Neural network application in power systems load forecasting

Ali Yazdian Varjani

University of Wollongong
NOTE
This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG
COPYRIGHT WARNING
You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
In the name of God, the merciful and compassionate
Neural Network Application in Power Systems Load Forecasting

A thesis submitted in fulfilment of the requirements for the award of degree

Master of Engineering (Honours)

From

Department of Electrical and Computer Engineering

UNIVERSITY OF WOLLONGONG

By:

Ali Yazdian Varjani

B.Sc. Elec. Eng. Sharif University of Technology

1994
Declaration

This is to certify that the work presented in this thesis was carried out by the author in the Department of Electrical and Computer Engineering, the University of Wollongong, and has not been submitted to any other university or institute.

Ali Yazdian Varjani
Acknowledgments

I wish to thank my supervisor Dr. Parviz Doulai for his supervision and invaluable guidance throughout this research.

Financial support provided by the Ministry of Culture and Higher Education of the Islamic Republic of Iran is thankfully acknowledged. Supplying the valuable statistical data by the Pacific Power is also thankfully acknowledged.

My special thanks to my wife and my daughter who have supported and put up with me throughout the course patiently.

I would also like to thank, Mr. Tavasoli, Mr. Jalilian and Mr. Shahri for their valuable technical discussions throughout my research. Special Thanks to Mrs. B. Evans for her assistance in thesis proof reading and to all the members of the Department of Electrical and Computer Engineering for their combined efforts.
This thesis is dedicated to my mother, my brother and my wife, without whose encouragement this thesis would not have been completed.
Abstract

This thesis aimed to study all available short-term load forecasting methods in an attempt to suggest a solution (algorithm/structure) which gives the most appropriate forecast output for a typical input data set containing historical load data with or without weather variables input data. In this study, matters such as forecast accuracy, speed, development/implementation costs, and historical data validation were observed very closely.

The two most successful time series based (conventional) load forecasting methods namely, auto regressive moving average and general exponential smoothing have been thoroughly studied and implemented, using available power utility input data. The relevant forecasting results along with discussion highlighting the inherent problems associated with conventional methods are presented.

To fully examine the operation of a neural network for short-term load forecasting, a brief introduction to its theory and its implementation techniques is presented. Two major types of neural network structure (connection) for one-hour and 24-hour ahead load forecasting were selected. This is supplemented by two learning algorithms that are used for training and testing of the networks. To obtain the best possible forecast results, some structure modification was implemented along with the introduction of a modified learning algorithm. The sample forecast results for a standard and a proposed systems are shown.

The proposed 24-hour ahead neural network forecast was compared with the results obtained from the most accurate conventional method using the same input data for training of the neural network and for model identification using this conventional method.
# Table of Contents

Declaration ........................................................................................................... II

Acknowledgments ............................................................................................ III

Abstract ............................................................................................................. V

Table of Contents ............................................................................................ VI

List of Figures .................................................................................................. XI

List of Tables .................................................................................................. XIII

Chapter 1 Introduction ................................................................................... 1

1.1 Power Systems Load Forecasting ................................................................. 1

1.2 Load Forecasting Categories ....................................................................... 2

1.3 Short-term Load Forecasting ...................................................................... 2

1.4 Forecasting Approaches ............................................................................ 4

1.4.1 Quantitative methods ........................................................................... 5

1.4.2 Qualitative Methods ............................................................................. 8

1.4.3 Artificial Neural Network ...................................................................... 10

1.4.4 Hybrid methods .................................................................................. 14

1.5 Thesis Outline ......................................................................................... 15

Chapter 2 Load Components ........................................................................... 18

2.1 Introduction ............................................................................................... 18

2.2 Load Curve ............................................................................................... 19

2.3 Load Decomposition ................................................................................ 20

2.3.1 Weather Sensitive Component .............................................................. 26

2.4 Forecasting ............................................................................................... 27

2.5 Summary ................................................................................................. 27
Chapter 3 Conventional Methods for Load Forecasting

3.1 Introduction

3.2 Time Series

3.3 Auto-Correlation

3.4 Partial Auto-Correlation Function (PACF)

3.5 Stationary ARMA Processes
   3.5.1 Auto Regressive Processes
   3.5.2 Moving Average Process
   3.5.3 Auto Regressive Moving Average Process

3.6 Non-Stationary Series
   3.6.1 Differencing
   3.6.2 Integrated Processes

3.7 Seasonal Models
   3.7.1 Seasonal Differencing
   3.7.2 General ARIMA(p,d,q)(P,D,Q) Process

3.8 Identification of Model Parameters

3.9 Forecasting

3.10 Forecasting: General Exponential Smoothing (GES)
   3.10.1 General Model
   3.10.2 Forecast for load time \( t \)
   3.10.3 Estimating the Coefficients
   3.10.4 Simulation

3.11 Summary
Chapter 4 Artificial Neural Networks ................................................. 51

4.1 Knowledge - Based Expert System (KBES) ........................................ 52

4.1.1 Artificial Intelligence in Power Systems ......................................... 53

4.2 Artificial Neural Network Modeling .................................................. 55

4.2.1 Definition ......................................................................................... 55

4.2.2 Structure ......................................................................................... 56

4.2.2.1 Artificial Neuron ........................................................................ 56

4.2.2.2 Multi Layer Perceptron ............................................................ 57

4.2.2.3 Learning process ......................................................................... 58

4.3 Neural Network Capabilities and Limitations ........................................ 59

4.3.1 Capabilities ..................................................................................... 59

4.3.2 Limitations ...................................................................................... 60

4.4 Applications ....................................................................................... 61

4.5 Neural Network Based Load Forecasting .............................................. 62

4.5.1 Neural Network Model Design ....................................................... 63

4.5.2 Data Pre-processing ........................................................................ 64

4.5.2.1 Neural Network Architecture .................................................... 68

4.5.2.2 Training and Testing ................................................................. 70

4.6 Summary and Conclusion .................................................................... 71

Chapter 5 Forecasting Results and Comparison ....................................... 72

5.1 Data Collection .................................................................................. 72

5.2 Model Development ........................................................................... 73

5.2.1 Operating Mode I: One-Hour ahead Forecast .................................. 75

5.2.1.1 Partially Connected Neural Network ......................................... 76

5.2.1.2 Fully Connected Neural Network ............................................. 79

5.2.2 Operating Model II 24-hour ahead forecast .................................... 80
5.2.2.1 MLP with Back-propagation Learning ........................................... 82

5.2.2.2 MLP Network with Combined Back Propagation and Self Organisation Map learning algorithm ...................................................... 84

5.3 Assessment Methods ........................................................................... 85

5.4 Conventional Forecasting Methods ...................................................... 86

5.4.1 General Exponential Smoothing ...................................................... 86

5.4.2 Integrated Auto Regressive Moving Average .................................. 87

5.5 Forecast Results: Neural Network Methods ....................................... 87

5.5.1 Operating Mode I: One-Hour ahead Forecaster .............................. 88

5.5.1.1 Partially Connected ...................................................................... 88

5.5.1.2 Fully Connected Configuration ..................................................... 92

5.5.2 Operating Mode II: 24-hour ahead Forecast .................................... 94

5.5.2.1 MLP with Back-propagation Learning ........................................... 95

5.5.2.2 MLP Network with Combined Back Propagation and Self Organisation Map learning algorithm ...................................................... 96

5.6 Tabulated Presentation of Forecast Results ....................................... 98

5.7 Conventional Methods: Forecast Result and Comparison .................. 100

5.8 Sensitivity Analysis ............................................................................ 101

5.8.1 Bad Data ......................................................................................... 101

5.8.2 Network Configuration Design ....................................................... 105

5.9 Summary ............................................................................................ 105

Chapter 6 Conclusion ............................................................................. 106

6.1 Conclusion ......................................................................................... 106

6.1.1 Conventional Forecasting Methods ................................................ 107

6.1.2 Neural Network Based Forecasting Methods ................................ 108

6.2 Comparison ....................................................................................... 110

6.3 Further Study ...................................................................................... 111
Appendix A Back Propagation

A.1.1 Learning Procedure

A.1.1.1 Algorithm

A.1.2 Back Propagation Convergence

A.1.2.1 Learning Rate

A.1.2.2 Cumulative Delta Rule

A.1.2.3 Hidden Units

A.1.2.4 Recurrent Network
List of Figures

Figure 2.1: A typical average 24-hour load sequence .................................................. 19
Figure 2.2: An average 24-hour load sequence for weekend ........................................ 20
Figure 2.3: 16-day hourly load data ............................................................................. 21
Figure 2.4: Power Spectral Density of load data .............................................................. 22
Figure 2.5: Hourly load and Base load component (MW) .............................................. 24
Figure 2.6: Hourly load and Low frequency components of load .................................. 24
Figure 2.7: Hourly Load and High frequency component of load .................................. 25
Figure 2.8: Hourly Load and Residual component of Load .......................................... 25
Figure 3.1: Two-week hourly load data ........................................................................ 31
Figure 3.2: Auto Correlation Function .......................................................................... 32
Figure 3.3: Partial Auto Correlation Function for stationary signal ............................... 33
Figure 3.4 Partial Auto Correlation Function for non-stationary signal ......................... 33
Figure 3.5: Theoretical ACFs of an AR model with order 19 .......................................... 35
Figure 3.6: Theoretical ACFs of an AR model with order 19 .......................................... 36
Figure 3.7: Two-week hourly power system load data .................................................. 38
Figure 3.8: Two-week hourly load data after taking logs ............................................. 39
Figure 3.9: Two weeks differenced hourly load data ..................................................... 40
Figure 3.10: The actual and ARIMA-based forecast load .............................................. 45
Figure 3.11: The actual and General Exponential Smoothing based forecasting load .... 50
Figure 4.1: General procedure for building of an expert system ................................... 53
Figure 4.2: Simulated neuron ....................................................................................... 57
Figure 4.3: Common activation functions ..................................................................... 57
Figure 4.4: Multi Layer Perceptron Network ................................................................. 58
Figure 4.5: Neural network configuration for short-term load forecasting .................... 70
List of Tables

Table 4.1: List of inputs data for proposed neural network ............................................. 68
Table 5.1: The input data for one-hour load forecasting Mode I ....................................... 77
Table 5.2: The input data structure for 24-hour ahead load forecasting .......................... 81
Table 5.3: The ARIMA coefficients ............................................................................... 87
Table 5.4: Error and Standard deviation for the forecast result obtained from one-hour ahead operating mode ............................................................................. 98
Table 5.5: Summarised forecast information for 24-hour ahead load forecasting networks .................................................................................................................... 99
Table 5.6: The output changes as a result of a 5% change in the input data for one-hour ahead forecasting networks .............................................................................. 102
Table 5.7: The output changes as 5% changes in input data for 24-hour ahead forecasting networks (Model A) ......................................................................................... 103
Table 5.8: The output changes as a result of a 5% changes in the input data for 24-hour ahead forecasting networks (Mode II-BP-SOM) ................................. 104
Chapter 1

Introduction

1.1 Power Systems Load Forecasting

A secure and highly reliable source of electricity is an essential part of our modern society. Providing a reliable supply of electricity at the lowest possible price would require (among other things) sufficient generation to exactly meet customers fluctuating demand and system losses. One way that facilitates achieving this exact match between demand fluctuation (say, on an hourly basis) and energy generation is to estimate the demand in advance. In principle, this can be done using known demand patterns and weather forecasts. For long-term demand estimation, other factors such as economic growth and load composition, have to be taken into account. In a broad sense, this practice is commonly referred to as "load forecasting".

Load forecasting is an area of great economic value to the electric power utility. It is also a problem to be tackled on a daily basis. Reliable forecasting tools would enable power utilities to plan for peak demands and to achieve more economical unit allocation, scheduling and pre-dispatching.

The lead time in load forecasting ranges from a few minutes ahead for security assessments, to over forty years for long economic operations and planning. This could constitute a very short and a very long-term frame for power systems load forecasting [9].
1.2 Load Forecasting Categories

Broadly speaking, the very first question in a forecasting task is "How far in advance is the forecast needed?" There is a natural lead time for any process. Based on this factor, load forecasting is commonly classified into three categories; long-term, mid-range and short-term. Long-term load forecasting, with the range of more than twenty years, is used for planning purpose. That is to determine the type and the size of generating plants that minimise both fixed and variable costs. Mid-range load forecasting covering between one to three years, is used to assess revenue impact imposed by changes in electricity tariffs. It is also used in decision making about purchase of fuel and electricity import/export to and from other power utilities. Finally, Short-Term Load Forecasting (STLF), hour-by-hour prediction, is made for the day's power demand in the range of one to 24 hours ahead. It is used by electric utilities to control power plants; this minimises the generation cost while maintaining a secure supply [9], [16]. The lead time for STLF varies from a few minutes, for security assessment, to one week for unit commitment. In most cases, the required demand should be forecast for the next 24 hours. In this thesis, only short-term load forecasting up to 24 hours ahead is studied. Different implementation methods are investigated in detail.

1.3 Short-term Load Forecasting

Since the mid-1960's numerous researchers have studied the problem of short-term load forecasting and provided moderately accurate and efficient load forecasting methods to tackle this problem. The main motivation for this considerable amount of research on this topic centred on the cost saving that could be realised with the improved accuracy of forecasting. Different model
types based on different behavioural assumptions about load shape and considerations about the nature of forecasting, have been used. These considerations, related to the nature of models, can be summarised as follows [47]:

1. Lead time,
2. Time to prepare forecast,
3. Pattern of data,
4. Data requirements,
5. Ease of understanding,
6. Cost,
7. Accuracy, and
8. Real Time Processing.

Each of these factors has a specific role in the selection of a desired method. Some features, such as real time processing (on-line processing) and the development costs are more important compared to other factors. The selection of an appropriate method, based on the above factors, is an optimisation problem because the weighting of each factor in the selection of a method varies with the method used. However, the important factors that influence the process of model selection are accuracy, data requirement and lead time.

For short-term load forecasting historical load data and other relevant data, such as weather and the type of day, are used. In terms of lead time, short-term load forecasting is usually done for 24 hours ahead when the weather data and other parameters for the following day become available.
1.4 Forecasting Approaches

There are several methods available for short-term load forecasting. These methods can be broadly classified into following groups [7]:

1. qualitative methods,
2. quantitative methods, and
3. hybrid methods.

The qualitative method uses the knowledge of experts to anticipate the future demand. This class of forecasting also accommodates all Artificial Intelligence (AI) based methods. The Expert System (ES) and Fuzzy Expert System (FES) are the two most common examples among of AI based methods but only ES will be discussed in this thesis.

In quantitative methods the historical load data is analysed and the underlying function is identified. Then, this function is used for extrapolation or forecasting of future values. The quantitative methods are generally divided into two following groups:

1. time series, and
2. causal models.

The Artificial Neural Network is considered as a quantitative method. However, because of its nature of intelligence, it is usually compared to other methods in the AI category. Two or more forecasting methods may also be combined resulting in a "Hybrid forecasting approach". A hybrid method also produces a single forecast output for a given input data.

The literature survey on STLF presented here focuses on published contributions since 1988. This is because summaries of earlier works have already been reported by Bunn [8], El_Magd [1] and Gross and Galiana [22]. In this thesis the literature reviewed by Rahman and Drezga has been partly used and updated [58].
1.4.1 Quantitative methods

Quantitative methods are based on several well-established and proven mathematical techniques. These include; Multi Linear Regression (MLR), General Exponential Smoothing (GES), Stochastic Time Series (for example Auto Regressive Integrated Moving Average, ARIMA), Bayesian (Kalman filter), and State Space (STS) techniques. A few carefully selected quantitative methods have been implemented for common test data input in Chapter 3. The resulting forecast outputs are then used for comparison purposes to illustrate the accuracy of the neural network forecasting approach.

For hourly short-term load forecasting, a state space model on load and weather data input was proposed by Campo and Ruiz [10]. Different models for summer and winter were presented where each model employed different weather variables. In this approach, the parameters of the model are updated for new data using the Kalman filtering technique. Data purifying and some other special procedures are also implemented to prevent the parameters become corrupted. Using this method, an average Root Mean Square (RMS) error as a percentage of the daily peak load in the range of 1.73% to 3.61% was achieved.

An adaptive Hammerstein model with an orthogonal escalator structure has been developed by Lu et al for the relationship between load and temperature [39]. Here, matrix operations which are that commonly used in the recursive least square algorithm, are avoided. A final forecasting output with the RMS error for one-hour lead time using actual temperature data was reported to be 2.16% of the daily peak load.
Hubel and Cheng used a multi-process Baysian dynamic linear model for forecasting with lead times of 16 to 64 hours ahead [31]. Five models for each day and four different sets of data for each season were identified. For each of these models a transfer function was proposed. For model identification a statistical tool, the discriminate function, was used. The absolute relative error for this method was reported to be in the range of 1.73% to 5.92%.

Barakat et al used three classical time series methods to predict short-term monthly demands for a fast growing system with dynamic load characteristics [2]. The load data was decomposed into three sub-components; representing seasonal changes, annual trend and special events which have a "moving nature"[2]. Then, each component was forecasted separately and the final forecast was obtained by superposition theorem. Although the accuracy of the forecasted demand was comparatively better than the forecast obtained with unadjusted data, the absolute error was still rather large, ranges from 2.3% to 69.4%.

