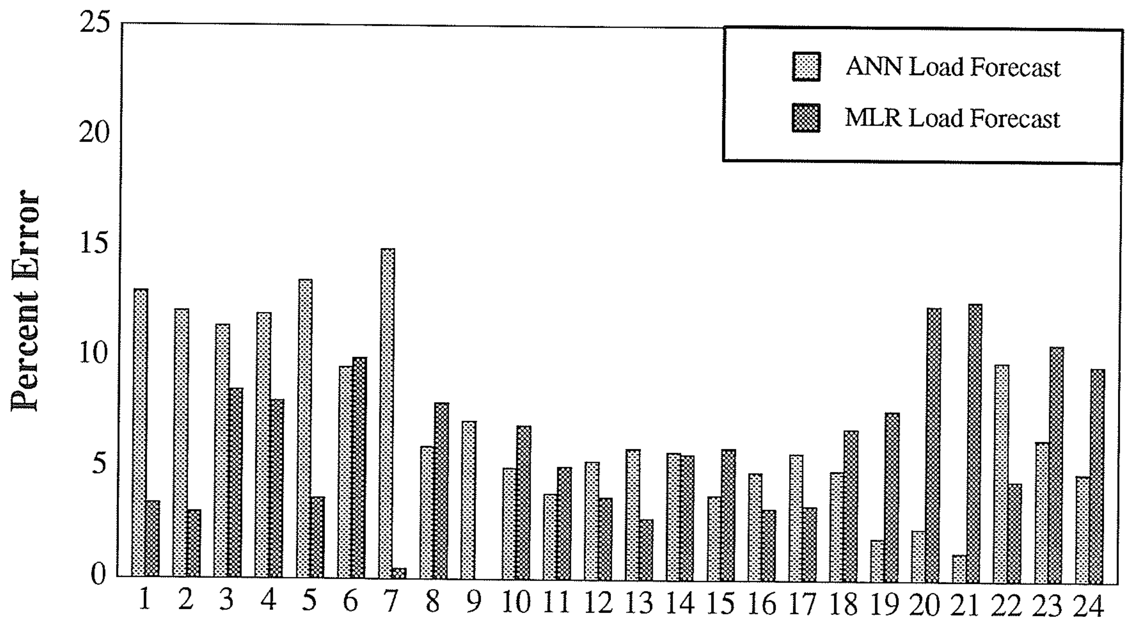


Forecast Error

FIGURE A13 : Forecast and Error for May 13, 1992

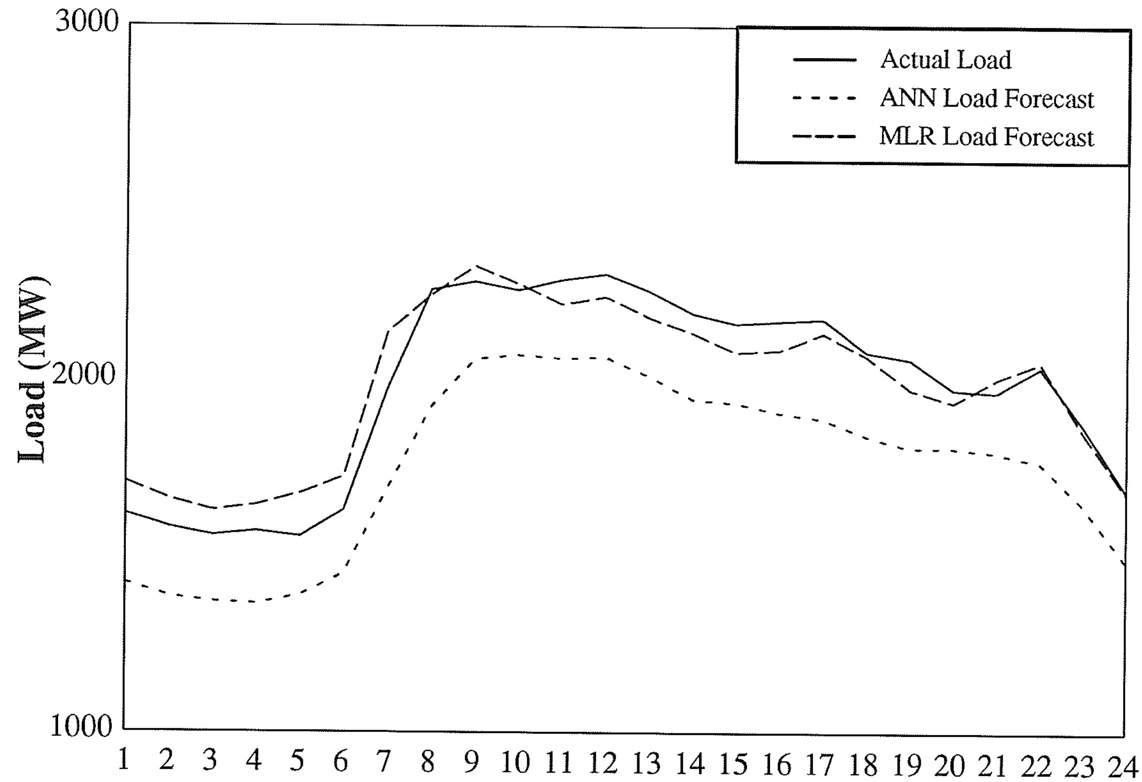


Daily Load Curve

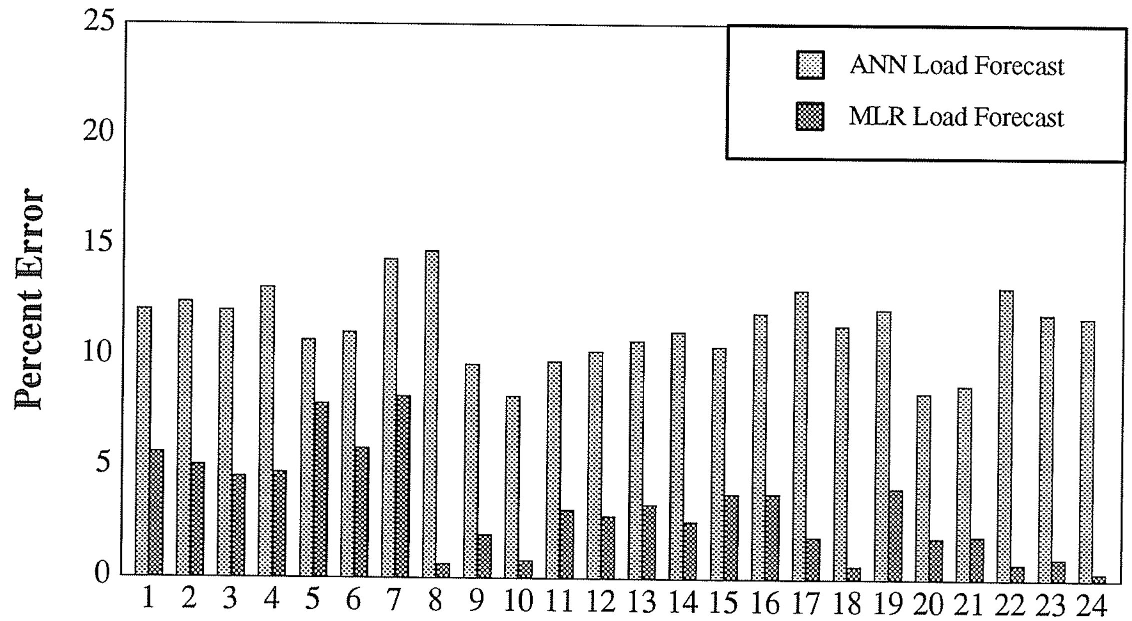


Forecast Error

FIGURE A14 : Forecast and Error for May 14, 1992

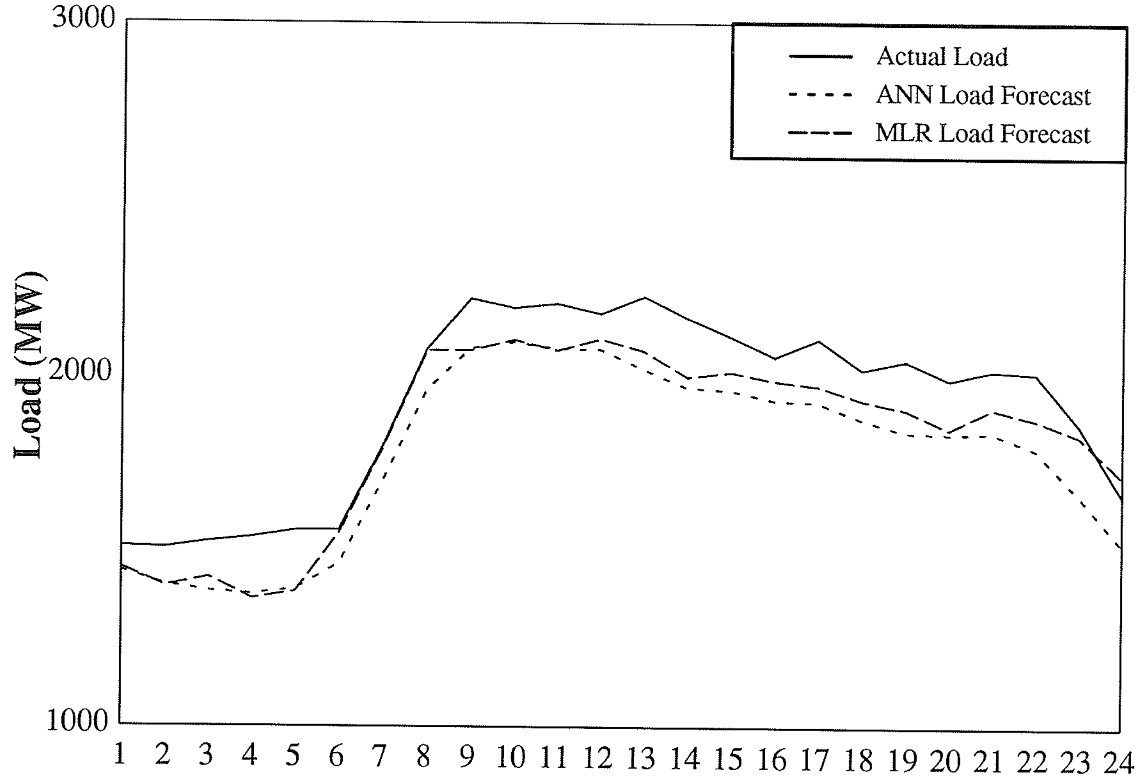


Daily Load Curve

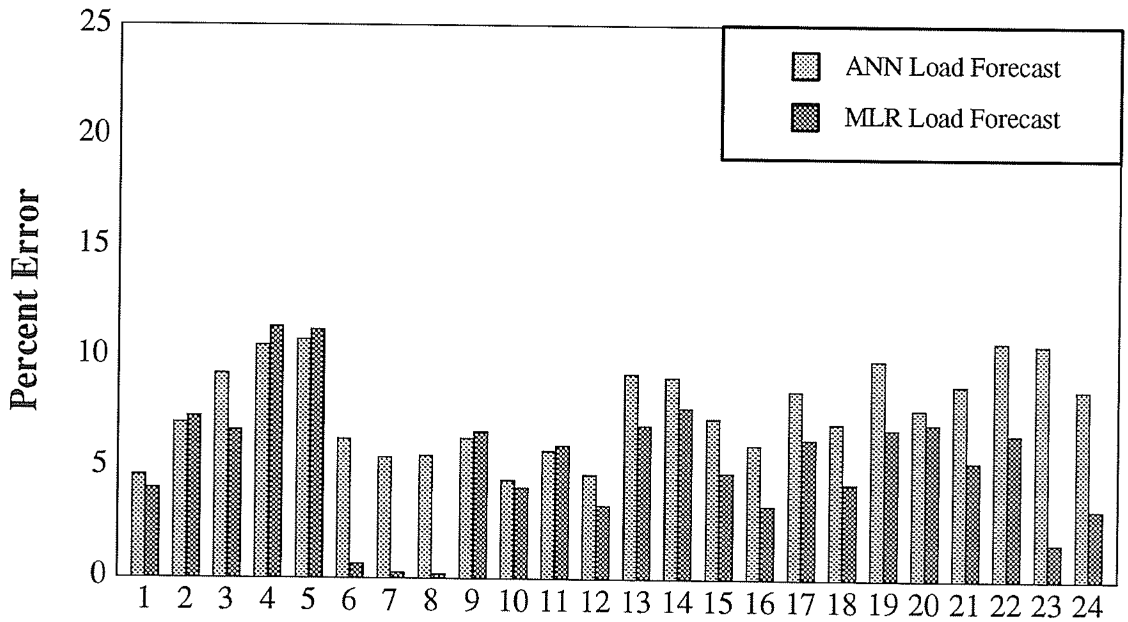


Forecast Error

FIGURE A15 : Forecast and Error for May 15, 1992

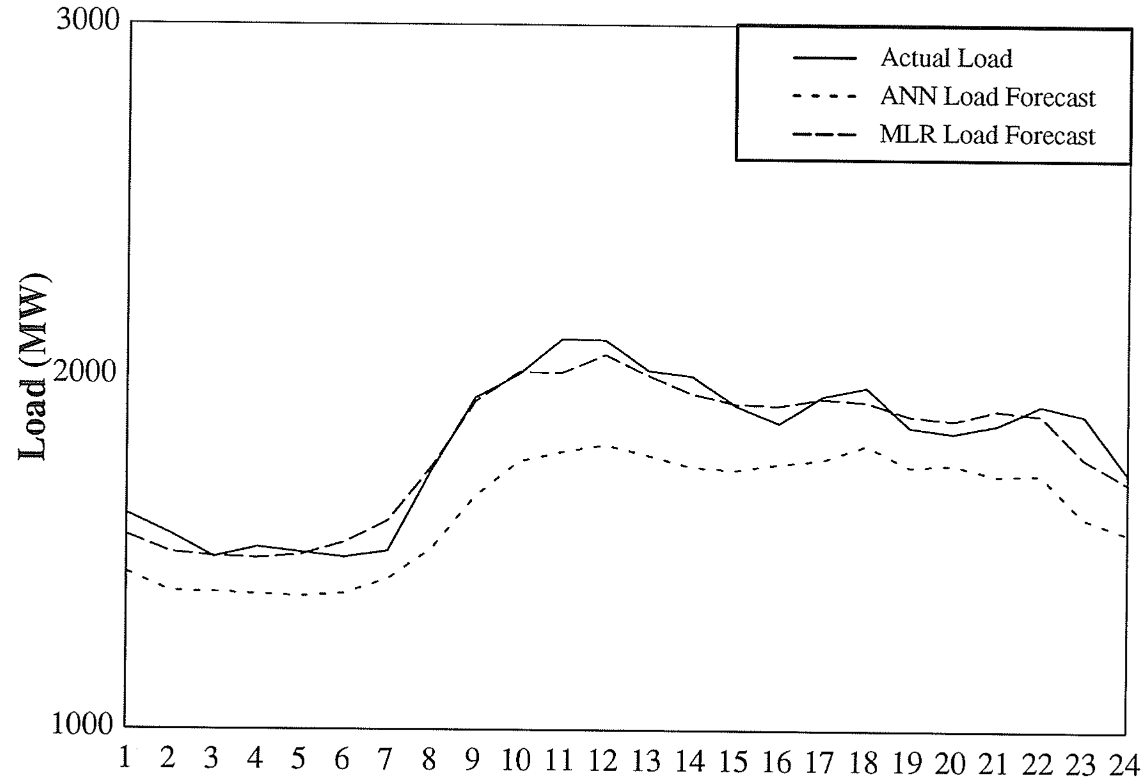


Daily Load Curve

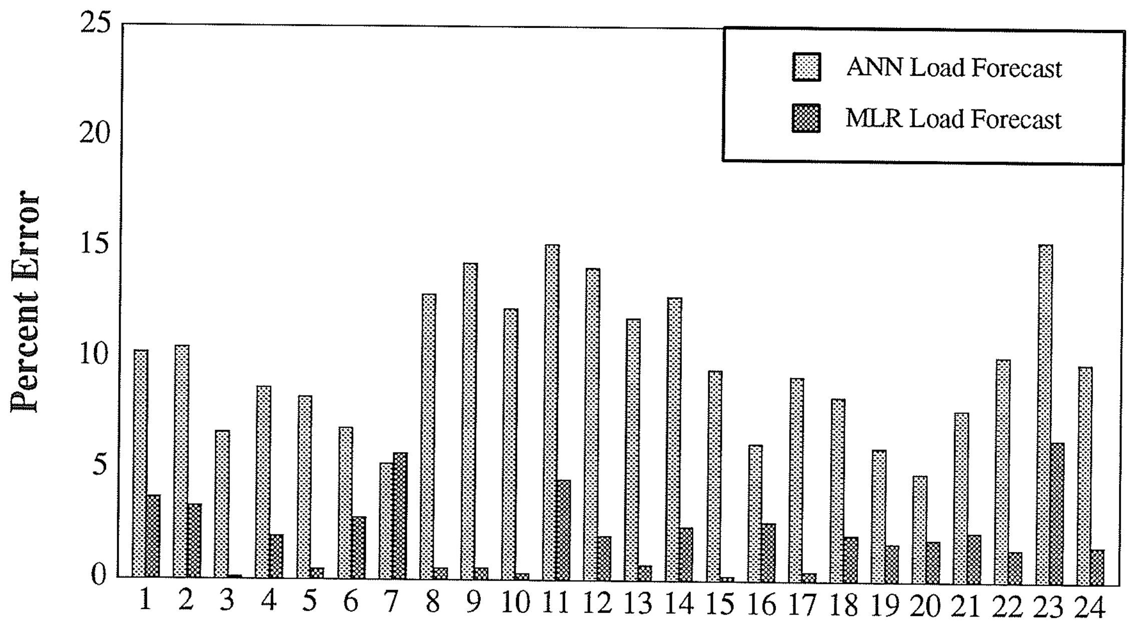


Forecast Error

FIGURE A16 : Forecast and Error for May 16, 1992



Daily Load Curve



Forecast Error

FIGURE A17 : Forecast and Error for May 17, 1992

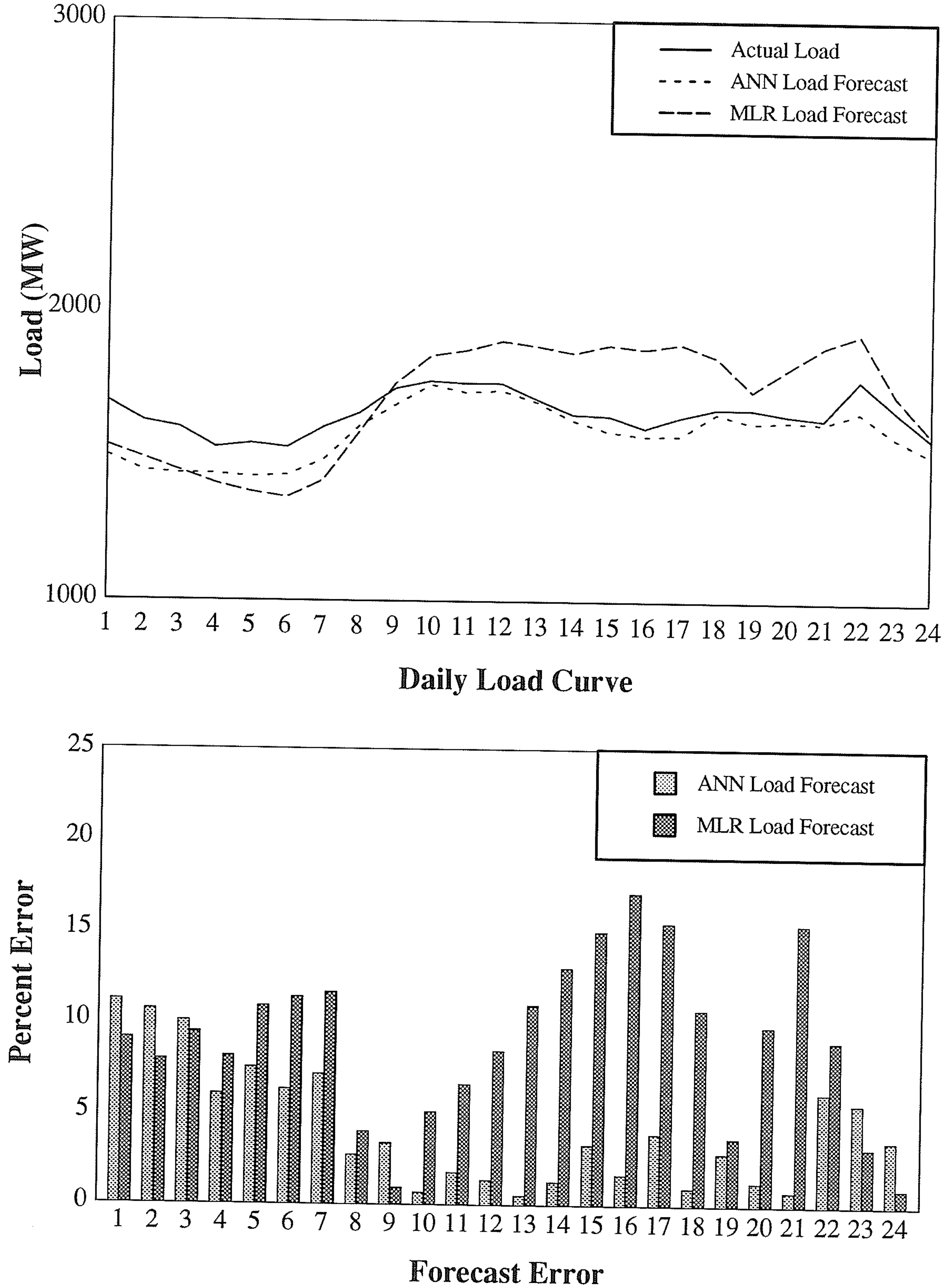
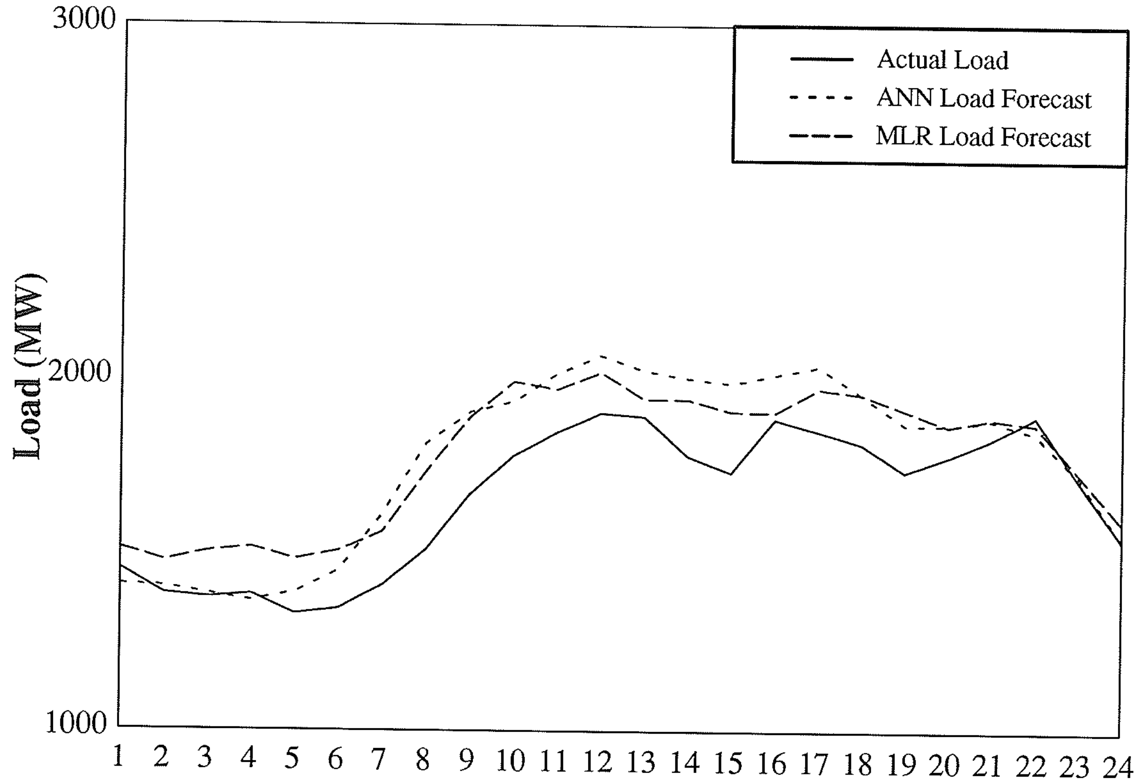
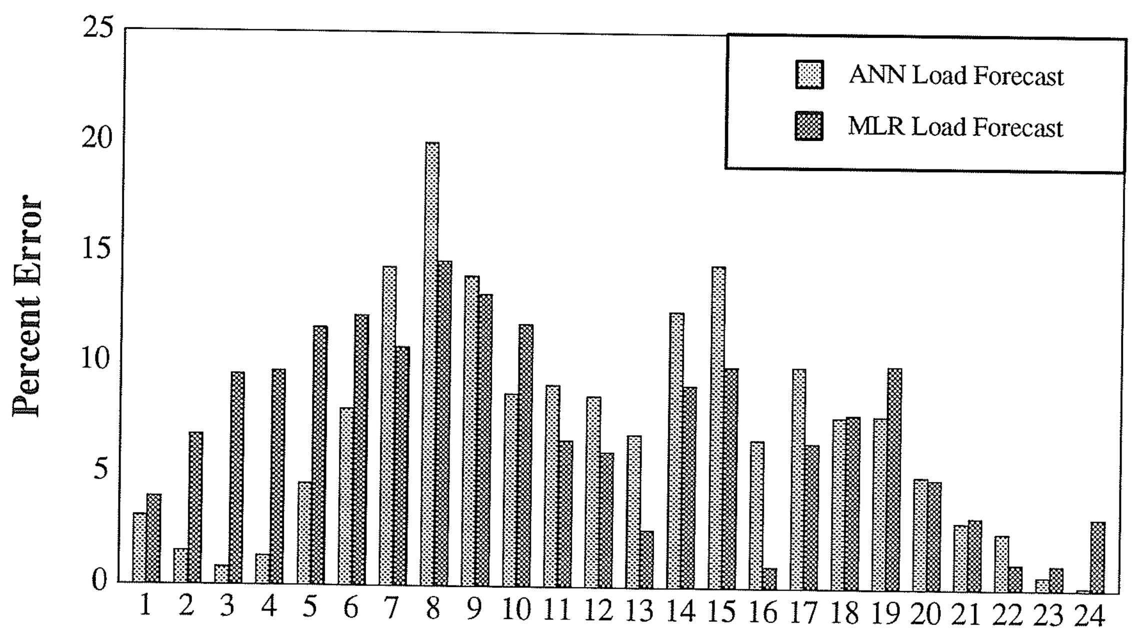