Papalexopoulos and Hesterberg attempted to developed an algorithm for daily peak load and load shape forecasting based on a complex methodology [49]. In their method, the linear regression initially was used for peak load. Some innovative model building, including accurate holiday modeling that used binary variables and temperature modeling, was introduced. Parameter estimation was performed using the weighted least-square linear regression technique. The model was evaluated under a wide variety of conditions using two years hourly data from Pacific Gas and Electric Company. The forecasting result showed an error that ranged between 1.37% to 1.94%, with a mean standard deviation of 1.62%.
A decomposition load forecasting model has been developed by Park et al [50] where the load time series was decomposed into three components; nominal load, residual load, and day type load. Components are forecasted by different approaches. For nominal load, a state-space model and Kalman filter were used, and the parameters of the model were adapted by an exponentially weighted recursive-least-square method. The type load component was extracted for weekend load forecasting based on the difference between the weekday loads and the Saturday or Sunday load and updated by an exponential smoothing method. Since, the procedure did not include the load data of preceding hours, the residual load was predicted by an auto-regressive (AR) model, and the parameters of the model were estimated using a recursive least-square method. The model was tested using data of January 1 to July, 31 1983 & 1987, this resulted in an absolute relative error from 1.40% to 2.23%.

Grady et al enhanced their original one-hour ahead, adaptive, non-linear load forecasting algorithm [39]. The main idea of the model remained the same, but the lead time was extended from one hour ahead to five days ahead. Wind-chill, temperature and humidity indices have been included for error reduction. To improve the forecasting performance, the new model employed separate models for day-time and night-time. The forecast results for five-day ahead showed an absolute relative error ranging from 3.5% to 4.2% for daily peak hours [21].

Bhattachrya and Basu reported on the implementation of a modified Kalman filter and Walsh transform to their previous load forecasting model [3]. They presented an approach for grouping the data into various subgroups, characterising each weekday separately. Then each group was replaced by its
Walsh transform while Kalman filtering was utilised for forecasting each subgroup. Finally, an inverse transform operation on each forecasted subgroup resulted in the final forecast. A 49-week historical data set was used for modeling the load for the fiftieth week and then actual data was used for model verification. Simulation results indicated an RMS error ranging from 6.11 to 10.58 MW for maximum peak of 200 MW. The individual hourly error, however, could become larger than the RMS error.

In another attempt, Mbamalu and El-Hawary used an algorithm that employs Iteratively Reweighed Least Square, procedure for parameter estimation of a seasonal auto regressive model [41]. This multi-step algorithm involved five computational steps including determining the Auto Regressive (AR) order, non-seasonal AR parameters, Inter-mediate series, and seasonal AR parameters. The results were then used for forecasting. In this attempt one month of hourly data was used for model identification and the highest average of the absolute values of forecast errors of 0.4545 was obtained.

1.4.2 Qualitative Methods

Broadly speaking, in Qualitative methods various combinations of human experiences along with mathematical analysis are used to estimate the future demand such as one day’s power need. Expert System (ES) and Neural Network (NN) are two important examples of qualitative methods.

Rahman and Bhatnagar developed a rule based system for short-term load forecasting with lead times of between six to 24 hours [60]. Correlation analysis was used to find the relevant variables such as: season, seasonal load shape, day of the week and dry bulb temperature. The knowledge, experience and logical
thinking of an experienced system operator have been emulated to form knowledge base (KB) of the expert system. Two algorithms, having some similarities, but one for one-to-six hour and the other for 24-hour ahead forecasting were implemented. The forecast absolute relative error ranges from 2.43\% to 3.48\% for six-hour ahead and from 2.42\% to 3.30\% for 24-hour ahead.

Jabbor et al developed an expert system for real time load forecasting up to 48 hours in advance [32]. Ten years historical data comprising hourly load observations and 12 weather variables were used to ensure reliable rules were obtained. This method is generally referred to as Automatic Load Forecasting Algorithm (ALFA) in relevant literature. The rule-base took into account daily, weekly, and seasonal variations of load as well as holiday, special events and load growth. Pattern recognition was used to identify similar weather patterns upon which forecasts have been made. The results showed an average absolute error of 2.2\% in a year.

Singh et al reported on an improved technique, originally introduced by Rahman and Bhatnagar [60], in [65]. Here, special considerations were given to operating conditions, sudden changes in prevailing conditions, and treatment of errors. To speed up the convergence of the required solution, sophisticated schemes were employed to select an appropriate load forecasting technique. This requires the decomposition of end user load and more detailed weather model. Then the resulting expert system was able to choose an appropriate methodology for STLF for a given input data whether related to domestic, industrial, or commercial loads.
In a utility sponsored research project, Ho et al developed a knowledge-based expert system for short-term forecasting of the Taiwan interconnected power system load [28]. Eleven shapes were established using the expert knowledge, experience of system operators and different load calculation tools. The special load types considered by the expert system included some extremely low load levels, the special load characteristics of the days following a tropical storm, the partial shutdown of certain factories on Saturdays, and the special events caused by a public holiday on a Friday or on a Tuesday. Once the expert system defined the type of day, a linear regression model was used to find the peak load for the next day. Then a normalised load curve and forecasted peak load were used to determine consumer demand for each hour of the day. The system was tested with one year’s data and an annual absolute relative error of 2.52% was achieved.

1.4.3 Artificial Neural Network

Artificial neural networks are parallel distributed models that are capable of performing non-linear modeling and adaptation without any assumptions about the model. The basic role of ANN in a load forecasting task is to provide a prediction of load demand for the next few hours or day(s). This is achieved through an off-line training process using historical data input.

The application of ANN to STLF was first reported by Park et al [51]. Their algorithm combined both time series and regression approaches in performing non-linear and adaptation modeling for forecasting. Multi Layer Perceptron (MLP), with the error back-propagation algorithm was used for short-term load forecasting of peak load of the day, total load of the day and hourly load. Different configurations with different numbers of hidden nodes were used for each case. The average absolute error of total load, hourly load, and the peak load forecasts using actual utility data, were shown to be 1.68%, 1.40% and 2.06%, respectively.
The feasibility of using a simple neural network for short-term load forecasting was investigated by Peng et al [57]. In this study, a combined linear and non-linear neural neuron in the hidden and the output layers was used. The input data consisted of an integrated load from the previous hour, and the maximum and minimum temperature for both the previous and present hours. The purpose of this system was to predict an integrated load for next day. The forecast was computed using weights that were re-estimated from recent observations. The model was tested by using load data obtained from a winter-peak load and resulted in a maximum relative error of 13%, with the majority of the errors in the range of 4% to 5%. The reason for such a significant error related to the fact that the training data set contained the holiday data which were treated as a normal day.

Lee et al has also reported on application of neural network in load forecasting [36]. In this work two distinct methods, namely a static and a dynamic method, were introduced and implemented. In the static approach a 24-hour load was forecast in one step and in the dynamic approach a 24-hour load was forecast in 24 sequential steps, using previous time forecasts. The load pattern was classified as weekday and weekend. Different neural network configurations with one or two hidden layers were tested with various combinations of neurons, and results were compared in terms of forecasting error. The average percentage relative errors of 1.89% and 1.83% were obtained for the static and dynamic methods respectively. These were comparable with the error of 1.4% obtained for the same input data using an adaptive analytical method [50].
A new strategy for data selection in the neural network approach was proposed by Peng et al [55]. In this strategy, a distance measure was used for clustering the load data into different groups to obtain an appropriate data set for training purposes. Moreover, an improved neural network algorithm was proposed in which a combination of linear and non-linear terms were used to map historical load and temperature inputs to the load forecast output. The network was tested using two years of utility data. The result showed mean absolute error of 6.2% when the holidays were included, and 5% when they were excluded.

Chen et al proposed a partially connected network in order to decrease the learning time and improve the performance for forecasting weather-sensitive loads [11]. The proposed model is able to differentiate between the weekday loads and the weekend loads. It consists of a main and three supporting networks capable of learning in cascaded fashion. The training data set consisted of the previous fortnight load and temperature data. This resulted in an average percentage peak error of 1.12%.

To speed up the training process of multi layer feed forward neural networks, an adaptive learning algorithm is proposed by Ho et al [27]. The effect of the learning rate and momentum on convergence of the back-propagation learning algorithm and the accuracy of forecast were investigated. It was found that these two factors have a significant influence on the convergence of back-propagation. They also examined the effect of the training data presentation on the convergence of learning algorithm. As a result of these studies the authors proposed a new algorithm that updates the learning rate and momentum during the training. The effectiveness of the neural network with the proposed adaptive
learning algorithm was tested on short-term load forecasting for the Taiwan electric power system. It was found that once the network is trained by the proposed learning algorithm, a desired hourly load forecast can be obtained efficiently and accurately. The simulation results showed a mean absolute error of 0.7% to 0.9%, for various seasons of the year.

Hsu *et al* proposed a combination model consisted of a feed forward and a clustering neural network [29],[30]. A Self Organisation Map (SOM) [33] was used to identify those days with similar hourly load patterns. A load pattern of the day under study was obtained by averaging the load patterns of several identical days in the past. A feed forward multi layer neural network was designed to predict daily peak load and valley load. Once the peak load and valley load and the hourly load patterns are available, the desired hourly loads can be readily computed.

Peng *et al* reported on the effectiveness of their new neural network approach for STLF [56]. They used a digital filter to decompose the load into several components. An Adeline structure [70] was allocated for each component of the load data and the Fast Fourier Transform (FFT) was employed to analyse the load components with a periodic nature. Moreover, additional modifications to the final forecast results were generated using a residual model based on the periodic components of dry bulb temperatures. A total of five Adeline networks were used to predict each hour of a 24-hour load curve in a one week interval. This approach resulted in an average absolute forecast error of 2.83%.

Other contributions on neural network application in STLF mainly cover discussions on matters related to learning techniques and network configurations [17].
1.4.4 Hybrid methods

Hybrid forecasting methods represent a combination of various techniques which produce single output for STLF. For instance, in most neural network approaches the properties of other methods (especially time series) have been employed for better modelling and finding appropriate configuration. The ARIMA model can be used to find the most highly correlated data vector for training and testing of a neural network.

Rahman and Hazim have proposed a combined knowledge-based and statistical technique for forecasting where the weather-load relationship was properly investigated [61]. In this algorithm a “pairwise” comparison technique was used to quantify categorical variables such as temperature. The least square estimation of the load was obtained using regression method. Site-specific features such as the temperature pattern, were modelled in order to customise the algorithm for a certain site with minimum effort. A monthly 24-hour ahead forecast was made for four sites resulting in an average absolute error for weekday forecast between 1.22% to 2.7%.

A combination of fuzzy expert systems [72] and neural network configurations was introduced by Lambert and et al for short-term load forecasting [35]. As the name implies, this method composes of two steps. First, an artificial neural network is trained to produce the first evaluation of a forecasted load. Then, a fuzzy expert system manipulates actual and forecasted values of real power and weather conditions to find the final forecasted load. The simulation results for a given data test showed a relative error that ranged from 0.36% to 3.76%.
1.5 Thesis Outline

This research project aims to investigate and analyse the existing short-term load forecasting methods and to implement an experimental neural network-based forecasting algorithm to be adopted for a specific site, such as the Australian South East interconnected power system. The content of this thesis can be broadly classified into three major parts.

- The first part deals with load definitions and its components. Conventional forecasting methods and their underlying theoretical principles are briefly investigated. Modeling and characterising of conventional forecasting methods in general and time series in particular are also briefly covered. This leads to a discussion on model design procedure and techniques for finding the best load model for a given load forecasting task and available input data set.

- The second part concentrates on the neural network system design and data management leading to training and testing data set. The various rules and assumptions for finding the appropriate model and configuration are explained together with the introduction of different tools needed to organise appropriate data sets associated with the neural network application for STLF.

- The third part deals with modeling and optimal design of a neural network system. This part shows how the theory developed in part I and II, can be applied to a neural network-based STLF and finding the appropriate size and types of data used for training and testing purposes.
Chapter 1: Introduction

The three parts given above are distributed over Chapters 2 and 3, Chapter 4, and Chapter 5 respectively. The followings outline a brief description of the content appeared in each chapter.

Chapter 2 provides a background for load characteristics. This includes load components in the frequency and time domains. Each load component is discussed and their associated origins and factors are explained.

Chapter 3 discusses conventional models by covering the stochastic time series method. Then a general model of ARIMA process will be presented, and finally General Exponential Smoothing is explained. Simulation results obtained from a few well-known conventional methods are used to partly evaluate the reliability of the neural network forecasts.

Chapter 4 begins with a brief introduction to Artificial Intelligence (AI) covering the Expert System and the Artificial Neural Network concepts and applications. Fundamentals of neural network and its different configurations such as Multi Layer are discussed. The design procedure of a neural network-based application in general and the use of NN for short-term load forecasting in particular are presented.

Chapter 5 illustrates how a neural network can be applied to derive explicit expressions for short-term load forecasting. The forecasting results of different ANN models for a given real data set are presented along with forecasting results obtained from carefully selected conventional methods.
Chapter 6 offers a comprehensive discussion on the application of NN to power system STLF. Simulation results obtained using real utility data here showed that the NN approach is superior to conventional methods. This chapter also suggests further work in NN-based load forecasting and draws final conclusions.

An attempt has been made to make each chapter of this thesis as self-contained as possible. The cross-referencing between chapters has been kept minimum and is mostly necessary when one wants to find out the underlying theoretical principles and the derivation of the results being used. Most of the general references are confined to the Introduction chapter and a few in other chapters.
Chapter 2

Load Components

2.1 Introduction

There are many factors affecting the power system loading condition. For instance, during a typical weekday, electricity consumption rises rapidly (almost a 30% increase) between 7 AM and 8 AM as the community prepares for work. After the increasing peak, the demand eases and flattens out as the day progresses. At the end of the working day, it experiences another sharp rise (approximately a 20% increase) between 5 and 6 PM as the community starts to heavily use household appliances. Industrial and commercial operations, on the other hand, constitute distinct demand patterns. Moreover, the difference between a typical summer and winter day peak is quite significant (nearly 50% higher in winter). For instance, the highest annual peak demand for NSW usually occurs around 6 PM in the June-July period.¹

Electricity Consumption can be broadly classified as domestic, industrial, commercial, rural and miscellaneous uses such as public lighting and traction. Some events such as sporting, religious and social occasions may change the overall demand pattern significantly. Under a normal loading condition, the annual local domestic consumption could be as high as 45% of the total energy supplied by the utility. The local industrial/commercial loads could also consume up to 50% of the total supplied energy.

¹The highest peak demand for the year 1993 supplied by pacific power, recorded on July the 4th, was 9888 MW with total daily energy consumption of 90GWhrs.
2.2 Load Curve

Normally, for studying the demand variation and comparing different classes of electricity consumption, a curve that indicates the value of the load at any time, is used. The ordinates of this curve (normally referred to as load profile or load curve) represent the magnitude of load on a system and the corresponding time. The use of load curves in the forecasting task is quite important as the accuracy of the forecast load can be evaluated using a typical corresponding load curve [9]. In expert system terminology for load forecasting, a load curve is also referred to as to “standard, nominal or reference load curve”. To illustrate a typical load curve, an average 24-hour load sequence, is presented in Figure 2.1. Each day has its own load curve and generally the load patterns are categorised on a seasonal, weekly and daily basis. The weekly load can be divided into four sub-classes; weekday, Monday & Friday, Saturday and Sunday & holiday. These different divisions are shown in Figure 2.1 where a 4-week average load is illustrated in a 24-hour time frame.

![Figure 2.1: A typical average 24-hour load sequence](image-url)
Figure 2.2: An average 24-hour load sequence for weekend

It can be seen from the figures that there are similar patterns for weekdays and different patterns for Saturday and Sunday. In the weekday pattern the Monday morning and the Friday afternoon have some differences with other weekday patterns. This classification can help to forecast each day’s pattern directly from similar historical patterns.

2.3 Load Decomposition

Periodical behaviour of the load suggests studying the load characteristic according to its periodic components. The frequency domain study of the load signal is an appropriate approach in which the signal is decomposed into different frequency components. Identifying these components, by power spectrum analysis, can help the realisation of each load component and its behaviour for forecasting tasks [4, 63]. The decomposition approach is considered as an established method for time series-based forecasting; having harmonically related frequency components [19]. The most important step of this approach is identifying different components of load in the frequency domain.
This task involves the use of the Fast Fourier Transform (FFT) algorithm which calculates the frequency spectrum of discrete sets of data points, for identifying the main harmonics of the load data. The power spectral density, a measurement of energy at each frequency corresponding to each component, is also calculated. Through spectrum analysis the load pattern can be divided into three components: a long-term trend, a daily component that varies with the day of the week, and a random component.