FIGURE A18 : Forecast and Error for May 18, 1992



Daily Load Curve



Forecast Error

FIGURE A19 : Forecast and Error for May 19, 1992

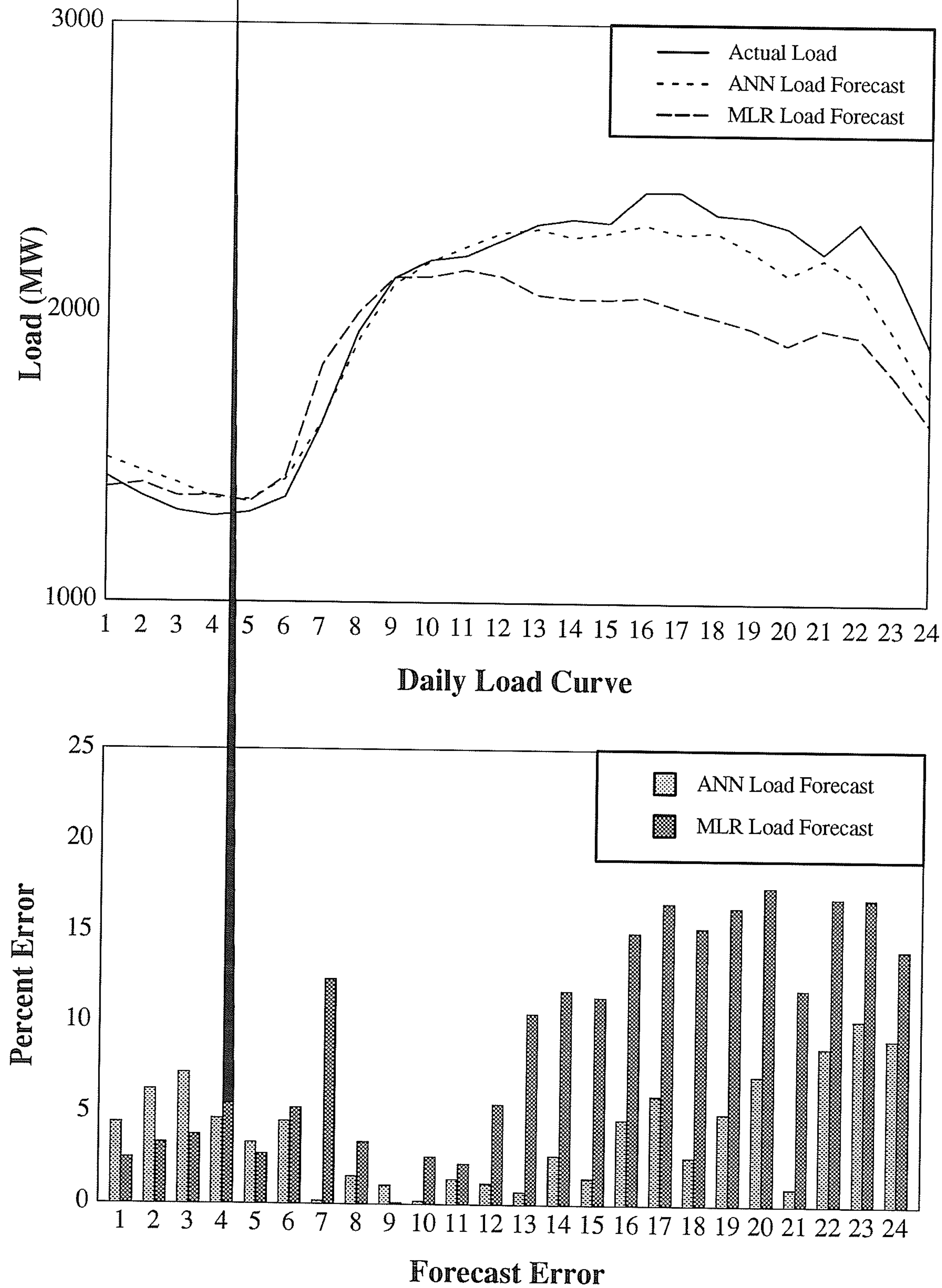
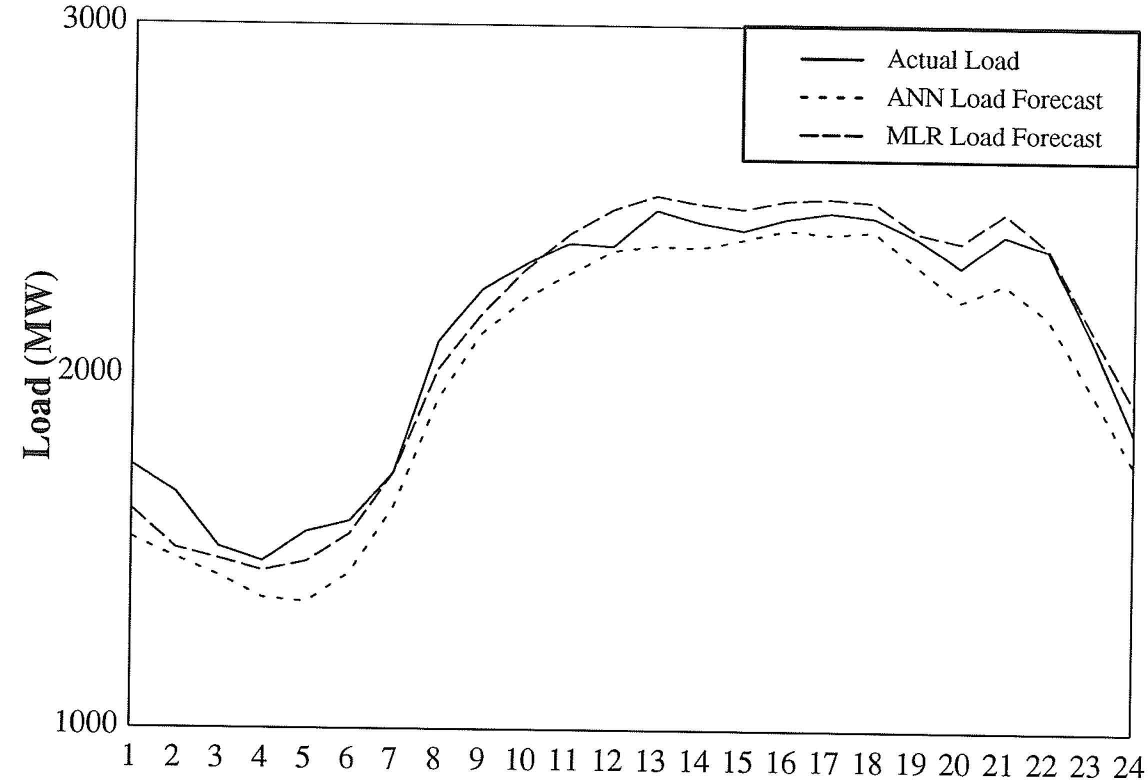
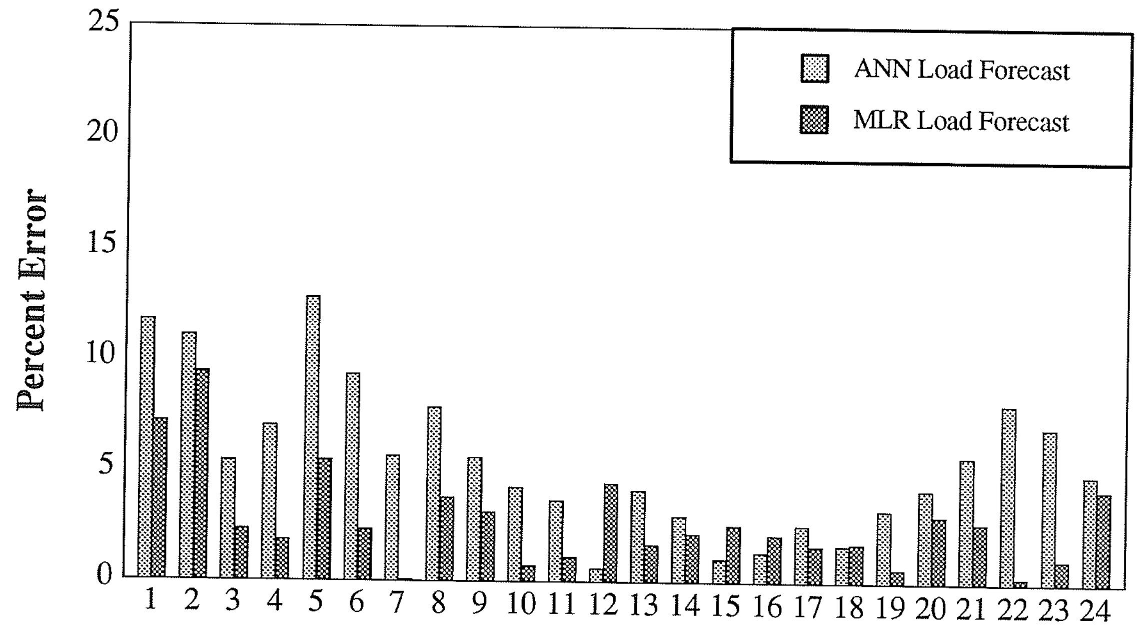


FIGURE A20 : Forecast and Error for May 20, 1992



Daily Load Curve



Forecast Error

FIGURE A21 : Forecast and Error for May 21, 1992

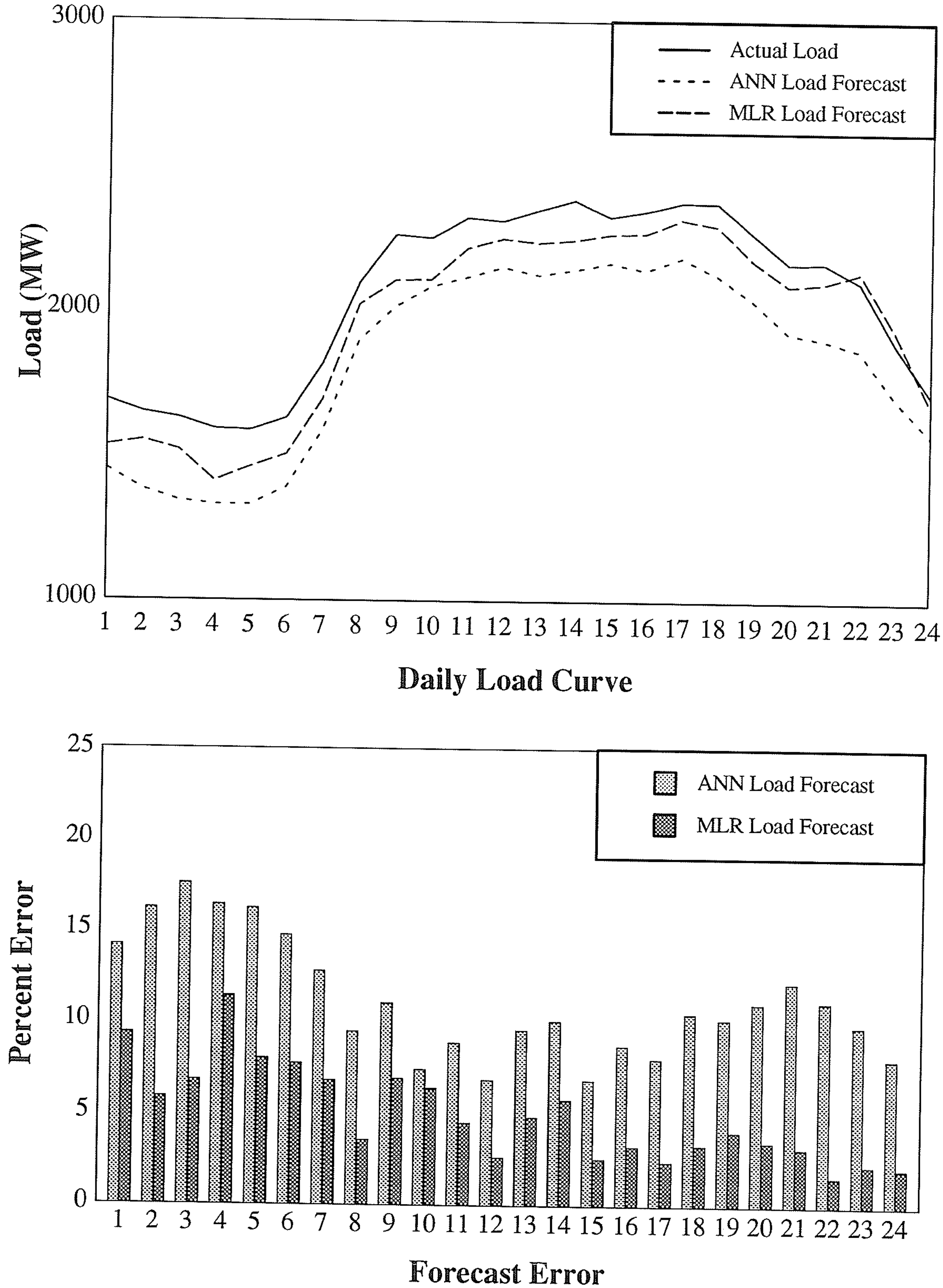
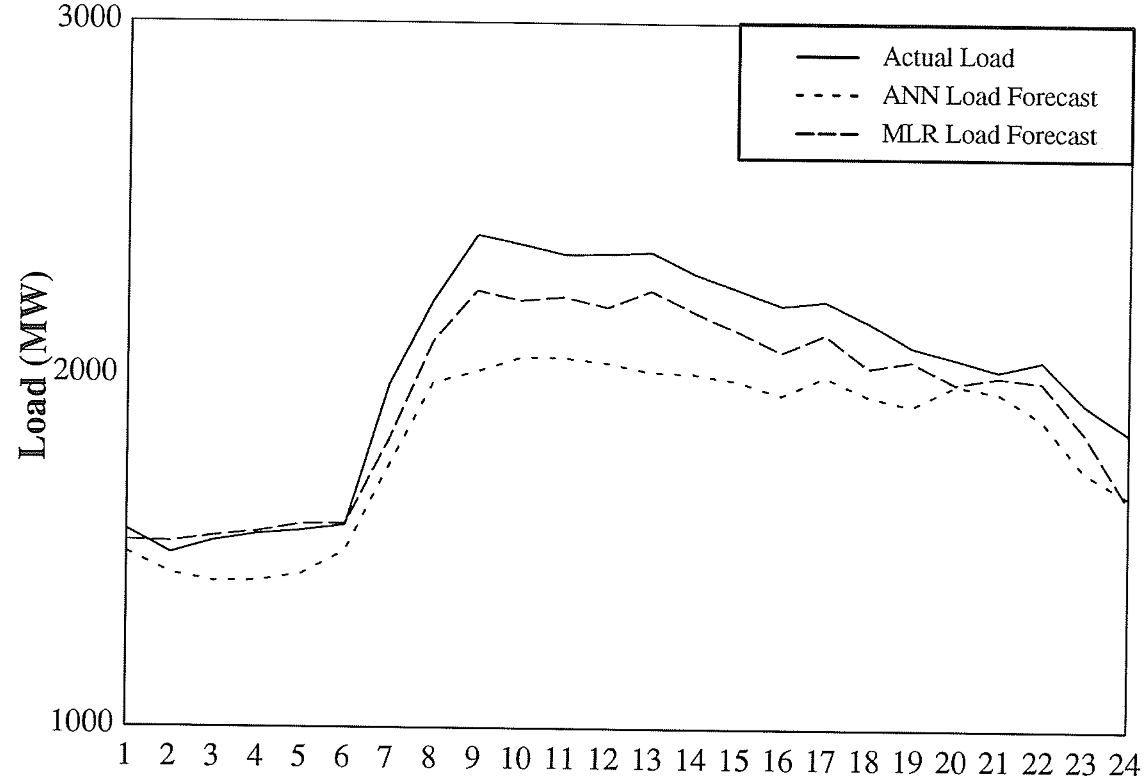
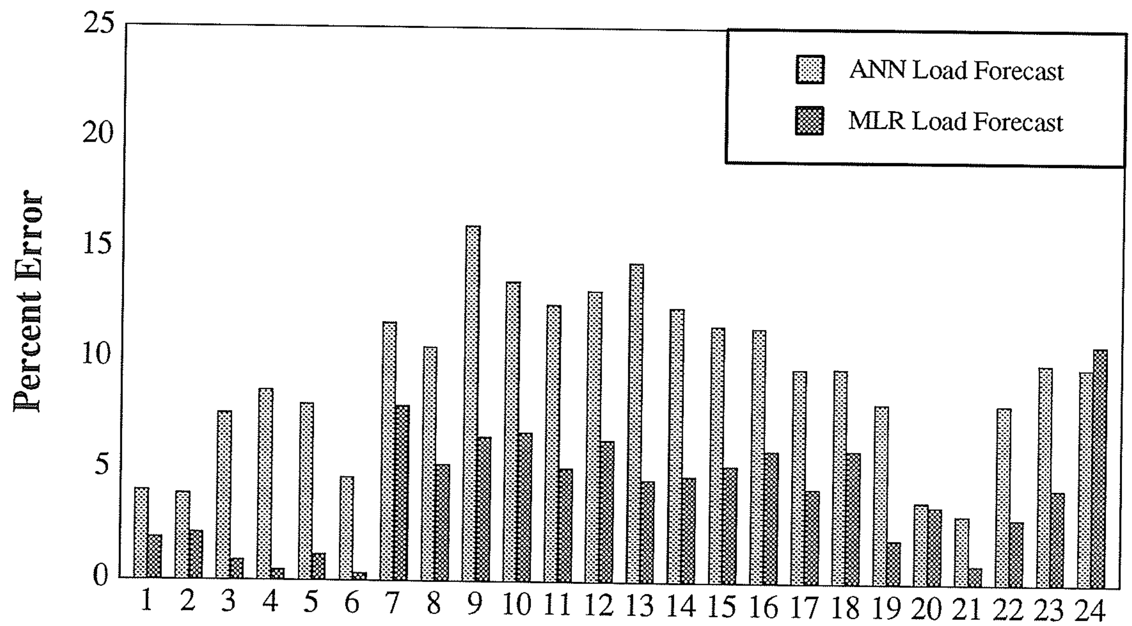