In Figure 2.3, a 16-day hourly load sequence is presented. This contains a large DC component that represents the average of the load series. For this reason, a mean value of load series can be subtracted from the initial load series. This allows the harmonic components of the load to be seen in a reasonable scale. The resulting power spectral density is presented in Figure 2.4.
Figure 2.4: Power Spectral Density of load data

With reference to Figure 2.4, the scaled frequency $f_s = 1.0$ corresponds to half of the sampling rate (Nyquist rate). This implies, the period of $T_s = 2$ hours, is twice of sampling period, $T = 1$ hour. Any frequency component corresponds to a time-domain series as follows. For example, the large component at frequency $f = 0.0833 f_s$ corresponds to its period $T = 1/f = 24$ hour.

As it can be seen from Figure 2.4, the FFT spectrum of load series has three distinct components. The first component of load is the base load. This constitutes an important component of the total load and occurs during the day. Its magnitude depends on the overall domestic economic activities and the climate. The second component is associated with fluctuations of the weather from normal conditions. The third component is related to variations of the load pattern through the day. Hence, with reference to Figure 2.4, the load demand at any time can be divided into the following components [56]:

1. Base load component (Figure 2.5),
2. Low frequency component (Figure 2.6),
3. High frequency component (Figure 2.7),
4. Residual component (Figure 2.8).
For simplicity, the low frequencies of the power spectrum are assigned to the base load component. The frequencies range from $f = 0$ to $f = 0.015 \, f_s$ with reference to the signal.

Inspection of Figure 2.4 reveals that a large periodic component with a frequency of $f = 0.0833 \, f_s \, (T=24)$ exits in the power spectrum. This component is defined as the low frequency component of load and represents the similarity of the load shape from one day to another. The second large frequency component occurs of $f = 0.16 \, f_s \, (T=12)$. This is related to the high frequency component of the load and represents the similarity of the load curve during a single day. Finally, the last component in the power spectrum is the residual component, and it appears at $f = 0.25 \, f_s \, (T=8)$. Although, the magnitude of the residual component is very small compared to the other components, it has to be considered in the calculation of the overall load forecasting. Each component has generally a distinct frequency and there is no frequency overlap between them. Therefore, each component can be properly filtered by means of a digital filter.

A mathematical software package, MATLAB™, was used for filtering purposes and the Chebyshev I and Butterworth digital filters were implemented on the original load series [40]. A modified time domain waveform using the inverse IFFT transformation was needed to determine each filtered component. Figure 2.5 shows the base load component and the original load. The high and low frequency load components are achieved by using two band-pass filters with $\Delta f_1 = [0.07 \, 0.1]$ and $\Delta f_2 = [0.15 \, 0.2]$ respectively. Figure 2.6 and Figure 2.7 show these filtered components from the original load data. The residual component which is the output of a high pass filter is shown in Figure 2.8.
Figure 2.5: Hourly load and Base load component (MW)

Figure 2.6: Hourly load and Low frequency components of load
Chapter 2: Load Components

Figure 2.7: Hourly Load and High frequency component of load

Figure 2.8: Hourly Load and Residual component of Load
2.3.1 Weather Sensitive Component

The influence of weather conditions on electric energy consumption is considered as one of the most important parameters in load forecasting. The weather sensitive component of a load varies in accordance with changes in some meteorological factors such as temperature, humidity, cloudiness, and wind speed. The following parameters, therefore, are to be considered during input data preparation for STLF [61]:

1. Hourly temperature,
2. The maximum and minimum temperature,
3. The 24-hour average temperature,
4. Humidity,
5. Cloud cover,
6. Wind speed,
7. The past 24-hour of items 1, 2 and 3., and
8. Rain in the past hour and past day.

Different models for expressing the relationship between the weather sensitive component of a load and weather parameters have been identified by researchers [61]. Most of these models are either complex and/or non-linear while some are simplified linear functions. For example, the weather sensitive component of a load, \( L_w \), has been expressed in a linear fashion as shown below [16]:

\[
W = K_1 \times (\text{Temperature}) + K_2 \times (\text{Light intensity}) + K_3 \times (\text{Wind velocity})
\]

where \( K_1, K_2, K_3 \) are coefficients that represent a change in the demand per unit change of the related variable from their base value (eg. \( K_1 = \frac{dW}{dT} \)). In this context, Ligesen et al assumed that the residual component of the decomposed load heavily depends on the weather variables such as temperature and wind
speed [37]. Peng et al implemented a decomposition method in which they used the filtered error of a forecasted residual as a temperature-sensitive load component [56]. This component was estimated from the error signal. The error signal was defined as the difference between the desired load value and the summation of forecasted load components including residual.

2.4 Forecasting

Once the load components identification is fully carried out, the forecast value of each component is obtained using an appropriate forecasting approach [56]. The overall forecasted load would then be equal to the inverse transformation of the sum of these components. In this regard, zero phase filtering should be used initially and then a linear summation can be applied to obtain the final forecast. As briefly explained in the previous chapter, the application of a neural network for forecasting each decomposed component was proposed by Peng et al [56]. They employed the Adeline and Madeline neural networks [20] for forecasting each component which was extracted using digital filters with a corresponding band-pass frequency. The spectral decomposition of the residual of the load data was identified, based on the periodic components of temperature.

2.5 Summary

In this chapter the load characteristics were described. The weekly load data was divided into weekday and weekend. The frequency domain analysis helped to carry out load decomposition using the Fast Fourier Transform and digital filters. Each component of a load series and its origin was identified and its frequency domain location was shown. It was found that among load components, the base load and high & low frequency load components have the most significant impact on short-term load forecasting.
Chapter 3

Conventional Methods for Load Forecasting

3.1 Introduction

One way to evaluate the performance of the proposed neural network approach to power systems STLF is to compare its forecasted output with the results obtained by conventional load forecasting methods using the same input data set. The comparison is to be made in terms of the forecasting error, computer run time, ease of modeling, and forecasting algorithm flexibility and adoption for new sites. For this purpose, the following widely used time series solutions have been carefully selected and implemented:

1) Integrated Auto-Regressive Moving Average, ARIMA, and
2) General Exponential Smoothing.

The main reason for this selection is that the general Box-Jenkins ARIMA method has been applied and is still being used in power utilities for short-term load forecasting [67, 23]. The simplicity of general exponential smoothing makes it a good reference for a comparison [44].

3.2 Time Series

Broadly speaking, a time series is a set of statistical observations arranged in chronological order. As an example, some relationships can be found between load data variables in a time horizon as the energy consumption at any time is related to the energy consumption at an earlier time. The study of these
relationships is called time series analysis. In broad terms, time series analysis deals with the methods of analysing past data and then projecting the data to obtain estimates of future values [5].

Here, a load series, \( y(t) \), is examined as the output of a linear filter with random series input [71, 5]. The theory of stochastic time series has been discussed in the relevant literature, and many papers on time series based load forecasting have been published. A brief review of this method and its characteristics, is presented here. This provides background information needed for its implementation and results interpretation.

It is assumed that the load series under study is stationary\(^1\). The load series is then viewed as being added with numerous random shocks, called “white noise”\(^2\) that is represented by \( \{ u(t) \} \). These random shocks are assumed to be independently distributed variables with zero mean and \( \sigma^2 \) variance [5]. The process can now be modeled either as an auto-regressive (AR) process or as a moving average (MA) process.

### 3.3 Auto-Correlation

The direction and strength of established relationships among observations within a time series can be measured by the auto-correlation function. For this purpose, two time series, \( y(t) \) and \( y(t+k) \), are considered as two different random variables and the corresponding correlation coefficients for these two variables are

---

\(^1\) A stationary process is one that whose mean, variance and auto-correlation function are constant through time.

\(^2\) White noise is a signal whose spectrom is constant over the entire frequency band.
Chapter 3: Conventional Methods for Load Forecasting

computed. The coefficients of the theoretical Auto-Correlation Function (ACF), $\rho_k$, are computed as follows:

$$\rho_k = \frac{\text{cov}(y(t), y(t + k))}{\sigma^2}$$  \hspace{1cm} 3.1

Where,

$$\text{cov}(y(t), y(t, k)) = E[(y(t) - \mu_y)(y(t, k) - \mu_y)]$$  \hspace{1cm} 3.2

$$\sigma_y^2 = E(y(t) - \mu_y)^2$$  \hspace{1cm} 3.3

$$\mu_y = E(y(t)) \quad \text{(mean of series)}$$  \hspace{1cm} 3.4

There are many ACFs for a given series but as a good approximation, $K < n/4$ is seen to be the maximum reliable value for ACF, where $n$ is the size of the series. In our case $K = 40$ has been chosen. It should be noted in practice that theoretical ACFs are hardly used. The Sample of AutoCorrelation Function (SACF) is found to be a good replacement for ACF in time series model identification. The SACFs are estimates of auto-correlation coefficients, $r_k$, and usually are computed as:

$$r_k = \frac{\sum_{i=1}^{N-K} (y(t) - \bar{y})(y(t + k) - \bar{y})}{\sum_{t=1}^{N} (y(t), y)^2}$$  \hspace{1cm} 3.5

Where, $\bar{y}$ is the mean of the time series. Among various tests for SACF, the test for linear association in the population between $y(t)$ and $y(t+k)$, is normally carried out. The test, $t$, is as follow:

$$t = \frac{(r_k - \rho_k)}{S(r_k)}$$  \hspace{1cm} 3.6

Where,

$$S(r_k) = (1 + 2\sum_{j=1}^{K} r_j^2)^{1/2} n^{1/2}$$  \hspace{1cm} 3.7
Usually, it is more convenient to illustrate the SACF’s behaviour in a graphical mode. Once the maximum lag, \((K)\), was specified, the graphical presentation of ACF can be shown for lags from zero to \(K\). The horizontal lines on the graph in the Figure 3.2 display the bounds \(\pm 1.96\sqrt{n}\) which are the approximate 95% bounds for auto-correlations of a white noise sequence. If the data is a large sample from an independent white noise series, approximately 95% of the sample ACFs should lie between these bounds. Large or frequent excursions from the bounds suggest that a model is needed [6]. For the historical load consumption data shown in Figure 3.1, the sample ACFs are calculated and shown in Figure 3.2.

![Figure 3.1: Two-week hourly load data](image)
The SACF can be used to find out whether the overall mean is stationary or not. If the SACF tends to decay quickly toward zero it signifies that the mean is stationary otherwise it is not. A quick decay means that the ACF should be well below the two standard error limits (upper and lower) by approximately lag 5 or 6. A sample ACF which is positive and very slow decaying, shows that the data may have a trend. A sample ACF with very slow damping periodicity shows the presence of a periodic seasonal component. For instance, in Figure 3.2 the ACF doesn’t decay quickly. This implies the presence of a periodic seasonal component of 24-hour that is related to the daily behaviour of load.

3.4 Partial Auto-Correlation Function (PACF)

Partial Auto-Correlation Function (PACF) is another useful measure of the correlation for a stationary series. It is defined (at lag \(k > 0\)) as the correlation between the residual of \(y(t+k)\) and \(y(t)\), after linear regression on \(y(t+1), y(t+2), \ldots, y(t+k-1)\). In other words, the effect of the intervening random
variables' \( y(t+k+1), y(t+k+2), \ldots, y(t+l) \) that are ignored in computing the \( r_k \) (SACF) are taken into account in this process. Figure 3.3 and Figure 3.4 show SPACF for a stationary and a non-stationary load data series respectively.

**Figure 3.3:** Partial Auto Correlation Function for stationary signal

**Figure 3.4** Partial Auto Correlation Function for non-stationary signal
3.5 Stationary ARMA Processes

The ACF and PACF processes can be used to develop an appropriate Auto Regressive Moving Average (ARMA) model for a given data. Three different ARMA modellings are considered here. These include (AR), (MA) and (ARMA) [6].

3.5.1 Auto Regressive Processes

In this method the time series \( y(t) \) is expressed linearly in term of past values of \( y(t) = \sum_{i} \phi_i y(t-i) + u(t,k) \), as shown below:

\[
y(t) = C + \phi_1 y(t-1) + \phi_2 y(t-2) + \ldots + \phi_p y(t-p) + u(t,k) \tag{3.8}
\]

\[
C = \mu_y (1 - \sum_{i=1}^{p} \phi_i) \tag{3.9}
\]

Where, \( \phi_i \) is the auto-regressive parameter and the \( u(t) \) is a random shock (noise) with zero mean and a \( \sigma^2 \) variance (white noise). The order of the AR process, \( p \), is the highest lag of the AR equation. This order can be found using the ACF and PACF. In all AR models the ACF decays to zero, and the PACF has spikes at lag \( p \). For example, for \( p=19 \), the theoretical ACF of an AR model is shown in Figure 3.5. Generally, a stationary AR model of order \( p \) has the following characteristics:

1. the theoretical ACF decays, either exponentially or with a damped sine wave pattern,
2. the theoretical PACF has a spike in lag \( p \), then all zeros.

Therefore, having the graphical representation of ACF and PACF, the order of the model can be identified according to the coefficient that exceeds the boundary limits. The order of the model will be used to find the non-zero parameters, \( (\phi_i) \), of the AR model.
3.5.2 Moving Average Process

In the moving average method the current value of the time series $y(t)$ is expressed linearly using the current and previous values of white noise $u(t)$. A common model for MA is expressed as:

$$y(t) = C - u(t) - \theta_1 u(t-1) + \ldots + \theta_q u(t-q)$$

or

$$y(t) = \Theta(B) \times u(t)$$

$$\Theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q$$  \hspace{1cm} 3.10

It is noted that a past random shock is not a past value of $y(t)$, it is the random shock component of $y(t-k)$. Therefore, an MA term represents part of a past value of $y(t)$.

The order of an MA process, $q$, is the highest lag length of the MA terms. It can be found using sample ACF and sample PACF methods. Generally, any stationary MA process has the following characteristics:

1. an MA theoretical ACF has a spike at lag $q$ then all zero (as shown in Figure 3.6),
2. an MA theoretical PACF normally decays to zero.
3.5.3 Auto Regressive Moving Average Process

Due to the Box-Jenkins contribution, the desired time series analysis is no longer limited to only auto-regressive or moving average forms [5]. A combined auto regressive moving average approach is found to be a useful general model and is considered to be a supplement to either forms. Combining Equations (3.8) and (3.10) yields Auto-Regressive Moving Average (ARMA) in the form of:

\[ y(t) - \phi_1 y(t-1) + \cdots + \phi_p y(t-p) = C + \theta_1 u(t-1) + \cdots + \theta_q u(t-q) \]  \hspace{1cm} (3.11)

Where, \( C \) is a constant formed by the mean of \( y(t) \) (that is, \( \mu_y \)) and the AR coefficients (that is, \( \phi \)'s) as shown below:

\[ C = \mu_y (1 - \sum_{i=1}^{p} \phi_i) \]  \hspace{1cm} (3.12)
For simplifying the ARMA equations a backshift operator, $B$, is defined as [48]:

$$B y(t) = y(t-1)$$  \hspace{1cm} (3.13)

In a general form, the equation 3.13 can also be written as:

$$B^m y(t) = y(t - m)$$  \hspace{1cm} (3.14)

Doing so allows the expression of any ARMA model in a backshift form. For example, an ARIMA $(1,1,1)$, $p=1$, $d=1$, $q=1$, can be written as:

$$(1 - \phi_1 B)(1 - B) y(t) = C + (1 - \theta_1 B) u(t)$$  \hspace{1cm} (3.15)

This concept has been generalised and used to express ARMA models as:

$$\Phi(B) y(t) = C + \Theta(B) u(t)$$  \hspace{1cm} (3.16)

Where, $\Theta_q(B)$ and $\Phi_p(B)$ are polynomials in $B$ of order $q$ and $p$:

$$\Phi(B) = 1 - \phi_1 B - \ldots - \phi_m B^p$$

$$\Theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q$$  \hspace{1cm} (3.17)

### 3.6 Non-Stationary Series

If a time series is not stationary, some transformation should be applied to create its corresponding stationary series. As mentioned before, a stationary process is one whose mean, variance and auto-correlation function are constant through time. When the variance of the data is not stationary it should be transformed such that a constant variance through time is the result. Noting that if the variance of a series is proportional to its magnitude, taking the square root of the series produces a constant variance. If the standard deviation of the data is proportional to its magnitude, taking the natural logarithms from data yields a new series with a constant variance.
Chapter 3: Conventional Methods for Load Forecasting

Normally, the power system load variation is not a stationary process. Figure 3.7 shows the non-stationary behaviour of a sample load series which is a two-week hourly load data set. Therefore, the natural logarithm of the data is required. The natural logarithm and square root transformation are members of the Box-Cov transformation (power transformation) [48] can be applied as follows:

\[ y(t)' = \frac{y(t)^\lambda}{\lambda} \]  

3.18

\[ \lambda = \begin{cases} 
\lambda > 1 & \text{when variance tend to fall as the magnitude of the series} \\
\lambda < 1 & \text{when variance tend to rise as the magnitude of the series} 
\end{cases} \]  

3.19

The result of this transformation for natural logarithms is shown in Figure 3.8.

Figure 3.7: Two-week hourly power system load data
Chapter 3: Conventional Methods for Load Forecasting

Visual inspection of the results after various transformations shows the individual performance of each transformation. This visual analysis works as well as some complicated mathematical methods [48]. All transformations on the data will be re-transformed at the end of the process to produce real forecasted values.