FIGURE A22 : Forecast and Error for May 22, 1992

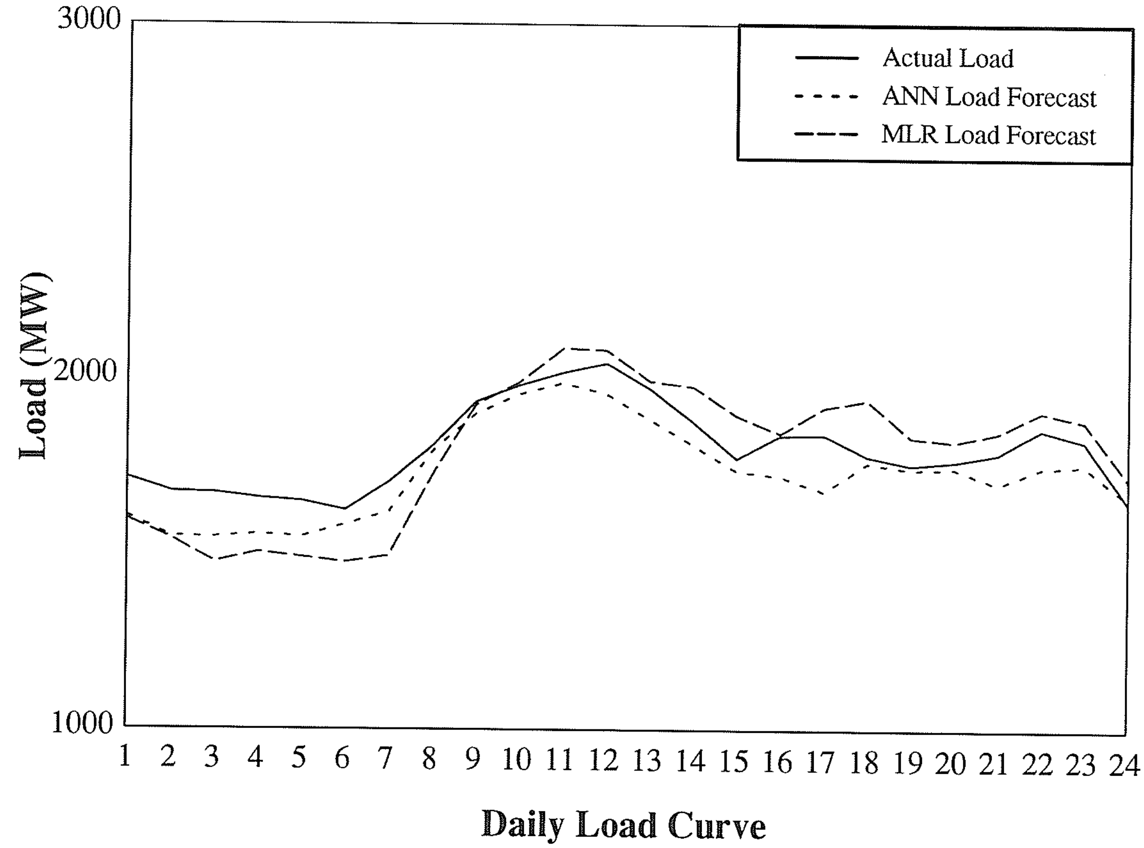


Daily Load Curve



Forecast Error

FIGURE A23 : Forecast and Error for May 23, 1992



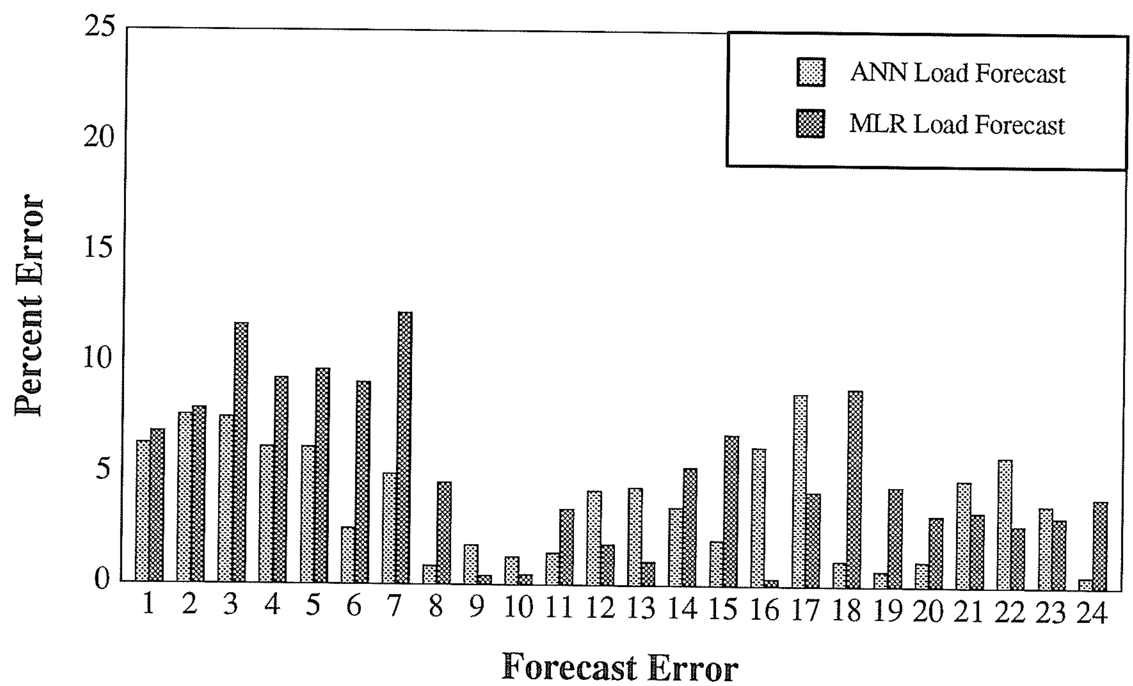
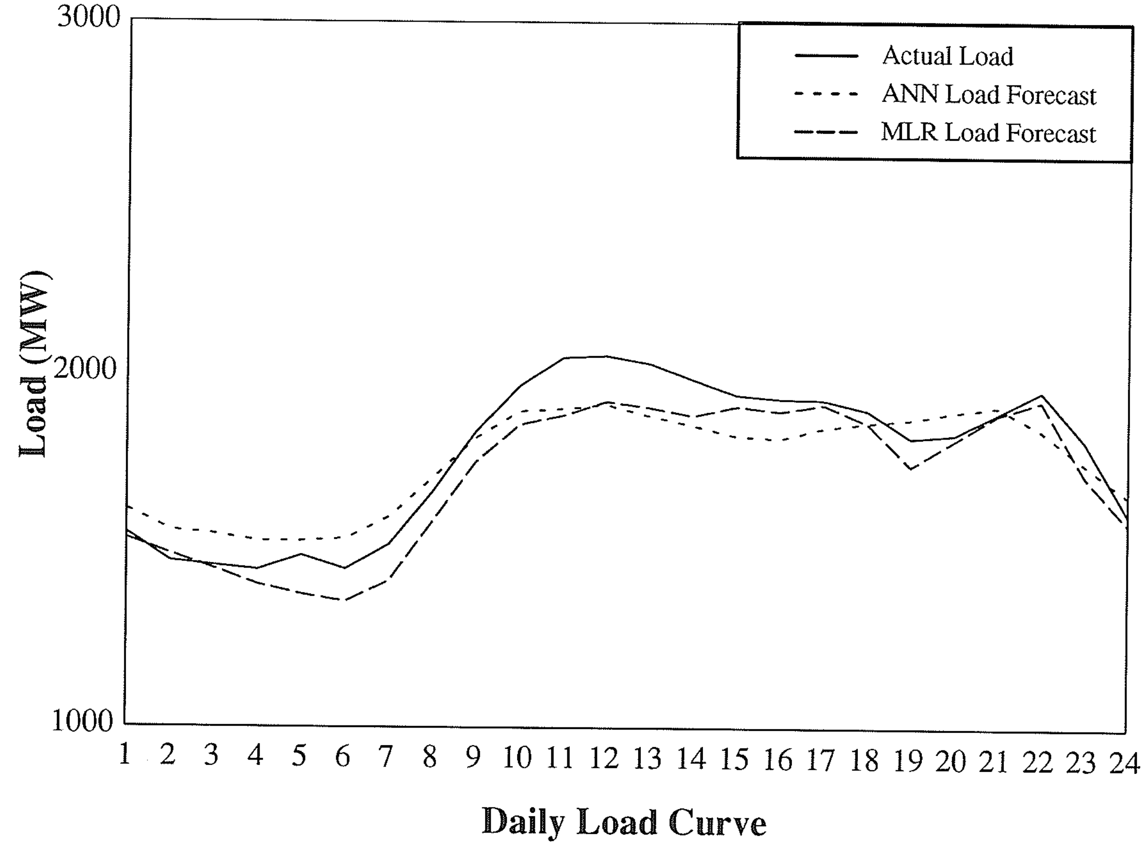


FIGURE A24 : Forecast and Error for May 24, 1992



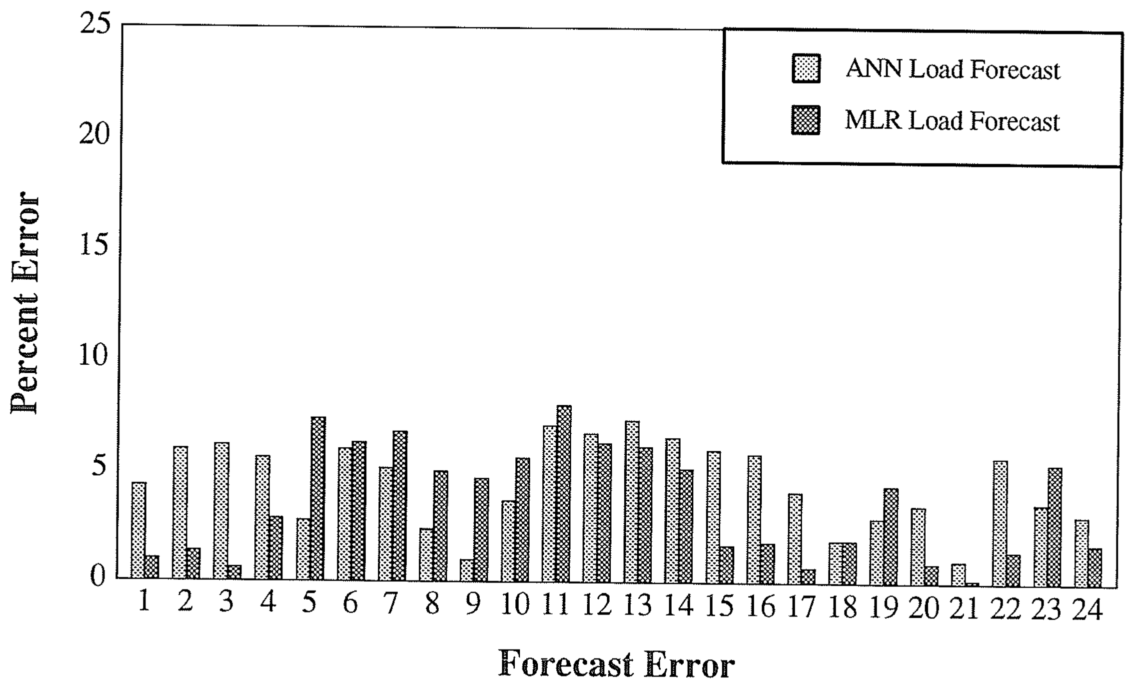
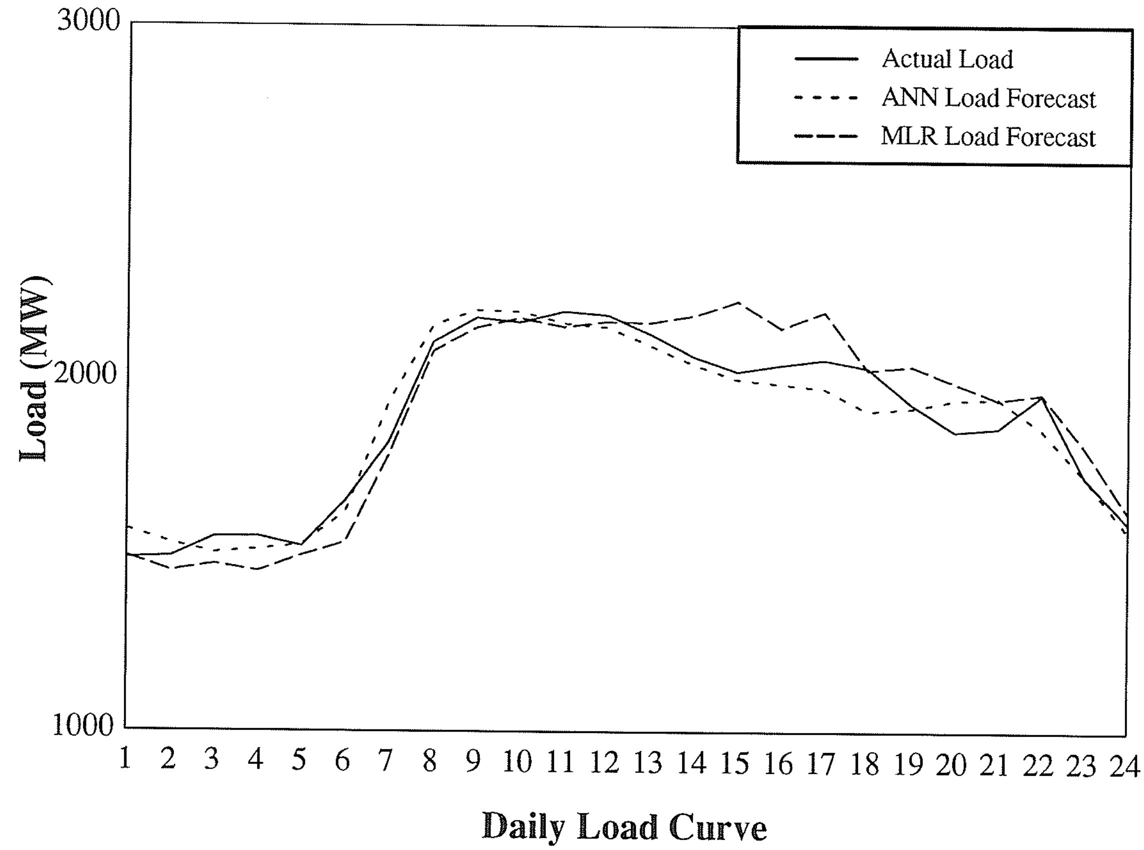


FIGURE A25 : Forecast and Error for May 25, 1992



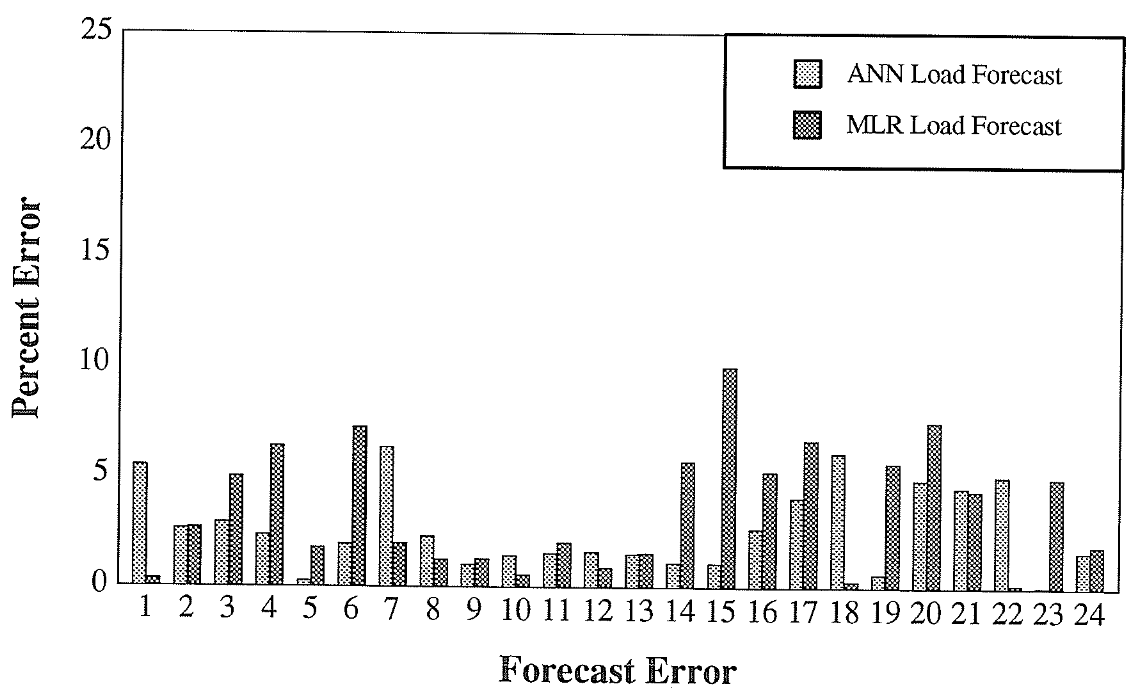
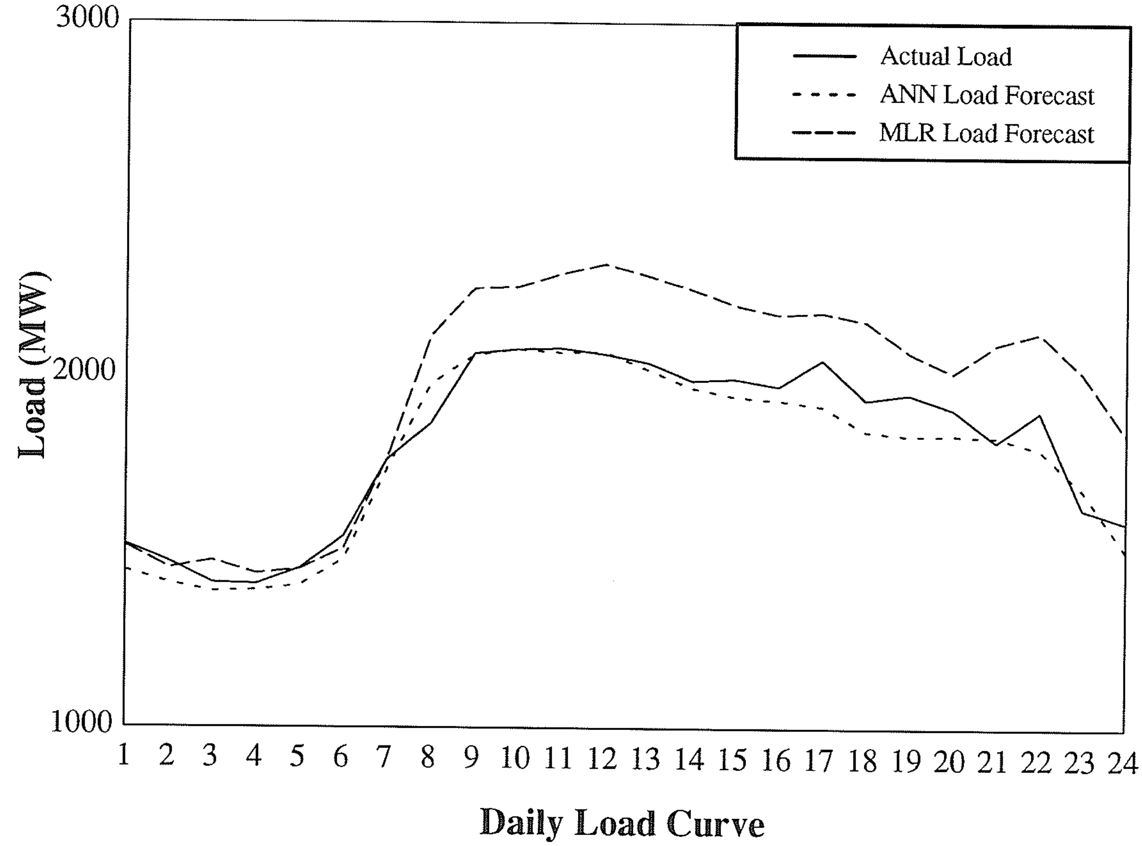