3.6.1 Differencing

If a time series does not fluctuate around a constant mean, as a stationary time series does, it should be transformed or differenced. The usual differencing is performed by subtracting two successive samples. It is, in fact, the predominant operation carried out for all cases and expressed as:

\[ w(t) = y(t) - y(t-1) \]  

3.20

If again the resulting series does not have an overall constant mean, the differencing procedure should be repeated. The second differencing or repeated first differencing can be formulated as:

\[ w(t) = w(t)^* - w(t-1)^* = (y(t) - y(t-1)) - (y(t-1) - y(t-2)) \]

3.21
Chapter 3: Conventional Methods for Load Forecasting

Usually, the first differencing is enough for most cases. The effect of differencing on the sample two-week hourly load data is shown in Figure 3.9.

3.6.2 Integrated Processes

For a non-stationary series, \( y(t) \), the same ARMA model can be used for the series (the one subjected to the differencing operation \( w(t) \)). For example, for the ARIMA(0,1,1) with \( B=1 \) \( p=0 \) and \( q=1 \) ARIMA term is written as:

\[
w(t) = C - \theta_1 u(t - 1) + u(t) \tag{3.22}
\]

Where, the parameter \( C \) is defined using Equation (3.12) where \( \mu_y \) is replaced by the mean value of the differenced series \( w(t), \mu_w \). By equating the two expressions for \( w(t) \) in (3.20) and (3.22) produces:

\[
y(t) = y(t - 1) + C - \theta_1 u(t - 1) + u(t) \tag{3.23}
\]

It should be noted that \( w(t) \) for \( t=1 \) cannot be computed.
Chapter 3: Conventional Methods for Load Forecasting

The opposite of differencing, in other words, integration, is carried out, and this accounts for the "I" in the acronym of ARIMA. Substituting $w(t)$ of equation (3.21) into equation (3.16) gives the general integrated auto-regressive moving average (ARIMA) model as:

$$
\Phi(B) \nabla^d y(t) = C + \Theta(B) u(t)
$$

where

$$
\nabla^d = (1-B)^d
$$

$$
\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p
$$

$$
\Theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q
$$

3.7 Seasonal Models

As an inherent characteristic of many time series, they exhibit a periodic behaviour in daily, weekly, yearly or other periodicities. This implies that a relationship between $y(t)$ and $y(t-s)$, $y(t-2s)$ can be expected to be observed. For example, in a power system’s load data, the load in each year normally has a similar pattern to the previous year. A seasonal pattern might also appear in a differenced, stationary series $w(t)$. Therefore, a different class of model can be defined to supplement the existing AR, MA, ARMA or ARIMA models. It is possible that seasonal and non-seasonal patterns occur together within a time series and in the SACF and SPACF. A purely seasonal process, of order $p$ and $q$, has theoretical ACFs and PACFs which is identical to those of a non-seasonal process, of order $p$ and $q$. It is noted that for a purely seasonal process the periodic patterns occur at lags $s$, $2s$, $3s$, instead of at lags $1, 2, 3, \ldots$
3.7.1 Seasonal Differencing

Observing a "wavy" pattern in a time series after regular differencing means that there is a seasonal fluctuation in the original time series. For example, in a power system load series, the daily peak load data are above the overall mean and the morning load data are below overall mean. In other words, it seems that the level of the series is shifting in a periodic fashion. By differencing from data with the lag of, \( s \), and the power of \( D \), the seasonal pattern can be removed [5].

\[
w(t) = y(t) - y(t-s)
\]

for \( D=1 \)  

In most practical cases, the value of \( D \) is assumed to be between zero and 2. \( D = 1 \) removes any large seasonal shifts in the level of a series. For example, in hourly load forecasting, \( S = 24 \), weekly forecasting, \( S = 52 \) and for monthly forecasting, \( S = 12 \) may be enough [41]. In our case, the \( S = 24 \) and \( S = 168 \) have been chosen for removing the seasonal trend. The \( w(t) \) for \( t < S + 2 \) cannot be computed because these data are not available for \( t < 2 \).

\[
w(t+s+2) = w(t+s+2) - w(B+1)
\]

3.7.2 General ARIMA\((p,d,q)(P,D,Q)\) Process.

The seasonal and non-seasonal processes can be combined to give a general form for ARIMA model. In an ARIMA model the current value of the time series \( y(t) \) is expressed linearly in terms of its values at periods \( y(t-1), y(t-2), ..., y(t-m) \) and in terms of a white noise \( u(t) \).

Having done the non-seasonal ARIMA model, the resulting series may still have seasonal and non-seasonal elements. This implies that the random shock series is in fact a correlated series, and hence more modification is required. In this case, the general form for a non-seasonal model with correlated noise can be expressed as:

\[
\Phi(B) \nabla^d y(t) = C^* + \Theta(B) b(t)
\]
Where,

\[ C^* = \mu (1 - \sum_{i=1}^{\phi} \Phi_i) \]

\[ \mu = \mu_w (d \neq 0) \]

\[ \mu = \mu_y (d = 0) \] 3.28

Now, \( b(t) \) can be expressed by an ARIMA model with seasonal lag:

\[ \Phi(B^s) \nabla_t^D b(t) = \Theta(B^s) u(t) \]

where

\[ \nabla_t^D = (1 - B^s)^D \]

\[ \Phi(B^s) = \prod_{i=1}^{\phi} \Phi_i B^{is} \]

\[ \Theta(B^s) = \prod_{i=1}^{\theta} \Theta_i B^{is} \] 3.29

Substituting Equation (3.30) into Equation (3.27) gives a combined hybrid seasonal and non-seasonal ARIMA(p,d,q)(P,D,Q) process as:

\[ \Phi(B^s) \Phi(B) \nabla_t^D \nabla_d y(t) = C + \Theta(B^s) \Theta(B) u(t) \]

\[ C = \Phi(B^s) C^* = \mu (1 - \sum_{i=1}^{\phi} \Phi_i)(1 - \sum_{i=1}^{\theta} \Theta_i) \] 3.30

if \( d = D = 0 \) then \( \mu = \mu_y \)

otherwise \( \mu = \mu_w \) 3.31

Where,

\[ \mu_w = \text{mean}(w(t) = \nabla D_s \nabla_d y(t)) \]

\( \phi = \) auto-regressive parameter,

\( \theta = \) moving average parameter,

\( B = \) back shift operator,

\( p = \) number of auto-regressive parameters,

\( q = \) number of moving average parameters,

\( d = \) number of seasonal differencing, and

\( s = \) number of periods per seasonal cycle.
3.8 Identification of Model Parameters

To implement the general model for a load series the following methodology is recommended [5]. First, the model from the general relationship in Equation (3.31) must be identified. The identification is done by examination of the sample auto-correlation function of the series compared with the theoretical auto-correlation patterns. For this purpose, the original series should be stationary. Having obtained an appropriate model from the identification process, the parameters \( \Phi \) and \( \Theta \) can be estimated. In other words, once the value of \( p \) and \( q \) have been specified, an estimation procedure is utilised for finding \( \Phi \) and \( \Theta \) by minimising the sum of squared residual term, \( a(t)^2 = E(y(t) - \hat{y}(t))^2 \), using a non-linear least square algorithm such as the Maximum Likelihood [5]. Other estimation algorithms have also been proposed for short-term load forecasting. Mbamalu and El_Hawary proposed an iteratively re-weighted least square estimation [41]. In our case, Maximum Likelihood and Least Square methods have been used [6]. Different models can be found using different \( \Phi \) and \( \Theta \) orders. The most suitable model can be selected according to AICC value. The best model, from the AICC value viewpoint, is the one with maximum AICC [48].

Having done the parameters estimation, the next step, is to examine the residual terms, \( u(t) \), to check the appropriateness of the model(s). If the fitted model is appropriate, then residual term should resemble a realisation of a white noise sequence \([0, \sigma]\). If this is not the case, the whole process, including the identification, estimation and diagnostic checking, should be repeated until an appropriate model is achieved [5].
3.9 Forecasting

One of the main purposes of time series modeling is the forecasting of future observations. Having a suitable model for any given data, then the future values of series can be forecasted. These values are the forecast of transformed series. To forecast the original series, it is necessary to invert all the data transformations that have been made to fit a zero-mean stationary model. These transformations consist of un-differencing and Box-Coved retransformation. The forecasted values using the developed model, is shown in Figure 3.10.

Figure 3.10: The actual and ARIMA-based forecast load
3.10 Forecasting: General Exponential Smoothing (GES)

The GES forecasting technique is based on the method of general exponential smoothing proposed by Brown [7]. The exponential smoothing method always produces an average where past observations are geometrically discounted according to their age [12]. It always requires an initial value of the smoothing function. In other words, the estimate is corrected with new data in proportion to the difference between the previous estimate and the new observation. This implies that the exponential smoothing is a sort of averaging tool.

3.10.1 General Model

In this technique the load data is represented as a linear combination of simple functions, referred to as the fitting function, and in addition noise components are introduced using the following expression:

\[ x(t) = \bar{a} \tilde{f}(t) + u(t) \quad 3.32 \]

Where, \( a \) is the coefficient vector, that is determined by minimisation of the weighted square error of the input data.

\[
\bar{a} = \begin{bmatrix}
  a_1(t) \\
  a_2(t) \\
  \vdots \\
  a_m(t)
\end{bmatrix}
\]

3.33

The elements of the coefficient vector are gradually changing and hence are considered constant over the lead time of the forecasting. The general form of model can be expressed as follows:

\[
x(t) = a_1 f_1(t) + a_2 f_2(t) + \ldots + a_m f_m(t) + u(t)
\]

\[
= \sum_{i=1}^{m} a_i f_i(t) + u(t)
\]

3.34
3.10.2 Forecast for load time \( \tau \)

Knowing the fitting function, \( f(t) \), and estimated coefficient \( a_i \) allows the calculation of the forecast output for the lead time \( \tau \) using the following equations:

\[
\hat{x}(t+\tau) = \hat{a}_1 f_1(t+\tau) + \hat{a}_2 f_2(t+\tau) + \ldots + \hat{a}_m f_m(t)
\]

\[
= \sum_{i=1}^{m} \hat{a}_i f_i(t+\tau)
\]

Where, \( f(t) \), the fitting function, is expressed as:

\[
f(t) = \begin{bmatrix} f_1(t) \\ f_2(t) \\ \vdots \\ f_m(t) \end{bmatrix}
\]

The fitting function can include:

1. simple mathematical functions of time,
2. empirical function, and
3. previous observations of the dependent series.

It is necessary that the value of the fitting function is known for current and future load series. The analysis of the load data determines the type of fitting function. For instance, the function for this case can be expressed as:

\[
f(t) = \begin{bmatrix} 1 \sin(w) \cos(w) \ldots \sin(w_m) \cos(w_m) \end{bmatrix}
\]

\[
w_m = \frac{2\pi}{24} k \quad (0 < k < m)
\]

Where, \( k \) is a positive number less than 168.
3.10.3 Estimating the Coefficients

The normal least squares multiple regression is used to calculate the coefficients \( a_i(t) \), so that the sum of the squares of the error is a minimum over a finite number of observations, \( T \). In other words, the discounted sum of the squared residual will be minimised according to:

\[
\sum_{j=1}^{n} \beta_j [x(t-j) - \sum_{i=1}^{n} a_i(t) f_i(t-j)]^2
\]

Where, \( \beta \) is a smoothing constant:

\[
0 < \beta < 1
\]

The other part of the process involves an \( N \)-component data vector.

\[
F(T) = \sum_{j=1}^{N-1} \beta_j f_i(t-j) f'_i(t-j)
\]

\[
g_i(T) = \sum_{j=1}^{N-1} \beta_j x(-j) f_i(T-j)
\]

Where, \( N \) is the size of recent samples. The coefficients can be updated using:

\[
a(T+1) = L' a(T) + F^{-1} f(0) \left[ x(T+1) - \hat{x}(T) \right]
\]

Where,

\[
F = F(\infty) = \sum_{j=0}^{\infty} \beta^j f(-j) f'(-j)
\]

The fitting function used in Equations (3.41) and (3.43) has to be generated by the following equation:

\[
f(t) = L f(t-1)
\]
Where, $L$ is a transition matrix;

$$
L = \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
0 & \cos w & \sin w & \cdots & 0 \\
0 & -\sin w & \cos w & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & \cos w & \sin w \\
0 & \cdots & \cdots & \cdots & -\sin w & \cos w \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix}
$$

### 3.10.4 Simulation

The general exponential smoothing technique has been applied for one-hour and 24-hour ahead forecasting, based on the observed values of the integrated hourly load demand [12]. The observed load $y(t)$ is represented as a linear combination of a function of time and noise [7]. In general, it is expressed in the form:

$$
y(t) = \bar{a}^T \tilde{f}(t) + u(t)
$$

$$
\hat{y}(t+\tau) = \hat{a}_1 f_1(t+\tau) + \hat{a}_2 f_2(t+\tau) + \cdots + \hat{a}_n f_n(t)
$$

$$
= \sum_{i=1}^{\infty} \hat{a}_i f_i(t+\tau)
$$

An initial value has to be introduced to initiate the process. The best initial value is found to be a simple average of the most recent observations. The simulation result for 24-hour load forecasting is shown in Figure 3.11
3.11 Summary

In this chapter, the theory and some practical information for two selected quantitative models, Integrated Auto Regressive Moving Average (ARIMA) and General Exponential Smoothing (GES), have been described. It has been found that for a time series model development the auto-correlation function and the partial auto-correlation function are very important in finding the order and the type of the required model. The effectiveness of each model is related to the model order approximation. This constitutes the first and the most important step in time series modelling. Two selected quantitative models have been developed and tested. The resulting forecasted output for a set of test data is shown in Chapter 5. A comparison with the result obtained by the NN approach is carried out and is presented in Chapter 6.
Chapter 4

Artificial Neural Networks

“The post industrial society will be fuelled not by oil but by a new commodity called Artificial Intelligence (AI). We might regard it as a commodity because, it has value and can be traded” [34].

In answer to the question of “What is AI?” one view says AI is an attempt to answer central question about the use of thoughts, feelings, and consciousness. While certainly very complex, our brains can be considered to be governed by the same physical laws as our machines. Viewed in this way, the human brain may be regarded as a very capable machine. Conversely, given sufficient capacity and the right techniques, our machines may ultimately be able to capture human intelligence [68]. The AI may also be defined as an attempt by computer scientists to create computer programs capable of solving problems associated with human thought [68].

The human brain is more intelligent than computers. For instance, computers are very bad at performing simple visual tasks; whereas it would be a simple and perfectly logical function to humans. This problem and other functionally similar problems that computers are incapable of performing well, have been tackled using artificial intelligence during the last three decades.
4.1 Knowledge-Based Expert System (KBES)

As a typical example of the AI, expert systems (ES's) were developed by computer scientists and technologists to capture some aspects of human intelligence. Generally, such a system consists of [34]:

1. a database of facts related to a particular discipline,
2. a knowledge base of confidently organised rules for drawing inferences from the databases, and
3. a high-speed inference.

Having an expert system for a specific job requires the knowledge of an expert human in that specific field. It also needs a Knowledge Engineer (KE) to structure and formalise this information from expert(s) and to represent them in the form of facts and rules (for example, IF-THEN production logic). This representation is called the Knowledge Base (KB) component of the expert system. The search for a solution through the KB and reasoning is done by the Inference Engine component of the expert system. A general procedure for building an ES is shown in Figure 4.1. In this procedure, the required database is gathered through the formalisation of knowledge from human experts and domain information by the KE. Then the inference engine including the search methods, should be selected. Finally the design and implementation of an appropriate user-interface for the expert system is carried out.
4.1.1 Artificial Intelligence in Power Systems

In recent years, the application of AI in power systems has attracted much attention. The use of an expert system for short-term load forecasting is one example in which the knowledge of an expert in the field of load forecasting is formulated to make accurate forecast of future demand. A literature review carried out by Zhang et al [73] showed that until the early 1988 only 8% of expert systems were used in power system applications. The ES based approach for short-term load forecasting was first developed by Rahman and Bhatnagar [60]. Other researchers have also published papers using a similar approach(es) for short-term load forecasting [32, 59, 60, 65, 73].