FIGURE A26 : Forecast and Error for May 26, 1992



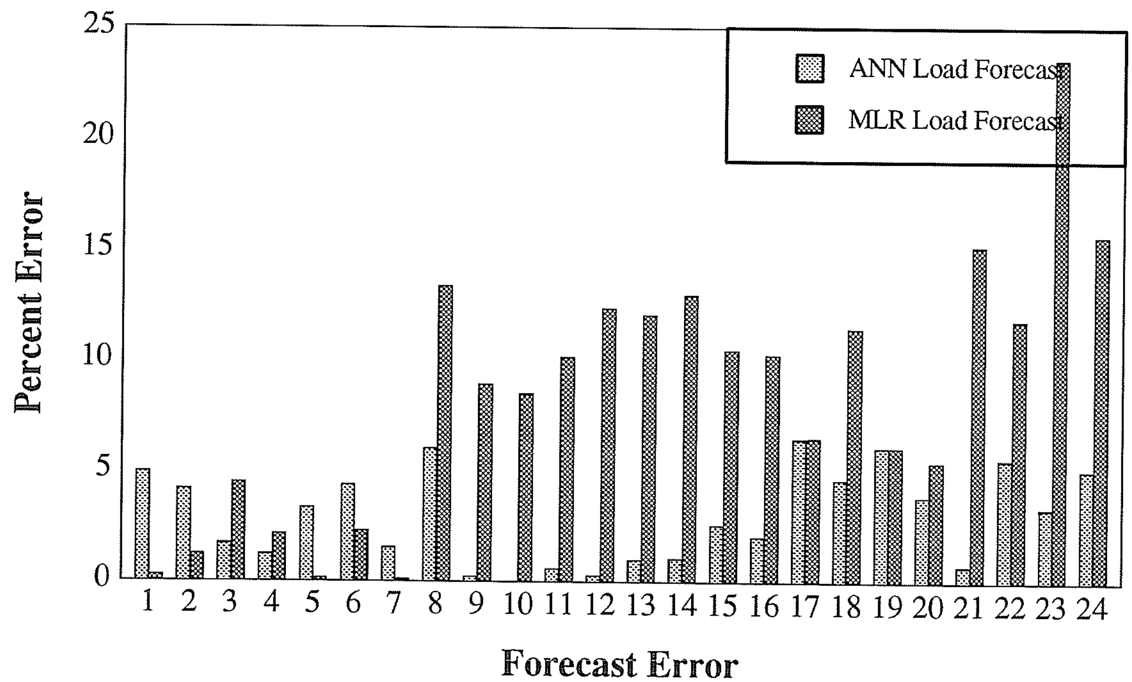
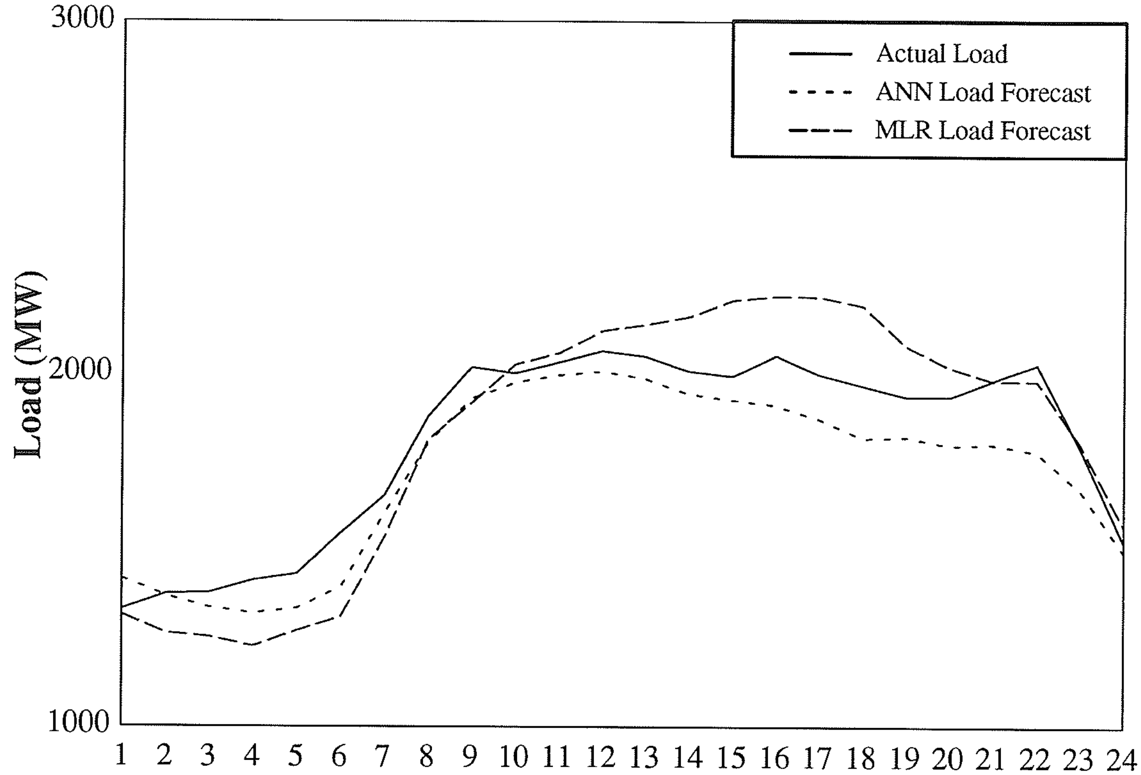
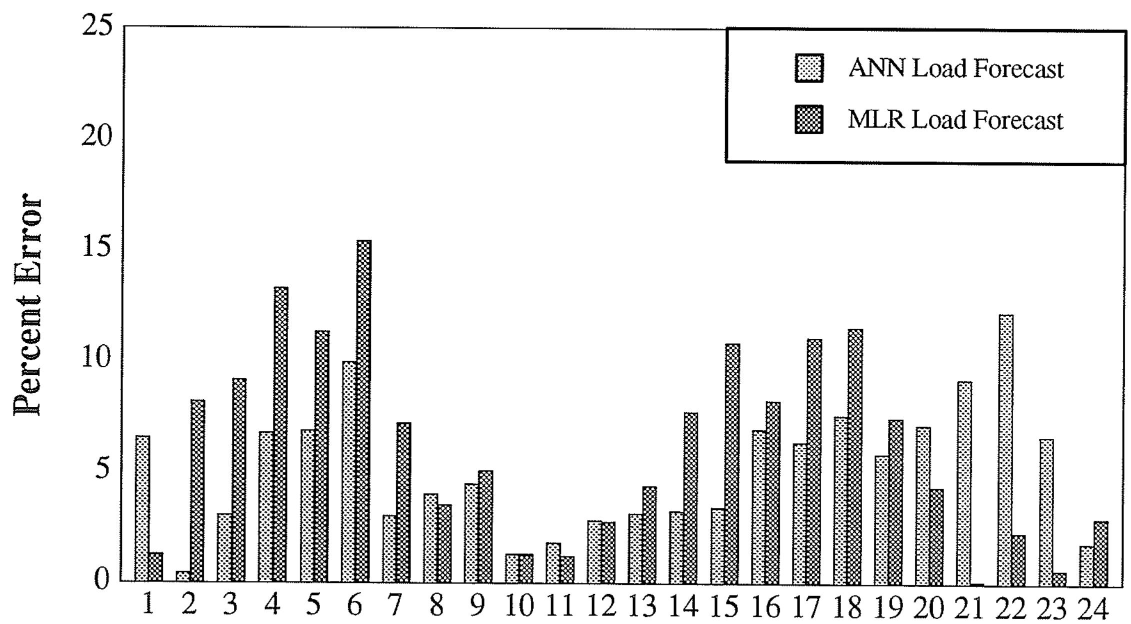


FIGURE A27 : Forecast and Error for May 27, 1992

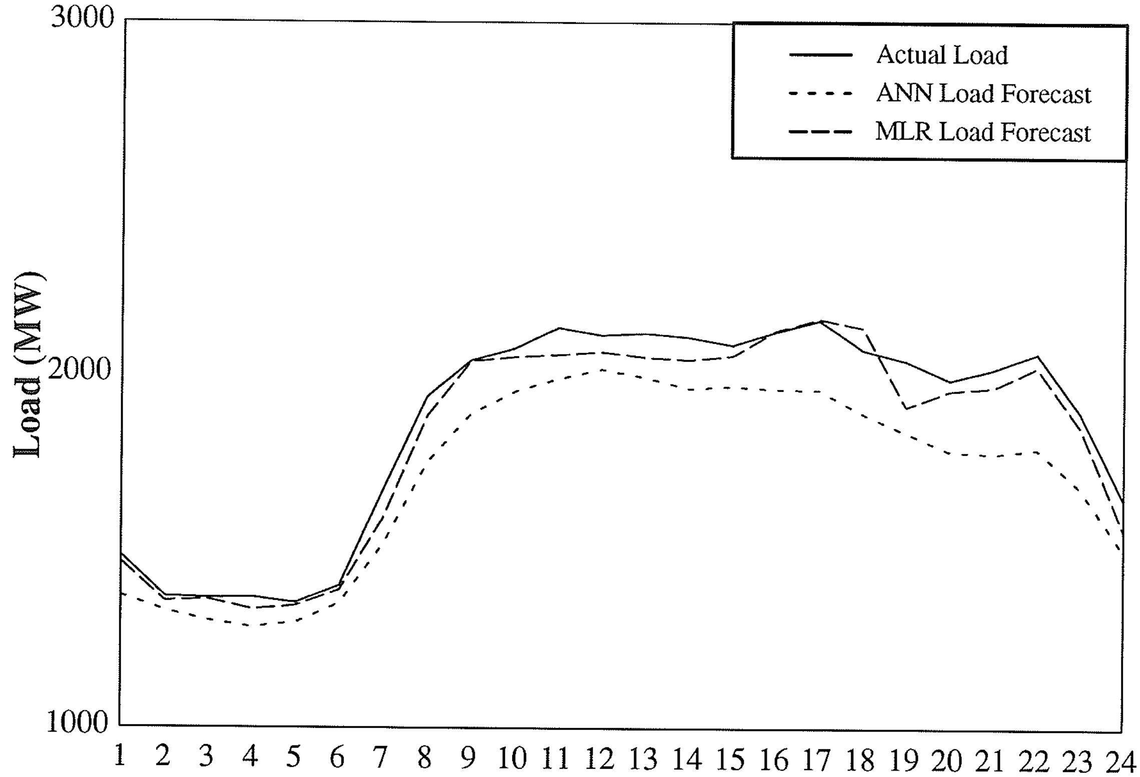


Daily Load Curve

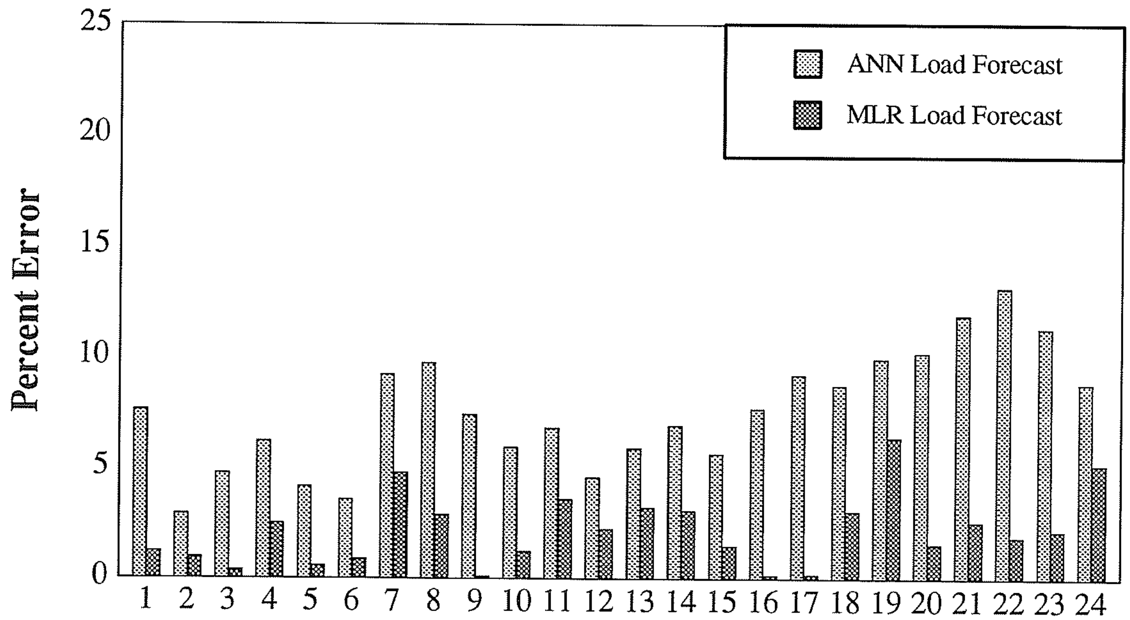


Forecast Error

FIGURE A28 : Forecast and Error for May 28, 1992

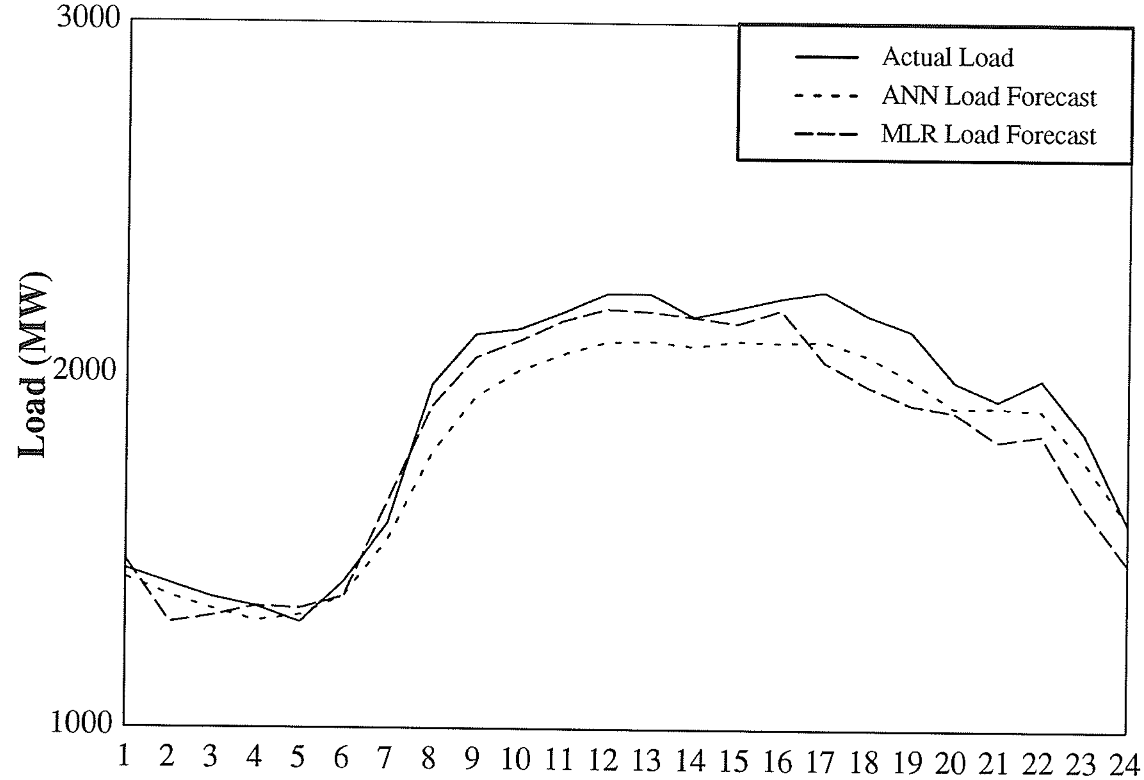


Daily Load Curve

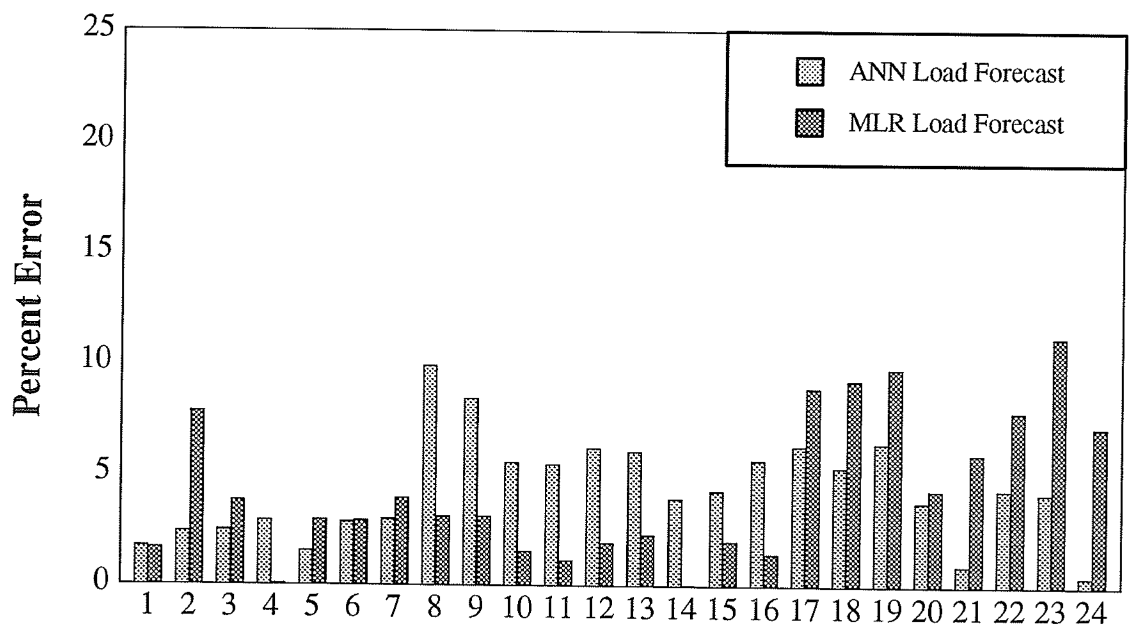


Forecast Error

FIGURE A29 : Forecast and Error for May 29, 1992

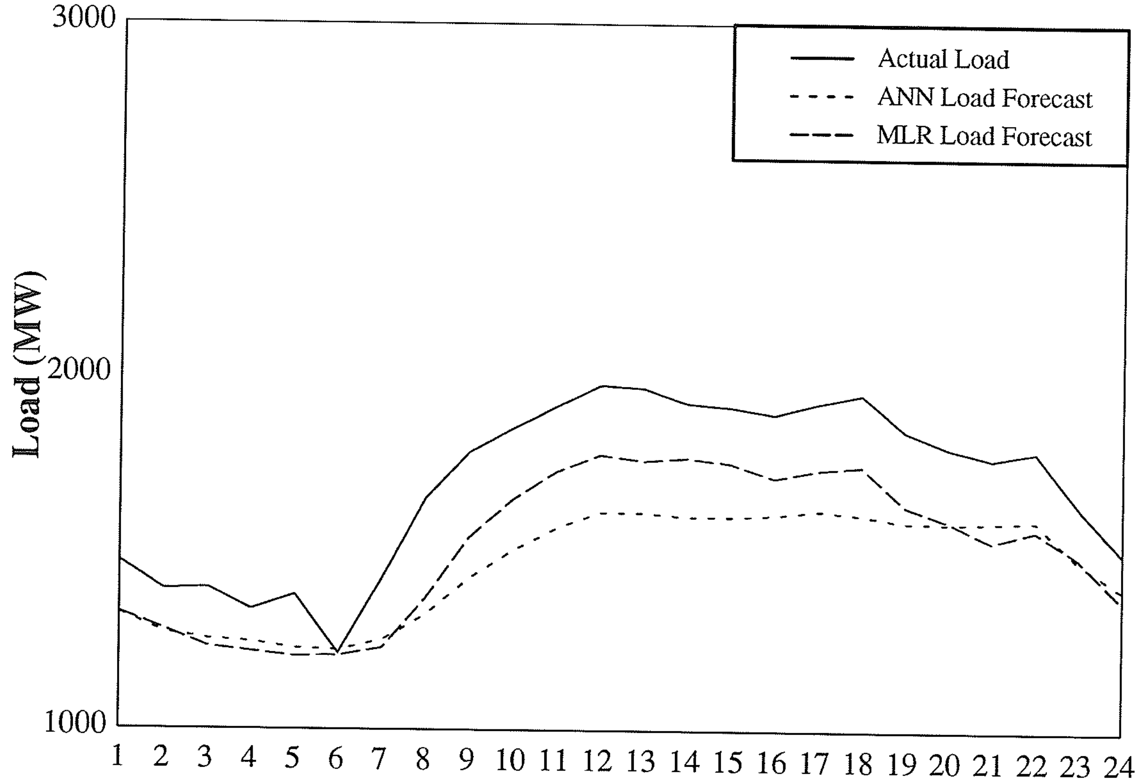


Daily Load Curve

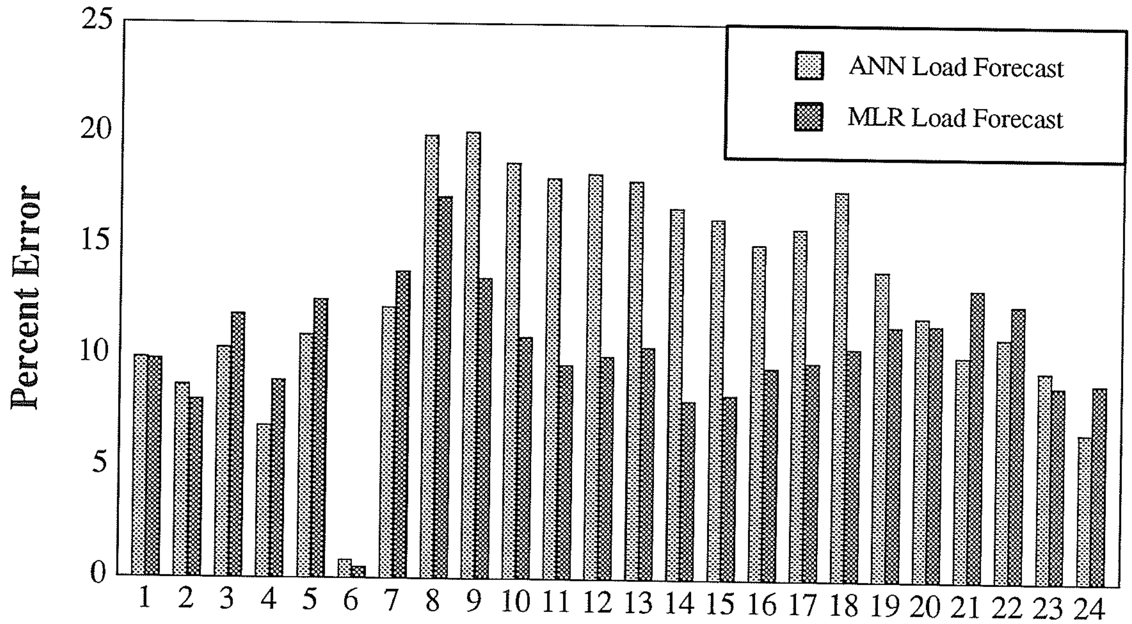


Forecast Error

FIGURE A30 : Forecast and Error for May 30, 1992

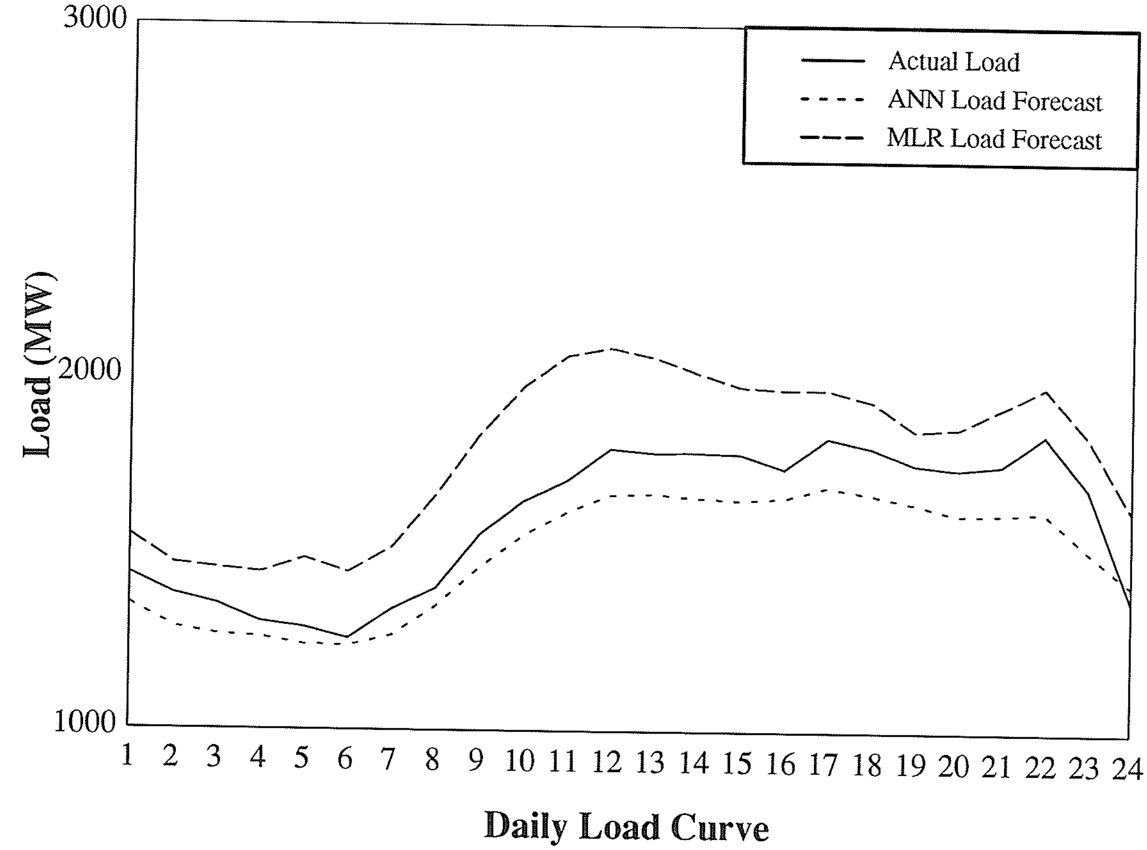


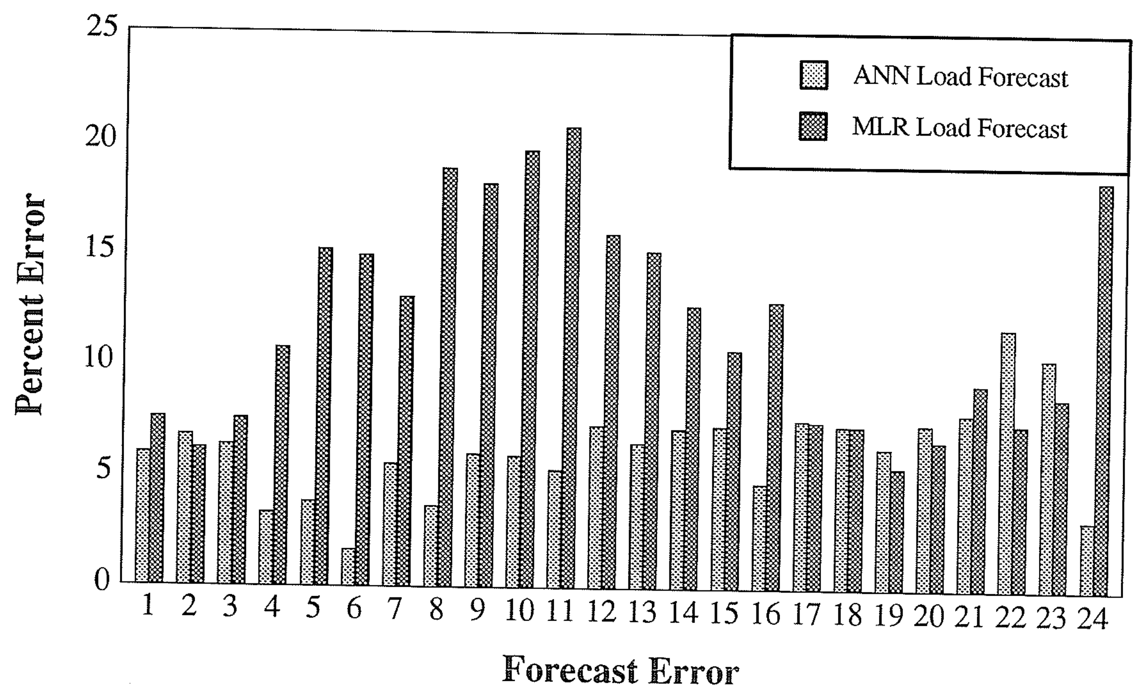
Daily Load Curve



Forecast Error

FIGURE A31 : Forecast and Error for May 31, 1992





APPENDIX B :

THE BACKPROPAGATION ALGORITHM

USING

CONJUGATE GRADIENT MINIMIZATION

The BP algorithm using Conjugate Gradient minimization is :

1. $t = 0$, $count = 0$, Randomly initialize weight values (of dimension N), set line minimum θ, set maximum iterations $\mathbf{t_{max}}$,

2. For each training pattern in the training set :

a. Apply randomly chosen input pattern to network inputs and propagate through to network outputs,

b. Calculate error function based on output training pattern, and partial derivatives of the error function for the output layer,

c. Propagate error back through each hidden layer, calculating remaining partial derivatives,

d. Sum error derivatives over entire training set,

3. Determine conjugate direction $\mathbf{h_t}$ for step t

$$\mathbf{h_t} = \begin{cases} -\nabla \mathbf{E}(\mathbf{w}(\mathbf{t})) & ,\mathbf{count} = 0 \\ \mathbf{g_t} + \gamma_{t-1}\mathbf{h_{t-1}} & ,0 < \mathbf{count} < \mathbf{N} \end{cases}$$

where for count > 0 : $\mathbf{g_t} = -\nabla \mathbf{E}(\mathbf{w_t})$, and

$$\gamma_{t-1} = \frac{(\mathbf{g_t} - \mathbf{g_{t-1}}) \cdot \mathbf{g_t}}{\mathbf{g_{t-1}} \cdot \mathbf{g_{t-1}}}$$

4. Calculate line minimum along conjugate direction, λ_t, if less then line minimum θ stop,

$$\lambda_t = \frac{\mathbf{g_t^T} \cdot \mathbf{h_t}}{\mathbf{h_t^T} \cdot \mathbf{A_t} \cdot \mathbf{h_t}} \qquad ,\mathbf{t} \geq 0$$

5. Update weights :

$$\mathbf{w}(\mathbf{t}+1) = \mathbf{w}(\mathbf{t}) + \lambda_t \mathbf{h_t}$$

6. Increment t and count, if $\mathbf{t} > \mathbf{t_{max}}$ then stop, else if $\mathbf{count} = \mathbf{N}$, the dimension of the weight vector, then restart count = 0, goto step 2.

APPENDIX C :

UNSUPERVISED LEARNING

FOR

SHORT TERM LOAD FORECASTING

Self organizing techniques applied to short term load forecasting have been investigated in the mid 1970's by T.S. Dillon [25]. The resurgence of neural networks in the 1980's, in particular the development of the backpropagation algorithm, led to an interest in ANN's applied to STLF using supervised techniques [8–18]. However the research of unsupervised learning techniques for STLF has remained for the most part unexplored.

Unsupervised learning differs from supervised learning in that there is no target output vectors in the training set. Each training pattern consists solely of an input vector.

This means that there is no method for evaluating an error function based on the output produced by these networks. Learning of synaptic weights is therefore unsupervised, meaning that upon presentation of an input vector, the network determines these weight updates dynamically – such that closely related input vectors will activate neurons which are near to each other. This is called clustering, and in general is performed on an arrangement of neurons in one two or three dimensions so as to be evaluated in real space.