As published literature shows, most of the applied expert systems in power system load forecasting employs models based on a reference day's load curve. Then, this curve is reshaped according to a set of rules which are affected by the
expected variations in the forecasted day, from that reference day to the next. Moghram and Rahman reported the successful application of ES for lead forecasting [60]. They employed a technique consisting of two parallel approaches. The weather variables, day types and diurnal effects were identified using expert system information. These parameters and hourly load data were used for generating the weighted average load forecast which also used some statistical techniques including the transfer function. Then, a purely rule based expert system driven by knowledge from electric utility experts, was used for final load forecasting [60]. In this work the load at hour “h” of the forecasted day is calculated as follows:

\[
y_h^{FF} = y_h^{F1} + \Delta y_h
\]
\[
y_h^{F1} = y_h^{Rm} + \Delta y_h^{in}
\]
\[
\Delta y_h^{in} = (y_{00}^T + y_{00}^R) \times (24 - h) / 24
\]
\[
\Delta y_h = \pm \Delta t \times F1 \times F2 \times 25 \quad dt \leq 10^\circ F
\]
\[
\Delta y_h = \pm \Delta t^2 / 4 \times F1 \times F2 \quad dt < 10^\circ F
\]

where

- \(y_h^{FF}\) = forecasted load at hour “h” of the day
- \(\Delta y_h^{in}\) = load correction due to inertia at hour “h”
- \(y_{00}^T\) = load at hour 00 of the target day
- \(y_{00}^R\) = load at hour 00 of the reference day
- \(h\) = hour of the day for which forecast is sought
- \(y_h^{F1}\) = first level forecasted load at hour “h”
- \(y_h^{Rm}\) = reference day for hour “h” modified to account for the day to day variations
- \(\Delta t\) = ambient (or effective) temperature difference between hours in forecasted and the reference day.
- \(F1\) = weighing factor that accounts for the relative change in temperature between the forecasted and reference day’s
- \(F2\) = weighing factor accounting for the different reference days temperatures
As can be seen, the parameter $F_2$ should be updated for each day. The expert system automatically plays a role in updating these equations and factors. The short-term load forecasting based expert system, however, presumes the existence of an "expert" capable of making accurate forecasts who will train the system. Moreover, because of its programming and knowledge transformation costs compared to other methods, the ES is found to be very expensive. It is also not always easy to express human expertise in a clear organised format. This may lead to an inconsistent set of rules.

4.2 Artificial Neural Network Modeling

4.2.1 Definition

"Artificial Neural Networks (ANN's) are massively parallel interconnected networks of simple organisations (processing elements) which are intended to interact with the objects of real world in the same way as the biological neural systems do" [33].

These parallel distributed models are potentially capable of performing non-linear modeling and adaptation without any assumptions about the model. In very broad terms, the ANN may be defined as an attempt to capture the human brains capabilities for solving complex problem.

The term "Artificial Neural Network" is used to describe a number of different models intended to imitate some functions of human brains. They are also represented with highly interconnected networks, so the name "connectionism" has also been used to refer to this field of study.
Chapter 4: Artificial Neural Network

The capabilities of neuron models (Perceptron) in calculating certain logical functions were shown by McCulloch and Pitts in 1943 [42]. Many researchers have sought to create mathematical models to imitate some of the human brain's functions. Since then, theoretical limits associated with neural networks were demonstrated by Minsky and Papert [43]. This caused researchers to lose their interest in neural networks for a while. Since 1985, some new mathematical models have been developed resulting in the complete elimination of the original neural networks theoretical limitations [62]. This has caused the current intensive research activities in this promising field.

4.2.2 Structure

4.2.2.1 Artificial Neuron

An artificial neural network consists of a number of artificial neurons that are connected through variable weighted connections. Each simulated neuron, as shown in Figure 4.2, is a single processing element that acts on the input data to produce an intermediate result. Typically, neurons are arranged in layers and connected to the nodes in the proceeding layer for input and the following layer for output. They are referred to as nodes, units, or processing elements (PEs) [20]. Each PE consists of a number of inputs and only one output. Figure 4.2 shows a schematic representation of a general PE model. Each unidirectional connection to the $i^{th}$ PE is associated with a numeric value called "weight" or connection strength. When the weighted sum of inputs (the net inputs) to a PE exceeds a certain threshold, the PE is fired and the output signal is produced.
The output of a PE is a function of the weighted sum of the inputs to that PE. This function is normally referred to as the activation function. The most commonly used activation functions, including linear, hard limit, hyperbolic tangent, and sigmoid are shown in Figure 4.3.

![Simulated neuron](image)

**Figure 4.2:** Simulated neuron

![Common activation functions](image)

**Figure 4.3:** Common activation functions

### 4.2.2.2 Multi Layer Perceptron

In a neural network, input patterns (data signals) are connected to the processing elements of the input layer and the outputs of these elements are connected to the inputs of the elements in the next layer [42]. There is one PE for each input in the
input layer and one PE for each output in the output layer. In addition, a few arbitrary hidden layers may also be inserted between the input and output layers. The resulting network configuration is shown in Figure 4.4, and is known as the Multi Layer Perceptron (MLP). The use of more hidden layers permits better handling of more complex non-linear functions.

![Multi Layer Perceptron Network](image)

**Figure 4.4:** Multi Layer Perceptron Network

### 4.2.2.3 Learning process

A neural network is trained to learn the relationship between input and output data. Learning is achieved through a learning rule that adapts the connection weights of the network when a set of input test data and corresponding desired output data are made known for the network. Generally, in terms of the training procedure and data presentation, the learning process is classified as unsupervised or supervised learning [20].
In an unsupervised learning process, only input data (unlabelled data) are made known to the network and each hidden processing element internally responds strongly to a different set of input or a closely related group of inputs. These sets of inputs represent clusters in the input space. Typically, these clusters may represent distinct real world concepts. One of the most commonly used unsupervised networks is the Self Organisation Map (SOM)[33]. The unsupervised learning feature of the SOM is used in combination with other networks to implement tasks such as prediction and classification. Here, the network first learns in an unsupervised mode and is then switched to a supervised mode.

In supervised learning, a desired output is presented for each input, and the network gradually configures itself to achieve that desired input/output relationship [46].

4.3 Neural Network Capabilities and Limitations

4.3.1 Capabilities

The general interest in neural networks arises from their fascinating features that enable them to overcome some of the limitations of conventional information processing systems; such as the need for detailed programming. They have some inherent distinct capabilities unavailable in other methods. A brief listing of these capabilities may help to further understand their operational characteristics [15].

**Parallelism:**

This is a prominent feature of any neural network configuration when they are considered as sets of processing elements operating simultaneously.

**Adaptation:**

This allows the neural network to take account of any additional input data to achieve improved performance. For example, in load forecasting some additional inputs, such as humidity, can be included in the input data set for training and recalling.
**Distributed Memory:** Unlike serial computers which have memory components, a neural network memory is associated with the weights of connections between processing elements. This is interpreted as distributed memory.

**Fault Tolerance:** This is a direct outcome of the distributed memory. Memory lost in conventional computers result in unacceptable output, while losing a connection weight (distributed memory) in NN configuration still offers a valid output [15].

**Generalisation:** A neural network can learn a general rule from the training patterns that maps the input/output relationship.

**Ease of Construction:** Neural networks are easily constructed on serial or parallel computers.

### 4.3.2 Limitations

Major limitations of neural networks are listed below. These shortfalls have slowed down the wide acceptance of neural networks in some applications.

**Lack of Parallel Hardware:** Neural network methods are well suited to be adopted on parallel computers. Simulating parallelism on serial computers will cause operational problems such as extensive run-time for training. The size of the problem grows in proportion to the size of the input data and neural network complexity.

**Input Data Pre-processing:** The quality of a neural network output is heavily related to the quality of its input data. This makes necessary the use of pre-processing operations such as filtering and transforming of the input data set.
Unclear Design Method: There is no specific rule to configure the desired network for a given problem. For instance, in a MLP network the number of hidden nodes and layers are not known. Generally, this problem is tackled through a trial and error procedure.

Lack of reasoning: It is impossible to substantiate the results that are obtained from a neural network. In fact, a neural network operates as a "black box" and hence it would be difficult to rely on their outputs. Normally, the quality of neural network outputs are being assessed by means of some statistical methods.

4.4 Applications

Neural networks are beginning to interest a large audience of researchers and engineers. This includes researchers in the field of power systems. Many ways by which artificial neural networks can be applied to power system problems are discussed by numerous researchers worldwide [17]. The early application of neural network in power systems was reported during the late 1980's, due mainly to the contribution of Sobajic and Pao [66].

The major applications of neural network in power systems are categorised into the following three main areas [17]: Regression, Classification and Combinatorial optimisation. Alternatively, they can be classified according to the power system problems as follows:

1. Economic Dispatch and Unit Commitment,
2. Fault and Protection,
3. Identification and State Estimation,
4. Load Forecasting,
5. Monitoring, Observability and Diagnosis,
6. Operation and Planning,
7. Power Quality,
8. Security Assessment and Control, and
Furthermore, problems with the following characteristics are envisaged to be well suited for neural network oriented solutions.

1. the rule used in solving the problem may be unknown or very difficult to explain or formalise,
2. the problem makes use of noisy data,
3. the initial conditions of the problem are varying,
4. high speed processing is needed, and
5. there may be no current technical solution.

4.5 Neural Network Based Load Forecasting

As was stated before, neural networks with their useful properties can perform do many functions including load forecasting. In load forecasting applications, the basic role of an ANN is to provide a prediction of power system demand for the next few hours, day(s) or week(s).

There are a number of good reasons to substantiate the superiority of a neural network-based load forecasting over conventional forecasting techniques. First, there are no specific rule and/or transfer function required to describe the relationship between the load variation and other parameters such as weather information. In the case of the expert system approach, the extraction of rules and knowledge from the relevant expert operators is very expensive and time consuming. Second, the data used in load forecasting is usually noisy and uncertain. Third, the performance of forecasting is very sensitive to the initial conditions such as the last day(s) or hour(s) temperature. Finally, because of the parallelism feature of a neural network, high speed data processing can be achieved.
4.5.1 Neural Network Model Design

To design a neural network model the following steps have to be taken;

1. defining the type of network,
2. selecting an appropriate training procedure,
3. choosing the required network configuration, and
4. preparing the initial values of the training parameters.

The suitability of the problem for solution by a neural network should be determined according to the characteristics described in Section 4.4. Then, the type of learning procedure, supervised or unsupervised, should be specified based on the nature of the problem. For instance, for a classification problem an unsupervised network is recommended for most cases. On the other hand, if there is a desired output set for each input set, the network with a supervised learning algorithm is suitable. Generally, for a given network, the process must have a large number of data, enabling learning to take place.

Regarding network configuration and the type of implementation, several well-known neural network models that have already been developed and used in many applications and therefore may be consulted. These include Back-Propagation, Hopfield, Kohonen (SOM), and Adeline [20]. There are no hard-and-fast rules for choosing an appropriate network for a particular application. A few general guidelines may be observed in the selection of the required neural network topology. In some cases, the size of the problem determines the possibility of dividing it into several sub-problems with different type of networks. For example, in a load forecasting problem, the load data can be classified into several categories using clustering neural networks such as SOM, then load forecasting using a MLP network can be implemented for each category.
Input data preparation and pre-processing constitutes an important step in neural network design. The following section covers this aspect of neural networks in some details.

**4.5.2 Data Pre-processing**

The first and perhaps the most time consuming step in developing a neural network is data pre-processing. Transforming and extracting meaningful information from the raw data requires considerable extensive time and in some cases a specialised data processing tool. This process may take up to eight-tenths of the whole design and implementation task [46]. Pre-processing, on the other hand, means transforming the data so that it becomes easier for the network to learn the input/output relationship. Pre-processing may involve mathematical operations such as normalising, ranking, and statistical operations such as correlation and skewness. Generally, the main goal here is to make a file containing a series of sample input patterns. The familiarity with the application can help in collecting the relevant data. For example, in short-term load forecasting, the load pattern always is a function of past and future load and temperature data. Therefore, the input data file should contain the most correlated past load and temperature data in a proper style and format.

The first step in data processing is to obtain accurate and relevant historical data, and then make it suitable for neural network training. This is a recursive step. In other words, when one set of data are collected then the statistical analysis may suggest that another data set is also necessary.

Sometimes, a small subset of available data is enough to train a network successfully. Usually, the seasonal effects, trends, and other relationships can be refined so that the network learns faster and easier. In load data, for example, the seasonal components can be identified and filtered or modified from data sets. This, it makes the network converge faster [24, 25].
Number of data points:

The number of data points is a critical factor, and often relates to practical concerns such as the cost of data gathering. The amount of data for training depends on many factors. The quality and accuracy of the data is very important especially when the amount of data is very low. The cost of data gathering specifies the total modeling cost. Usually, a large training set of data reduces the risk of under-sampling of the underlying function. On the other hand, a small number of input may affect the capability of the neural network to generalise the input/output relationship.

Network Size:

The network size is another factor in choosing the number of training data points. The size of a network is influenced by the complexity of the function(s). On the other hand, a more complex application seeks more a complex network with more hidden layers, nodes and more input contributors. A complex and large network may be trapped in local minima in the error surface during the learning process, and does not converge to a global minimum. For instance, in short-term load forecasting, 24 nodes may be used to represent the indices of the hour of the day. This design could leave the network trapped in a local minimum during the learning process as that no convergence occurs toward the global minimum. Therefore, some other alternatives such as coding for the hour of the day should be used to decrease the size of network.

\[\text{\textsuperscript{1}}\text{The Error Surface or Function is defined as the network error. It is assumed the existence of a global error function without actually specifying it. The main target in learning is to determine the changes in weights in order to decrease the global error.}\]
In practice, the required size of data depends on several factors such as network size, testing needs, and the cost of data gathering. However, as a rule of thumb, having five to ten training patterns for each weight seems adequate [25]. A detailed study of this step, known as sensitivity analysis, will be presented later in Chapter 5.

Noisy Data Set:

The network can learn using a small, noisy, or skewed training data set. This, however, may result in an unreasonable range of error. If the network is trained to perform in a noisy environment, the inclusion of some noise to the input vectors during training helps the network to converge.

Normalisation:

The last step of data pre-processing involves the normalisation of the input data set. This is done to prevent the simulated neuron being driven too far into the saturation region. Here, the minimum and maximum values of variables are mapped to the “low values” and “high value” in the variable range respectively. Then, all other values are mapped linearly between the “low” and “high” values. The normalisation procedure can be expressed mathematically as follows:

1. \[ \text{SCALE} = \frac{(\text{HIGH} - \text{LOW})}{(\text{MAX} - \text{MIN})} \]
2. \[ \text{OFFSET} = \frac{(\text{MAX} \times \text{LOW} - \text{MIN} \times \text{HIGH})}{(\text{MAX} - \text{MIN})} \]
3. \[ \text{Xadj} = \text{SCALE} \times \text{X} + \text{OFFSET} \]

where, MAX= 0.9 and MIN= 0.1. The missing data can be mapped to the middle of the range; i.e., \( 0.5 \times (\text{HIGH} + \text{LOW}) \).
Data Set for Testing and Training

Testing requires an independent data set. The network should be tested using another data set, that is one that has not been used in training. A simple way to prepare a test data set is to divide the available data into two parts, two-third for training and the rest for testing [25]. Providing this division of data processing enables the designer to choose an appropriate network configuration. Network testing and training will be explained in detail in section 4.5.2.2.

Input Data Set for Power System Load Forecasting

To find a proper input data set for training and testing of the proposed neural network, several approaches have been implemented. The auto-correlation function (ACF) and partial auto-correlation functions (PACF) have been found to be appropriate tools for this purpose. Investigation of the ACF and PACF shows that there is a strong correlation between load consumption for particular hours in a day and the corresponding data of the previous day. The same correlation exists between days in a week and the previous week [11].

Generally, from a power utility viewpoint, the daily load variations are classified into two major profiles; weekday and weekend. Therefore, the index for the day of a week may be chosen as an additional input. The daily load variation also strongly depends on the hours of a day. The hour of the day index is then used as an input.

In short-term load forecasting the way data is presented to the ANN is also important. In the case study, seven neurons are used to indicate the day of the week. To represent Friday, the corresponding input node is set to “1” while the other (representing Saturday to Thursday) are set to “0”, as shown in Table 4.1. This table also shows other inputs; twelve historical load input data, seven
indices for the type of day and five binary inputs for hours of the day. This representation has shown the best performance compared to other arrangements [11, 26]. Furthermore, to differentiate between weekday and weekend load profiles different values for Saturday and Sunday were used.

4.5.2.1 Neural Network Architecture

Since there is no single approach for choosing the best NN configuration for given data, finding the best network architecture is commonly based on a trial-and-error process. Sometimes, the accessible data dictates the choice of network. For example, in our case study, the only available input data input was hourly load and temperature for a specific period of time. This limited the network configuration to have only these two types of inputs.

<table>
<thead>
<tr>
<th>Input No.</th>
<th>Load data</th>
<th>Input No.</th>
<th>M</th>
<th>T</th>
<th>W</th>
<th>T</th>
<th>F</th>
<th>S</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Load (d, t)</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Load (d, t-1)</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Load (d, t-2)</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Load (d-1,t)</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Load (d-1,t-1)</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>6</td>
<td>Load (d-1,t-2)</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>Load (d-7,t)</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>Load (d-7,t-1)</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>Load (d-7,t-2)</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Load (d-8,t)</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>Load (d-8,t-1)</td>
<td>23</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>Load (d-8,t-2)</td>
<td>24</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.1: List of inputs data for proposed neural network
Chapter 4: Artificial Neural Network

If target results are not known, a supervised algorithm will be useless. This leaves the unsupervised algorithm such as Self Organising Map (SOM) as the only available option. Now, questions such as "How many and what combination of nodes and layers are needed to solve a particular problem" should be answered. Usually, there is no specific rule for tackling these questions for a given problem. The networks with too many hidden nodes tend to memorise the training data and those with too few cannot learn to respond to the problem accurately.

Generally, for load forecasting, three layers are sufficient, but sometimes a problem seems to be solved easier (that is faster) with more than one hidden layer. The only certain parameter is the number of neurons in the output and input layers. A simple and useful procedure is to start with a small number of hidden nodes. Then increase the number of nodes while the performance of the training is controlled by testing the network at each step. It is possible to remove hidden neurons that are superfluous later. This refers to those nodes associated with connected weights that have had very little change from their starting value. These nodes can be removed or disabled. There is also an automatic method for "pruning" the superfluous nodes from network [62].

One example of the most successful NN configuration used in power system applications is the feed forward Multi Layer Perceptron (MLP) with the Back-Propagation (BP) learning algorithm [62]. This configuration has been widely used for different power system applications in general and load forecasting in particular. This configuration is relatively simple in topology as well as learning algorithm. A detailed explanation of the Back Propagation algorithm is given in Appendix A. Figure 4.5 shows a schematic illustration of the proposed feed-forward network.
4.5.2.2 Training and Testing

The final development stage is the training and testing of the developed network. To train a network, the historical input and output training patterns are shown to the network repeatedly until an error criterion such as RMS is fully satisfied.

The network may memorise the training examples rather than learning the input/output general relationship. The real goal during training is to reach the maximum test-set accuracy while avoiding the data memorising problem. Testing during training shows when to stop training to prevent inappropriate memorisation; this is commonly referred to as "overtraining". It also shows which configuration is the best. Neural networks are useful only if they return appropriate results with data not used in training. A good technique for preventing NN from overtraining is to stop when the resulting mean square error of the testing data set stops improving [24, 25].

Figure 4.5: Neural network configuration for short-term load forecasting
4.6 Summary and Conclusion

In this chapter, a brief review of expert systems and artificial neural network was presented. The artificial neural network and its underlying design and operational considerations were discussed in more detail. A general discussion on the neural network capabilities and limitations was presented. The design procedure for neural networks was explained in some detail. Based on this procedure, a general configuration of neural networks, namely multi layer perceptron, was found suitable for power system short-term load forecasting. Finally, a brief background of the proposed neural network configuration was presented and various considerations on data pre-processing were given. Complete neural network development and relevant forecasting results will be shown in the next chapter.
Chapter 5

Forecasting Results and Comparison

The conventional and neural network based approaches to power systems short-term load forecasting have been studied in previous chapters. In this chapter different neural network configurations, specifically designed for one-hour to 24-hour ahead load forecasting, are examined. To obtain a meaningful forecast it was necessary to use real hourly load data and temperature data. Different possible neural network models were trained and tested using similar input data sets. A detailed description of training and testing procedures are also presented in this chapter. The sensitivity analysis of the neural network to changes in the input data set was investigated and results are shown. Identical input load data and temperature data were used in conjunction with two conventional forecasting methods; namely "auto regressive moving average" and "general exponential smoothing". The corresponding forecasted results were used to further assess the quality of the proposed neural network schemes. A comparison of the results together with other forecast quality assessment results are given in this chapter.

5.1 Data Collection

Historical load data including hourly active power consumption, voltage at bulk supply, and reactive power requirements for winter and summer periods was gathered from the local electric power utility. The collected load has a maximum and minimum load of 10,000 and 3000 MW respectively. The local power utility
has also supplied the required temperature data. The maximum and minimum hourly average temperatures were 35°C and 6°C respectively. The available load data has a winter peak demand with a yearly peak in June.

5.2 Model Development

Several back propagation learning algorithms are adopted and tested for the proposed neural network. Using heuristic methods, the input layer is designed to contain enough processing elements (nodes) to match the number of crucial input variables. Input layer nodes can be configured to accept any combination of the following crucial input variables:

1. Past day(s) average temperature(s),
2. Past hour(s) temperature(s),
3. Temperature forecast for future hour(s),
4. Indices of the hour of the day,
5. Indices of the day of the week,
6. Past hour load data (active power),
7. Wind velocity,
8. Rain/ humidity, and
9. Ambient light conditions.

Data availability as well as its collection costs could impose constraints on the choice of input information for the input layer. Clearly, one that can not be ignored is the load data. It covers the past hourly and daily average load. Because of the different shapes of the load profile for different days of a week, the indices of the day of a week have to be used as unique indicator. Similarly, the indices of the hour of a day are crucial.
The performance of an ANN model depends not only on the input variable selection but also on the network dimension. The network size is proportional to the number of input and output nodes. In turn, the number of nodes in the input layer is found according to the most correlated historical load data and temperature. In order to reliably deriving the necessary historical load data, the correlation between the elements load data were analysed. In particular, the auto-correlation and partial auto-correlation function of the hourly load time series were used in finding the correlation of the current hour with the past hour’s load. It was found that the next hour load is affected by the past hour’s load and the pattern through which the current load is included. For example, Monday’s load profile is affected by Sunday’s and Saturday’s loads and also the previous Monday load. A moving time window was used to select the required data for network training. After collecting a sufficient number of data points, it was necessary to divide the resulting data set into two sub-sets namely; training and testing.

Different combinations of the following learning methods were chosen and tested to find the most appropriate neural network model for one-hour to 24-hour ahead load forecasting:

1. Various back propagation learning algorithms including Cumulative Delta rule, Normalised Cumulative Delta, and Delta rule [20],
2. Discrete activation functions: Sigmoid, Tangent Hyperbolic, and Sinusoidal functions,
3. Different slopes for activation functions (from 0.1 to 10),
4. Several connection formats: full and partial network connection,
5. Different number of hidden nodes
6. Separate number of training epochs, and
7. Different data presentations (sequential or random) during training
A great deal of time was devoted to ensure all possible options, mentioned above, were thoroughly considered. This was followed by an intensive effort for algorithm development, network implementation, testing and preliminary result collection. It was found that the following package of system components has a good potential to offer the most accurate short-term load forecasting for the given site and its available input data.

1. Tangent Hyperbolic is used as the activation function.
2. Variable learning rate is used for learning algorithm.
3. Random presentation is used to introduce the input data set to the network.
4. Normalised cumulative back propagation is employed as the learning algorithm.

Moreover, in terms of the forecasting lead time, two neural network configurations are suggested to achieve one-hour and 24-hour ahead load forecasting. The relevant neural networks proposed by other contributors are also briefly investigated and the similarities and differences are discussed.

5.2.1 Operating Mode I: One-Hour ahead Forecast

In a one-hour ahead forecasting network, one node in the output layer is used for presenting the next-hour load, and the load forecast has to be propagated on an hourly basis. This implies that load at time (t+1h) is used in forecasting the next hours load, (t+2h). This procedure continues for all hours of the forecasted period. The input data for this type of forecasting was found by using the partial auto-correlation function on an hourly load series. It contains the current and two subsequent hourly load data, the same three hours from the day before, and finally the same hours from the previous week. Considering the type of connection, two groups of networks namely, fully connected and partially connected network, are presented for one-hour ahead load forecasting.
5.2.1.1 Partially Connected Neural Network

The original idea of this configuration was proposed by Chen et al [11]. It consists of one fully connected main network with a few auxiliary, supporting partially connected networks. The main network provides a basic forecasting value. The other auxiliary networks provide a better correlation between the forecasting load the other influential parameters. Figure 5.1 shows a schematic representation of a partially connected network.

In this configuration, the input layer has twenty nine input nodes. These include twelve correlated past load data (obtained using the PACF analysis), the current and two subsequent hourly temperatures, seven indices for the weekday and finally five coded indices of the hour of the day. Table 5.1 summarises all input data variables for this configuration. As was previously mentioned, there is no specific rule to find the appropriate number of hidden nodes or the number of hidden layers. Hence, the corresponding numbers were found through a trial and error process.

The temperature data gathering has always been one of the most difficult tasks in load forecasting. A reliable temperature forecast and actual temperature database are normally very rare. Thus, a network that can forecast the hourly load without temperature data would be extremely useful. Therefore, two types of network, with and without temperature input data, are proposed here.
Chapter 5: Forecasting Results and Comparison

Figure 5.1: Partially connected neural network configuration

<table>
<thead>
<tr>
<th>Input No.</th>
<th>Load data</th>
<th>Input No.</th>
<th>Indices day of a week</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Load (d, t)</td>
<td>M T W T F S S</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Load (d, t-1)</td>
<td></td>
<td>1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>3</td>
<td>Load (d, t-2)</td>
<td></td>
<td>0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>Load (d-1,t)</td>
<td></td>
<td>0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>Load (d-1,t-1)</td>
<td></td>
<td>0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>6</td>
<td>Load (d-1,t-2)</td>
<td></td>
<td>0 0 0 0 1 0 0</td>
</tr>
<tr>
<td>7</td>
<td>Load (d-7,t)</td>
<td></td>
<td>0 0 0 0 0 0 0.5</td>
</tr>
<tr>
<td>8</td>
<td>Load (d-7,t-1)</td>
<td></td>
<td>0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>9</td>
<td>Load (d-7,t-2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Load (d-8,t)</td>
<td>Indices of Hour of a day</td>
<td>1 2 3 .. .. 23 24</td>
</tr>
<tr>
<td>11</td>
<td>Load (d-8,t-1)</td>
<td></td>
<td>0 0 0 .. .. 1 1</td>
</tr>
<tr>
<td>12</td>
<td>Load (d-8,t-2)</td>
<td></td>
<td>0 0 0 .. .. 0 1</td>
</tr>
<tr>
<td>13</td>
<td>Temp (d, t)</td>
<td></td>
<td>0 1 1 .. .. 1 0</td>
</tr>
<tr>
<td>14</td>
<td>Temp(d, t-1)</td>
<td></td>
<td>1 0 1 .. .. 1 0</td>
</tr>
<tr>
<td>15</td>
<td>Temp(d,t-2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>AveTemp(d-2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>AveTemp(d-1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: The input data for one-hour load forecasting Mode I
5.2.1.1.1 Temperature Included

The daily load peaks are influenced by the most recent and the average daily temperatures. The recent three-hour temperature data corresponding to the recent three-hour's load data are considered as input. The heat accumulated inside a building, as the temperature increases slows down the load variation. If the temperature is decreasing on a winter's day, the accumulated residual heat slows down any sharp increase in the load [61]. To consider this effect the average load temperature for the two days before the forecasting day are taken into account. This increases the temperature data input numbers to five variables. In the context of the learning algorithms, two variations for this configuration can be envisaged. These are cascade and normal learning algorithms.

**Cascade Learning:**

In cascade learning, the network is first trained by the whole input data. The fully connected hidden neuron's weights are then allowed to be changed in the training procedure. For this purpose the auxiliary hidden nodes (for example week, hour and temperature) are disabled during the first stage of learning. In the next stage, all of the hidden neurons are trained with a low learning rate.

**Normal Learning:**

In normal learning, the network is trained with data repeatedly from the starting point and all the hidden neurons are participated in the training.

5.2.1.1.2 Temperature Excluded

This configuration is the same as the partially connected network with temperature data mentioned in section 5.2.1.1.1 except that the five temperature input data are excluded from the input data set.
5.2.1.3 Chen et al proposed network

The original partially connected network structure proposed by Chen et al [11] has also been implemented and tested. This structure is shown Figure 5.2. Here, a connection exists between the current node activated by the current input load data and the output node. There is also an additional connection between the current temperature input and the output node. The forecasted result obtained from the original structure implementation will be used for comparison purposes.

Figure 5.2: Partially connected network proposed by Chen et al [11]

5.2.1.2 Fully Connected Neural Network

A fully connected configuration has also been developed and implemented to measure the effectiveness of the partially connected network. Similarly, as for the partially connected network, the temperature data can either be included or excluded from the input data set. This results in the following two types of network.
5.2.1.2.1 Temperature included

Figure 5.3 shows a fully connected network using temperature data. In this configuration all the input nodes are connected to nodes in the hidden layer. Therefore, there is no difference between nodes in the hidden layer as was the case in the partially connected networks. This implies that there is no need to employ a cascade learning procedure, hence a normal learning procedure is adopted to train the network. Furthermore, the input data set for this network is identical to the one used for the partially connected network covered in section (5.2.1.1.1).

5.2.1.2.2 Temperature Excluded

Similar to the partially connected network, the temperature data can be excluded from the input data set. Here, the input nodes are connected to all nodes in the hidden layer and a normal learning algorithm is applied to train the network.

5.2.2 Operating Model II 24-hour ahead forecast

As an alternative strategy for short-term load forecasting, it is possible to get a 24-hour ahead forecast result in one using the one-hour ahead forecasting concept explained previously. In 24-hour ahead forecasting, the network will forecast the next 24 hour’s load using the last 24 hours of load data and the past and future temperature data. These models have 24 output nodes where each node represents one hour of the 24-hour ahead load forecast. Several neural network configurations have been designed, implemented and tested. The main objective was to find the best network that gives a minimum error. The average temperature of the past two days and next the day’s predicted maximum and minimum temperature data have been incorporated into the input data. Table 5.2 summarises the total input variables that are needed for training and testing. Methods with different numbers of hidden nodes and different learning
procedures; namely the delta and cumulative delta rule, have been investigated. Two types of learning algorithms the Back-propagation and combination of Self Organisation Map (SOM) and Back-propagation, have also been adopted for each of the proposed neural network configurations.

![Diagram of a neural network configuration with input layers, hidden layer, and output layer.](image)

**Figure 5.3:** Fully connected neural network configuration with temperature data

<table>
<thead>
<tr>
<th>Input No.</th>
<th>Load</th>
<th>Input No.</th>
<th>Indices the day of week</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Load ((d, t-24))</td>
<td><strong>M</strong></td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Load ((d, t-23))</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Load ((d, t-22))</td>
<td>31</td>
<td>0</td>
</tr>
<tr>
<td>.</td>
<td>&quot; &quot; &quot; &quot; &quot;</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>Load ((d,t-2))</td>
<td>34</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>Load ((d,t-1))</td>
<td>35</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>Ave. Temp((d-2))</td>
<td>36</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>Ave. Temp((d-1))</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>Ave. Temp((d))</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>Max Temp((d+1))</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>Min Temp((d+1))</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 5.2:** The input data structure for 24-hour ahead load forecasting
5.2.2.1 MLP with Back-propagation Learning

In these networks, a normal multi layer perceptron structure associated with a back-propagation learning algorithm is used. These networks consist of three layers namely, input, hidden and output layer. The input layer consists of 24-hour loads for the current day, indices for weekdays, and temperature data; including average daily temperatures for the past two days, and the forecasted maximum and minimum temperature of the "forecast" day. The hidden layer nodes consist of general and special nodes. The special nodes are associated to week and temperature nodes. The indices of weekdays have been connected to week nodes together with general nodes. The output layer consists of 24 nodes, each represents the next 24-hour load. Different scenarios in terms of the type of connection, number of hidden nodes, and learning procedure have been implemented and tested.

Model A

In these networks, the input layer contains thirty six nodes, twenty four nodes corresponding the 24 hourly load's of the current day, seven indices for the day of the week, three daily average temperatures of past two days and current day, and the predicted maximum and minimum temperatures for next day. The hidden layer consists of a number of general nodes that are connected to all the input nodes, and one supporting node that is connected to the indices of the week input data only. These nodes are referred to as week nodes. Different numbers of general nodes has been chosen and tested. Figure 5.4 shows the schematic representation of this configuration.
Model B:

These networks are similar in the number of inputs and outputs to the previous networks (model A), except that there are extra connections between the temperature nodes and the week-day indices' nodes to the output nodes. These direct connections make the temperature and weekday indices variables distinct from the other variables. Here, temperature and weekday indices have greater influence on the forecasted output. This is needed to push the network towards convergence. Figure 5.5 shows a schematic representation of this scheme.

Figure 5.4: 24-hour ahead load forecasting, Model A
5.2.2.2 MLP Network with Combined Back Propagation and Self Organisation Map learning algorithm

A Combined Back Propagation (BP) and Self Organisation Map (SOM) learning algorithm can be used to implement a combination clustering (grouping) and forecasting simultaneously. The SOM uses an unsupervised learning technique to cluster various types of data patterns into similar classes. In this configuration, SOM has been used to identify the type of the day. This replaces the use of seven numbers as indices of the day of a week. The outputs of SOM are fed into a BP network as input data. Figure 5.6 shows schematically the inter-relation between back-propagation and self organisation map mechanisms in delivering the final 24-hour ahead load forecast.

---

1 Self Organisation Map is an unsupervised network that normally has been used for classification. The main role of this network, in power system load forecasting, is to classify the load data into several groups in terms of their pattern similarity.
5.3 Assessment Methods

Different error measurements have been utilised to evaluate the accuracy of the forecast results. Although there is no widely-accepted criterion for load forecasting tasks, the following criteria can be identified and used to assess the forecast results obtained from different load forecasting techniques [58]:

1. the average root mean square (RMS) of the daily peak load,
2. the absolute relative error of the daily peak load,
3. the mean absolute relative error (MARE),
4. the mean standard deviation,
5. the mean relative error,
6. the mean absolute error, and
7. the absolute average error.
The mean absolute relative error (MARE) has been chosen by many researchers to evaluate the performance of a forecasting algorithm. This performance measurement is used as an error criterion to compare and choose the most appropriate network structure and learning algorithm for short-term load forecasting. The mean absolute relative error is defined as the mean of the absolute value of the difference between the forecasted and the actual loads divided by the actual load for each hour. The formula for this measurement is expressed as:

\[ E = \frac{\text{actual load} - \text{forecast load}}{\text{actual load}} \times 100 \]  

In addition, the forecast results obtained from the most appropriate, conventional methods have been included in the neural network forecast result to further support the superiority of neural network approaches compared to conventional short-term load forecasting methods.