The input vector consists of three components of information, as opposed to two for supervised learning. The first two are the same as for supervised learning, weather and date/time data, while the third component is the load (since there is no particular output for unsupervised learning).

These unsupervised algorithms require much more computational overhead then the supervised learning algorithms. This computational overhead grows rapidly with the size of the network. Therefore to keep the training time manageable it was decided to investigate networks with serial load presentation, and overall small input vector dimensions (<10).

When training the network, the weight vectors are updated to become similar to the training vectors. In this way the weight vectors become a representative set of vectors for the training set. Then when some unknown input vector, with one or more elements of the vector missing, is presented to the network, the neuron whose weight vector is most similar to the input vector becomes the winner. The winner's weight vector becomes the model for the input vector, and any missing elements can be supplied with the corresponding elements from the weight vector. Once trained, load forecasting will be performed by showing the network input vectors with the load element missing. The load forecast is then supplied by the corresponding element of the weight vector chosen as most similar to the input vector.

Two algorithms for unsupervised learning were tried, Kohonen Self Organizing Maps [26] (KSOM) and Fuzzy Kohonen Clustering Networks [27] (FKCN). The KSOM algorithm was used initially, but the FKCN algorithm was found to have advantages over KSOM, (not label dependent and terminates naturally), and therefore replaced it in the research [27].

In general these networks did not perform as well for load forecasting as those using supervised learning. For May 1992, the month for which Manitoba Hydro chose for comparison, unsupervised learning using the FKCN algorithm with 51 cluster centers (neurons) had an overall average error of 8.3% or 152 MW.

While for forecasting itself, unsupervised learning did not perform as well as desired, it still proved a useful tool for STLF. The KSOM weight maps yielded information which could be translated into rules for an expert or fuzzy system, while also reenforcing the results obtained that indicate which input types should be used and which have no real correlation to the load. In particular they indicated that there was no coherent mapping for sky cover, and while the wind speed had some relevance, the mapping of windchill and wind together indicated that again wind speed was best presented as a component of windchill. With the FKCN algorithm, overall forecasting improved, and it was found that by tailoring the training set to summer[20] and winter sets that many of the problems with averaging could be eliminated (that is the averaging of high winter peaks, with low spring / fall peaks).

20. Summer training includes Spring and Fall.

Furthermore unsupervised techniques yielded information, not available to supervised techniques, on the actual relationships between weather and load. That is, forecasting of load is just one area that a trained network can be used for. Just as easily could one or more other vector element(s), such as temperature or wind chill, could be left out. The result is to find weather conditions that would result in certain types of loads, for instance system peak loads. This would give the system operator knowledge about specific weather conditions to look out for, and to aid in the task of scheduling (maintenance, load sales/purchases, etc.), which is the basic purpose of STLF.