### 5.4 Conventional Forecasting Methods

Among the widely used conventional forecasting methods two carefully selected techniques are implemented and the forecast results based on the available input data set are illustrated for comparison purposes. These techniques are General Exponential Smoothing and Integrated Auto Regressive Moving Average. Similar to the neural network-based forecasting methods, these two conventional techniques are capable of forecasting the next hourly load demand using the hourly load series data with and without temperature data.

#### 5.4.1 General Exponential Smoothing

As mentioned in section (3.10.2), the accuracy of forecasting using the general exponential smoothing technique heavily relies on two factors, \( \beta \) and \( w \). Values of \( \beta = 0.99 \) and \( w = 168 \) were found to be the best choice to achieve a minimum mean absolute relative error (MARE) for the given input data set.
5.4.2 Integrated Auto Regressive Moving Average

The Integrated Auto Regressive Moving Average (ARIMA) model has been developed using two-month hourly load data. The ARIMA model has been identified after removing the seasonal trend and performing several transformations using the auto correlation function and the partial auto correlation function. The coefficients of the model have been estimated using the Gaussian Likelihood Least Square method. The seasonal trend has been taken from the load series using differencing in lag 1, 12, and 168 (hour). Table 5.3 shows the estimated coefficients for the general model derived in Section (3.6.2):

\[ \Phi(B) \nabla^d y(t) = C + \Theta(B) u(t) \]

where

\[ \nabla^d = (1 - B)^d \]

\[ \Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p \]

\[ \Theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q \]

| \(\Phi\) | 0.36 | -0.08 | -0.15 | 0.05 | -0.10 | 0.03 | 0.03 | 0.06 | -0.10 | 0.18 | -0.16 | -0.06 |
| 0.06 | -0.13 | -0.01 | 0.01 |
| \(\Theta\) | -0.06 | -0.04 | -0.01 | 0.00 | 0.02 | 0.04 | -0.01 | -0.07 | 0.07 | -0.07 | -0.02 | 0.04 |
| 0.01 | -0.04 | 0.04 | -0.02 | -0.07 | 0.05 | -0.02 | 0.01 | -0.01 | 0.02 | -0.05 | -0.75 |

Table 5.3: The ARIMA coefficients

5.5 Forecast Results: Neural Network Methods

The neural network approach to forecasting was carried with the aid of commercial software called NeuralWare®. For one-hour ahead load forecasting, the models explained in Sections 5.2.1.1 and 5.2.1.2 were implemented using the available training and testing data set. For 24-hour ahead load forecasting, the models described in section 5.2.2 and 5.4 were implemented. The historical input data for a winter peak load was collected. The network trained with two-third of the input data and tested with the remaining one-third.
5.5.1 Operating Mode I: One-Hour ahead Forecaster

5.5.1.1 Partially Connected

Different models based on the number of hidden nodes using a partially connected network configuration were implemented and tested. A two-week hourly load and temperature data set were used for training and a one-week data set was used for testing.

5.5.1.1.1 Temperature Included

The most recent hourly temperatures and the previous two day’s average temperatures have been used in the training and testing data set. Different number of nodes in the hidden layer and two types of training, cascade and normal have been implemented.

**Cascade Learning:**

Using a cascade learning algorithm, the network with 12 hidden nodes showed a minimum mean absolute relative error. The resulting MARE and its standard deviation are 1.54% and 1.42% respectively. The forecast and actual load for one-week hourly based forecasting are shown in Figure 5. 7. As it can be seen the load pattern has been followed accurately by the neural network output. The only errors of consequence happen in the rapid rise or fall of load demand in the vicinity of any localised peak. This could be related to non-appropriate temperature data. In spite of that, a good forecast performance has achieved as the major error source is related to the minimum load.
**Normal Learning:**

The same input data set and network configuration were used in association with a normal learning algorithm. All input and hidden nodes participated in the learning. A network with 10 hidden nodes showed a minimum mean absolute relative error. The forecast results for the next week after the training data set are shown in Figure 5.8. The MARE and its standard deviation are 1.59% and 1.50% respectively. The result for this network is a little worse than the previous network (cascade learning). Again, the large error occurs at those points where the load curve has changed suddenly.

Similar to the network with cascade learning, it can be seen that the load pattern has been followed by the neural network output accurately. Substantial errors only occur when there is rapid rise or fall of the load demand in the vicinity of a localised peak.
Chapter 5: Forecasting Results and Comparison

5.5.1.1.2 Temperature Excluded

The forecast results for the partially connected network without any temperature data in the input layer were obtained for different number of nodes in the hidden layer of the corresponding networks. The network with four general nodes and one week-node showed a minimum error over the testing data set. The result is shown in Figure 5.9 having a MARE and a standard deviation of 1.48% and 1.46% respectively. The average error for this network is less than for the two previous networks with the temperature data. However, the error in the peak points of load curve is slightly more than the previous two networks. This highlights the effect of temperature on the peak load consumption that has not been incorporated in this network. The problem associated with peak forecasts has not been fully resolved. This is related to the lack of reliable temperature data.

Figure 5.8: Partially connected NN with temperature data input.
5.5.1.1.3 Chen et al Forecast result

A partially connected network based on the exact network proposed by Chen et al has been constructed and implemented. The result of this implementation is shown in Figure 5.10. The MARE and its standard deviation for this model are 3.97% and 3.34% respectively. It was expected that this forecast would contain large error as it employed a normal back propagation learning algorithm.

The proposed neural network scheme (with and without temperature) offers much better forecast results as shown in Figure 5.7 and Figure 5.8. This is due to the adoption of a learning algorithm which includes the variable learning rate and the activation function.

![Figure 5.9: Partially connected NN forecast result without temperature](image-url)
Chapter 5: Forecasting Results and Comparison

Figure 5.10: Partially connected NN forecast results proposed by Chen et al

5.5.1.2 Fully Connected Configuration

The same input data set for training and testing of the partially connected models also were used for a fully connected configuration network. The networks with and without temperature were implemented and tested.

5.5.1.2.1 Temperature included

The networks with temperature nodes in the input layer and a fully connected configuration with different numbers of hidden nodes were implemented. Among all the networks tested, the one with eight hidden nodes showed the minimum error. The MARE and its standard deviation for this model are 1.61% and 1.44% respectively. Figure 5.11 shows the forecast result for one-week hourly forecasting. The forecast results obtained from this network are slightly higher than the corresponding partially connected network. No improvement has been obtained using a fully connection configuration.
5.5.1.2.2 Temperature excluded

In the absence of temperature data, five networks with different numbers of hidden nodes were implemented. Among all the networks tested, the one with five hidden nodes showed a minimum error. The MARE and its standard deviation for this model are 1.51% and 1.44% respectively. Figure 5.12 shows the one-week forecasting result. The overall average error for this network is less than its fully connected network with temperature data in input layer. This can be justified because the temperature data for this case only covers part of the total load. However, its peak forecasting error is much larger than the previous forecasting network. The reason for this large error is related to the effect of the temperature on the peak load consumption.

![Graph showing one-week forecasting result](image)

Figure 5.11: Fully connected NN forecast with temperature data input
5.5.2 Operating Mode II: 24-hour ahead Forecast

Several networks proposed in section 5.2.2 were implemented using the same input data set and special learning procedure, similar to those introduced for one-hour ahead load forecasting networks. For these networks, different activation functions and three varieties of back propagation learning algorithm; namely, delta rule, cumulative delta rule, and normalised cumulative delta rule, were examined. Examination of these learning algorithm outcomes suggest the same procedure as for the one-hour ahead models can be used in this case. Hence, the tangent hyperbolic and normalised cumulative delta rule have been chosen as the activation function and learning algorithm respectively. The combination of BP and SOM (BP/SOM) with different corresponding learning procedures have also been investigated.
5.5.2.1 MLP with Back-propagation Learning

Two variations of the proposed neural network model shown in Figure 5.4 and Figure 5.5 have been implemented and tested. The input data used for training and testing, the learning algorithm, and the amount of training are identical for both models, labelled Model A and Model B.

**Model A:**

The network shown in Figure 5.4 was trained using two month’s hourly load and temperature data. Different possible configurations, in terms of the number of nodes in the hidden layer, were examined. The resulting networks were tested with the following month’s daily load data. The network that gave the “best” forecast contains four general nodes in its hidden layer. The corresponding forecast result is shown in Figure 5.13. The MARE and its standard deviation are 3.92% and 3.18% respectively. The network properly followed the load pattern but relatively large errors occur around the peaks of the load curve.

![Figure 5.13: The forecast result for 24-hour ahead load forecasting Mode II-A](image)
5.5.2.1.1 Model B

The same procedure as for Model A was used for this model. The MARE and its standard deviation are 2.80% and 2.56% respectively. The forecast result is shown in Figure 5.14. The peak forecasting error in this model is seen to be considerably less than the previous model. This is due to the effect of the predicted maximum and minimum temperature that were directly connected to the output as well as connected to the nodes in hidden layer.

5.5.2.2 MLP Network with Combined Back Propagation and Self Organisation Map learning algorithm

The MLP network connection with a combined back propagation and self organisation map learning algorithm makes use of the benefit of the data set grouping of the SOM algorithm and the function estimation of the BP algorithm. Several varieties of this connection were tested using the same data as that used in the normal 24-hour ahead BP networks. The forecast result for the SOM/BP learning algorithm with 5x5 hidden nodes in the SOM network and four hidden nodes in the BP network showed the best forecast result in terms of the accuracy compared with the other network configurations in this family (ie. BP/SOM). The MARE and its standard deviation for the network using the available load data are 3.95% and 3.41% respectively. The main source of error in this model, as can be seen from Figure 5.15, relates to the valleys of the load profile. This can be associated with improper classification of the type of the day. Noting that, for a reliable classification, a sufficient amount of historical load data is needed. For this model, a two-month data set was used for training. The result showed that, the neural network could not learn the classification properly, and hence large errors resulted due to the discrimination between weekday and weekend.
Chapter 5: Forecasting Results and Comparison

Figure 5.14: The forecast result for 24-hour ahead load forecasting Mode II-B

Figure 5.15: The forecast result for 24-hour ahead load forecasting using BP/SOM learning algorithm
Chapter 5: Forecasting Results and Comparison

5.6 Tabulated Presentation of Forecast Results

The forecasting result obtained from implementing the proposed networks have been summarised and presented in tabular form for both operating modes (Mode I and Mode II). The MARE and standard deviation of all networks implemented for one-hour ahead forecast shown in Table 5.4.

### Mean Absolute relative Error

<table>
<thead>
<tr>
<th></th>
<th>Hidden Nodes(*)</th>
<th>Hidden Nodes(*)</th>
<th>Hidden Nodes(*)</th>
<th>Hidden Nodes(*)</th>
<th>Hidden Nodes(*)</th>
<th>Connection Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Temperature</td>
<td>1.48 (5)</td>
<td>1.73 (7)</td>
<td>1.55 (9)</td>
<td>1.52 (11)</td>
<td>1.61 (13)</td>
<td>Partially Connected</td>
</tr>
<tr>
<td>With Temperature</td>
<td>2.26 (8)</td>
<td>1.86 (10)</td>
<td>2.05 (12)</td>
<td>1.67 (12)</td>
<td>1.59 (14)</td>
<td>Partially Connected</td>
</tr>
<tr>
<td>Cascade &amp; with Temperature</td>
<td>1.61 (8)</td>
<td>1.54 (10)</td>
<td>1.54 (12)</td>
<td>1.88 (12)</td>
<td>1.91 (14)</td>
<td>Partially Connected</td>
</tr>
</tbody>
</table>

### Standard Deviation of Error

<table>
<thead>
<tr>
<th></th>
<th>Hidden Nodes(*)</th>
<th>Hidden Nodes(*)</th>
<th>Hidden Nodes(*)</th>
<th>Hidden Nodes(*)</th>
<th>Hidden Nodes(*)</th>
<th>Connection Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Temperature</td>
<td>1.46 (5)</td>
<td>1.62 (7)</td>
<td>1.47 (9)</td>
<td>1.46 (11)</td>
<td>1.57 (13)</td>
<td>Partially Connected</td>
</tr>
<tr>
<td>With Temperature</td>
<td>1.70 (8)</td>
<td>1.88 (10)</td>
<td>1.74 (12)</td>
<td>1.52 (12)</td>
<td>1.50 (14)</td>
<td>Partially Connected</td>
</tr>
<tr>
<td>Cascade &amp; with Temperature</td>
<td>1.51 (8)</td>
<td>1.42 (10)</td>
<td>1.42 (12)</td>
<td>1.50 (12)</td>
<td>1.72 (14)</td>
<td>Partially Connected</td>
</tr>
</tbody>
</table>

### Table 5.4:

Error and Standard deviation for the forecast result obtained from one-hour ahead operating mode.

---

2 As an example, the marked cells in Table 5.4 gives the MARE and standard deviation respectively for the network with a partial connection with temperature input and cascade learning procedure and 12 hidden nodes.
Summarised information for the networks used for 24-hour ahead forecasting is shown in Table 5.5.

<table>
<thead>
<tr>
<th>Model A and B</th>
<th>Hidden Nodes G+W</th>
<th>Connection with output layer</th>
<th>Mean absolute relative error (%)</th>
<th>Standard deviation error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal BP</td>
<td>5</td>
<td>No connection</td>
<td>4.5</td>
<td>2.91</td>
</tr>
<tr>
<td>Model A</td>
<td>5+1</td>
<td>&quot;</td>
<td>4.23</td>
<td>3.12</td>
</tr>
<tr>
<td>Model A</td>
<td>4+1</td>
<td>&quot;</td>
<td>3.92</td>
<td>3.18</td>
</tr>
<tr>
<td>Model A</td>
<td>5+1</td>
<td>&quot;</td>
<td>4.69</td>
<td>3.52</td>
</tr>
<tr>
<td>Model A</td>
<td>6+1</td>
<td>&quot;</td>
<td>3.94</td>
<td>3.03</td>
</tr>
<tr>
<td>Model B</td>
<td>5+1</td>
<td>Tmax &amp; Tmin</td>
<td>3.90</td>
<td>3.56</td>
</tr>
<tr>
<td>Model B</td>
<td>6+1</td>
<td>Tmax &amp; Tmin</td>
<td>3.96</td>
<td>3.21</td>
</tr>
<tr>
<td>Model B</td>
<td>7+2</td>
<td>Tmax &amp; Tmin</td>
<td>4.64</td>
<td>3.48</td>
</tr>
<tr>
<td>Model B</td>
<td>7+1</td>
<td>&quot;</td>
<td>3.85</td>
<td>3.28</td>
</tr>
<tr>
<td>Model B</td>
<td>7+1</td>
<td>All Temperature</td>
<td>3.32</td>
<td>2.79</td>
</tr>
<tr>
<td>Model B</td>
<td>7+2</td>
<td>Temperature and Week</td>
<td>3.93</td>
<td>3.35</td>
</tr>
<tr>
<td>Model B</td>
<td>7+2</td>
<td>&quot;</td>
<td>2.81</td>
<td>2.56</td>
</tr>
<tr>
<td>Model B</td>
<td>7+2</td>
<td>&quot;</td>
<td>3.21</td>
<td>2.699</td>
</tr>
</tbody>
</table>

G=General
W=week

<table>
<thead>
<tr>
<th>BP_SOM</th>
<th>SOM Nodes</th>
<th>BP hidden nodes</th>
<th>Connection to output layer</th>
<th>Mean absolute relative error (%)</th>
<th>Standard deviation error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>5x5</td>
<td>1</td>
<td>No connection</td>
<td>7.58</td>
<td>4.55</td>
</tr>
<tr>
<td>14</td>
<td>5x5</td>
<td>3</td>
<td>&quot;</td>
<td>7.27</td>
<td>4.94</td>
</tr>
<tr>
<td>15</td>
<td>4x4</td>
<td>3</td>
<td>All inputs</td>
<td>6.97</td>
<td>4.41</td>
</tr>
<tr>
<td>17</td>
<td>5x5</td>
<td>4</td>
<td>No connection</td>
<td>7.94</td>
<td>4.25</td>
</tr>
<tr>
<td>18</td>
<td>5x5</td>
<td>4</td>
<td>All input</td>
<td>3.94</td>
<td>3.41</td>
</tr>
</tbody>
</table>

Table 5.5: Summarised forecast information for 24-hour ahead load forecasting networks
5.7 Conventional Methods: Forecast Result and Comparison

A 24-hour ahead conventional load forecasting model has been selected and its forecast result is shown along with a neural network based forecast result in Figure 5.16. The selected conventional method is an ARIMA model that gives the most accurate forecast result for available input data, compared to other conventional methods. The neural network forecast result in this comparison resulted from the network shown in Figure 5.5 in conjunction with a locally modified back propagation learning algorithm. To ensure an impartial comparison, the same input data used for training the neural network was employed for the ARIMA model identification. A visual comparison between the forecast results shows the superiority of the NN-based model performance over the best candidate among the conventional methods. Moreover, the error calculation carried out for this comparison supports the superiority of the neural network performance. Other neural network based load forecasting features such as ease of implementation, flexibility and above all absence of any kind of load modeling will further encourage the use of neural network for power systems short-term load forecasting.

![Figure 5.16: ARIMA Forecasting result and Mode II_A comparison](image-url)
5.8 Sensitivity Analysis

5.8.1 Bad Data

Sensitivity of a neural network to a noisy input data is crucial in short-term load forecasting. By their very nature, historical load and temperature data may contain noisy and missing data. Another source of noisy or bad data is uncertainty of some of the predicted parameters in the input data. All these may cause increase forecast error. For example, the performance of predicted temperature or other weather variables have a significant effect on the performance of the forecasting results. This problem is crucial when the training data is contaminated with bad or noisy data. In this section, the effect of bad data in a testing data set is investigated. Here, the assumption of noise-free training data is important. In order to make a noisy data set, the level of added noise to the actual data is considered as a constant percentage of its value.

Noisy data is created by dithering the input data from their actual values by ±5% of input range values set in the normalising, explained in Section (4.5.2). Based on the above assumption, the network is trained with a training data set without any bad data. However, a little noise and/or missing data is acceptable as the network can extract a general rule based on the randomly distributed input data set. Five networks with minimum MARE were selected. Three one-hour and two 24-hour ahead load forecasting networks were subjected to this testing. Table 5.6 shows the effect of various noisy and bad input data sets on the output for one-hour ahead mode. Table 5.7 and Table 5.8 show the effect of noisy input data on the output for 24-hour ahead operating mode (Model A and BP/SOM model).

As was expected, the network output is very sensitive to some of the input data. For instance, with reference Table 5.6, the change in the current hourly load by 5% makes a significant change in forecast result. The sensitivity of the load in partially connected networks using temperature data is less than in the other two networks.
Table 5.6: The output changes as a result of a 5% change in the input data for one-hour ahead forecasting networks
Table 5.7: The output changes as 5% changes in input data for 24-hour ahead forecasting networks (Model A)
## Table 5.8: The output changes as a result of a 5% changes in the input data for 24-hour ahead forecasting networks (Mode II-BP-SOM)
5.8.2 Network Configuration Design

Another aspect of the sensitivity analysis is related to finding a proper input data set and choosing a proper network configuration. The sensitivity of the output to an input data can be utilised to find the most influential input data in achieving the overall output. For instance, the sensitivity analysis of a forecast load to the humidity input may suggest whether to include or exclude it. This procedure is very useful when the number of input data is very large and consequently the network becomes trapped in local minima. This procedure has been used in our work to eliminate the surplus input data and substitute it with other inputs. In Table 5.6, for example, 5% changes in the past two day average temperature node, in partially connected network, make only -0.2% change in output node. Therefore, this node can be eliminated from the input nodes, and as a result the network can converge faster. Other aspects of the sensitivity analysis can be related to the effect of other network parameters such as type of connection, number of layers and activation function.

5.9 Summary

In this chapter, two general models, one-hour and 24-hour ahead forecasting network, have been introduced. Different network configurations and different learning algorithms have been proposed and tested. The forecast result for each network has been shown along with a relevant brief discussion. For given input data, the best candidate among the conventional forecasting methods was selected. Its corresponding forecast result was compared with the result obtained from a proposed neural network connection in conjunction with a modified learning algorithm. The sensitivity of a neural network to changes in input data has been analysed. The results for this sensitivity analysis have also been given for selected networks used for both one-hour and 24-hour ahead load forecasting.
Chapter 6

Conclusion

6.1 Conclusion

Load forecasting is an area of great economic saving to electric power utilities. A reliable short-term load forecasting tool would enable power utilities to plan for peak demands and to achieve more economical unit allocation, scheduling and pre-dispatching. During the last three decades, many forecasting methods have been developed and applied for various sites using different formats for the input data sets and different pre-processing functions. In broad terms, the forecasting performance and the overall modeling and implementation costs associated with each method are considered to be the main concerns. In this regard, the neural network solution to load forecasting in general and short-term load forecasting in particular is most likely to be a preferred option. Foreseeable further advances in digital technology, providing more accessible and more economical parallel computers, and also availability of more historical data, especially temperature, are ensuring that a neural network approach to load forecasting will be widely accepted in the near future by power utilities world-wide.

This thesis aimed to study all available short-term load forecasting methods in an attempt to suggest a solution (algorithm/structure) which gives the most appropriate forecast output for a typical input data set containing historical load data with or without weather variables input data. In this study, matters such as; forecast accuracy, speed, development/implementation costs, and historical data validation were observed very closely.
Close examination of the load profile is necessary to find the input variables that significantly change the course of energy consumption in a forecast lead time. Implementation of some statistical/mathematical analysis routines on the historical data is also necessary for load model identification or system training. Based on a detailed description of the relevant introductory issues which have been presented in this thesis, short-term load forecasting has been classified into two major classes; conventional methods and neural network techniques. Here, a brief summary of each class is presented, while particular emphasis is given to the neural network based forecasting techniques.

6.1.1 Conventional Forecasting Methods

Conventional forecasting methods such as time series, have been applied to power system short-term load forecasting during the last few decades. Two established time series methods namely; “auto regressive moving average” and “general exponential smoothing” have been thoroughly studied in the course of this research work. This study included the basic theory and load model development leading to a general discussion about the procedures which are needed to find the desired model parameters. It is found that the auto correlation function (ACF) and partial auto correlation function (PAFC) still are well suited for load model identification and its parameter estimation, although the uncertainty in the estimated model parameters is yet to be resolved.

In brief, the following deficiencies have been found to be closely associated with conventional methods of load forecasting;

1. major difficulty in model development (finding the relationship between load and the variables influencing the load consumption),
2. uncertainty in the estimated parameters,
3. site dependent features of the developed models that make them difficult to be adapted for another site.

These deficiencies in fact motivated the investigation of some alternative forecasting mechanisms.
6.1.2 Neural Network Based Forecasting Methods

A more detailed general discussion about neural network and its capabilities to perform short-term load forecasting is presented in this thesis. In the light of the deficiencies mentioned for conventional methods, it was found that the neural network solution overcame the prominent problem of load modelling and its parameters estimation. Other advantages associated with the neural network approach, including ease of development and implementation, flexibility and speed, further support the superiority of the neural network solution over conventional counterparts. The historical load and temperature data sets have been collected and proper data pre-processing functions have been implemented to make them suitable for the neural network training process. Two types of operation modes, one-hour and 24-hour ahead forecasting, have been chosen for model (that is network structure and learning algorithm) development. Different configurations for the proposed neural network have been investigated. Sample forecasting result for each network configuration is shown in terms of different learning algorithms.

A preliminary investigation has resulted in the use of the following system components for the proposed neural network scheme:

1. Tangent Hyperbolic used as the activation function,
2. normalised cumulative back propagation employed as the learning algorithm,
3. random presentation used to introduce the input data set to the network, and
4. variable learning rates used to speed up the convergence of the learning algorithm during the network training.
Based on an extensive effort and results' interpretations, the following choices were found to be proper for the required neural network configuration:

1. Multi layer perceptron (MLP) configuration with the back propagation learning algorithm for one-hour ahead load forecasting networks,

2. fully and partially connected configuration, for one-hour ahead forecasting network with and without temperature data, and

3. usual MLP with back propagation and combined back propagation with Self Organisation Map learning algorithm for 24-hour ahead forecasting network.

Regarding the input data validation and preprocessing, an auto correlation function and a partial auto correlation function were employed to tailor the historical data such as that used for the network training and testing purposes. Further investigation on the load shapes revealed that each day should be separately considered during training of one-hour and 24-hour ahead forecasting networks.

The proposed networks for two operational modes were implemented using the available input data. The mean absolute relative error criterion (MARE) is used to evaluate the forecasting performance. The results indicated that the one-hour ahead forecasting network with partially connected and using temperature data has a proper accuracy. And, the other proposed one-hour ahead forecasting networks have shown relatively less accuracy.

For 24-hour ahead forecasting the standard “Multi Layer Perceptron” was found to be appropriate. For this network connection, two learning algorithms were suggested. These are MLP with proposed learning algorithm and combined MLP with normal BP and SOM learning algorithm.
The sensitivity of a few selected networks those that have shown the minimum error, to bad and noisy input data have been investigated. As was expected, this sensitivity analysis has shown that the neural network output is sensitive to the bad or noisy data especially to those data that have been found correlated to the output using ACF and PACF functions. The sensitivity analysis also showed that the partially connected network with temperature input data is moderately less sensitive to the input data than the other two selected networks. For the 24-hour ahead forecasting network, because of the large number of input and output nodes, the results showed different sensitivity values corresponding to each input data. This suggested the idea of removing a few selected input variables to speed up the convergence.

6.2 Comparison

A 24-hour ahead conventional load forecasting model was selected and its forecast result is shown along with a neural network based forecast result. The selected conventional method is an ARIMA model that gives the most accurate forecast result for the available input data, compared to other conventional methods. To ensure an impartial comparison, the input data used for training the neural network was employed for the ARIMA model identification. A visual comparison between the forecast results showed the superiority of the NN-based model performance over the best of the conventional methods. Moreover, the error calculation carried out for this comparison supports the superiority of the neural network performance. Other neural network based load forecasting features such as ease of implementation, flexibility and above all absence of any kind of load modeling will further encourage the use of neural networks for power systems' short-term load forecasting.
6.3 Further Study

Further investigation is required to determine the reliability the ANN based load forecasting. Because of the large scale load data that is normally associated with power system application, the ANN approach must consider an appropriate data handling scheme. This can facilitate incorporating more variables to the neural network input layer nodes. Among other relevant issues, the following questions are considered crucial and should be dealt with accordingly:

1. How large should an ANN be and which configuration can be used?
2. How much data is required for training and what type of data should be contributed?
3. What criteria should be chosen to evaluate the performance of the neural network approach?
4. What procedure can be selected to update a neural network?
5. How can a neural network be adopted for an on-line operation?

One possible solution to some of these problems may be found in using a hybrid method such as combined neural network and Fuzzy expert system approach. This hybrid system uses the robust identification ability of a neural network, reasoning of an expert system, and capabilities of Fuzzy logic.
References


Appendix A

Back Propagation

To enable a neural network successfully operation an initial training (learning) process would be required. The most popular algorithm for training a neural network is a back propagation learning algorithm (BP). This algorithm is developed to overcome the limitation of the original perceptron [54, 62]. Simple linearly separable problems can be classified by Perceptron, and more complex, non-linear problems can be solved by Multi Layer Perceptrons (MLP), shown in Figure A.1. In a MLP network adjusting the weights between nodes in layers is needed to minimise the outputs errors. Back-propagation learning algorithm is used for this purpose. It solves this problem by assuming that all processing elements (PEs) and connections are somewhat to responsible for an erroneous output. Therefore, the output error backwards through the connections to the previous layers. This process is repeated until the input layer is reached [46].

A neural network using BP learning can be understood on several levels. On one level, it is a collection of vector equations, on another, a computer program, and finally, a layered system of interacting nodes. The BP is relatively easy to be used. As it slows the convergence, its application is limited to the problems that have relatively stable underlying relationships. The back propagation main strengths are related to its ability to store many more patterns than the number of input dimensions and to extract arbitrary complex non-linear mapping. The back propagation main limitations are related to its extremely long training time, its
off-line coding requirement, and the inability to know how to precisely generate any arbitrarily mapping procedure. It should be noted that there is no guaranteed best solution for a back propagation. In other words, it is quite possible for a back-propagation network to learn the training data perfectly and still fails on the testing data.

![Multi Layer Network](image)

**Figure A.1** Multi Layer Network

### A.1.1 Learning Procedure

During a BP training, a network passes each input pattern through the hidden layer to generate a result at each output node. Next, the network passes the derivatives of output squared errors back to the hidden layer using the original weighted connections. A continuous differentiable non-linear activation function is needed for this method. Normally, the sigmoid logistic function is used as activation function in most cases. Each hidden node, then, calculates the weighted sum of the back propagated errors to find its indirect contribution to the known output errors. After each output and hidden node finds its error values the node adjusts its weights to reduce its error. Here, the weight strengths are changed in proportion to the error.
times the input signals which diminishes the error in the direction of the gradient
descent [62]. This implies that neural network learns from off-line training pattern
inputs and desired output for each input pattern. Then, it makes a generalisation for
further input patterns.

In this method the squared error between the calculated output of network and the
desired output is back propagated to the previous layer(s) to minimise error through
adjusting the weights. Before training is begun, the interconnection weights are
assigned to random values; Gaussian or Uniform. It is very important to make sure
that the input data presentation to the network is well randomised. Well ordered or
structured presentation of the training set often leads to failure convergence.

A.1.1.1 Algorithm

Assuming $t_{pj}$ is the desired output for $j^{th}$ component of output pattern for pattern
$p$ and $O_{pj}$ is the $j^{th}$ component of the calculated output pattern produced by
network, the measured error for one pattern can be expressed as:

$$E_p = \frac{1}{2} \sum_j (t_{pj} - O_{pj})^2$$

Apdx 1

And, the output of the network is specified as:

$$O_{pj} = \frac{1}{1 + e^{-net_{pj}}} = f_i (net_p)$$

Apdx 2

$$net_{pj} = \sum_j w_{jk} O_{pk}$$

Apdx 3
Appendix A: Back Propagation

Where, $f_i$ is an activation function and $w_{ji}$ is the weight that should be adjusted.
The following gradient descent of $E_p$ regarding $w_{ji}$ is used:

$$-\frac{\partial E_p}{\partial w_{ji}} = \delta_{pj} O_{pi}$$

Apdx 4

Where, $\delta_{pj}$ is defined in two ways; first, for the output PE, it is given by:

$$\delta_{pj} = (t_{pj} - O_{pj}) f'_j (net_{pj})$$

Apdx 5

and for a PE in an arbitrary hidden layer, it is given by:

$$\delta_{pj} = f'_j (net_{pj}) \sum_k \delta_{pk} w_{kj}$$

Apdx 6

Where, $f'_j$ is the derivative of $f_j$.

The weights are adjusted by:

$$w_{ji}(n+1) = w_{ji} + \eta \delta_{pj} O_{pi}$$

Apdx 7

where, $\eta$ is a gain term (learning rate). The learning rate governs the rate at which weights which are allowed to change at any given presentation. Higher learning rates speed up the convergence process, but can result in overshooting or non-convergence. Slower learning rates produce more reliable results at the expense of increased training time. To decrease the training time of a BP
algorithm, Rumhart have added a portion of the last gradient (weight change), entitled the momentum, to the current weight to keep the weights moving across flat portions of the error surface. This can be written as:

\[ w_{ji}(n+1) = w_{ji} + \Delta w_{ji} \]

Apdx 8

\[ \Delta w_{ji} = \eta \delta_{jp} O_{pi} + \alpha(w_{ji} - w_{ji}(n-1)) \]

Apdx 9

The overall training process is summarised by the flowchart shown in Figure A.2.

---

Figure A.2: Back propagation training flowchart
Once a network settles on a minimum, whether local or global, learning halted. If the error is still unacceptably high, either a change in the number of hidden nodes or in the learning parameters will often fix it, or network can be restarted by different initial values for weights.

A.1.2 Back Propagation Convergence

The convergence is always guaranteed under certain network and training set condition to any desired error [69, 20]. However, using a BP, does not guarantee to find the global minima during training, only the local error minimum can be obtained. This situation often leads to severe oscillations in the weight change during training in that point. Several attempt should be done for solution. These include; dynamic learning rate, other derivation of delta rule, optimal number of hidden units and finally other topologies such as Recurrent network.

A.1.2.1 Learning Rate

Learning rate usually lies somewhere between zero and one. As the network grows large, these values should be set lower, but typically in the same ratio. Also, as the learning process continues, decreasing the value will cause the network to convergence better. The learning rates can be changed dynamically as the learning process proceeds [27].

It was found that different learning rates for different layers in the network help better convergence. In these simulations, the learning rates for those layers close to the output were set to be lower than those layers near to input. This is especially important for applications where the data does not derive from a strong underlying model.
Appendix A: Back Propagation

A.1.2.2 Cumulative Delta Rule

Cumulative Delta Rule is one of the variant of general Delta rule. It was introduced in an attempt to alleviate the problem of structured presentation of the training set. It accumulates the weight change over several training presentations and make the application all at once. Normally, the cumulative delta rule is much less sensitive to the order of the training set [46].

A.1.2.3 Hidden Units

The optimal number of hidden layers is another issue that has been tackled by some researchers. Some heuristic methods for pruning away hidden layer nodes have been proposed during training [62]. Also a dynamic node creation that introduces grows hidden layer nodes as they are needed, can be used to find the optimal number of hidden nodes [64].

A.1.2.4 Recurrent Network

An alternative connection topology that uses the local feedback in nodes has been proposed and applied [64, 13]. This topology introduces feedback into the recall dynamics and suitably called recurrent back propagation network.