Deanship of Graduate Studies
Al-Quds University

Electric Power Load Short Term Forecasting

Rae'd Yousef Mohamed Basbous

M.Sc. Thesis

Jerusalem-Palestine

1430(Hijri) **/ 2009**

Electric Power Load Short Term Forecasting

Prepared By:
Rae'd Yousef Mohamed Basbous

B.Sc.: Computer Systems Engineering, 2000,
Polytechnic University, Palestine

Supervisor: Dr. Labib Arafeh

A thesis submitted in Partial fulfillment of requirements for the degree of Master of Electronics and Computer Engineering/ Department of Electronics and Computer Engineering/ Faculty of Engineering/ Graduate Studies - Al-Quds University

1430 (Hijri)/ 2009

Al-Quds University
Deanship of Graduate Studies
Electronics and Computer Engineering Department

Thesis Approval

Electric Power Load Short Term Forecasting

Prepared By: Rae'd Yousef Mohammed Basbous
Registration No.: 20510239

Supervisor: Dr. Labib Arafeh

Master thesis submitted and accepted, Date: 24/03/2009
The names and signatures of the examining committee members are as follows:

1. Head of Committee: Dr. Labib Arafeh Signature:...............
2. Internal Examiner : Dr. Salah Odeh Signature:...............
3. External Examiner : Dr. Radwan Tahboub Signature:...............

Jerusalem – Palestine

1430 / 2009

Dedication

To My Country "Palestine", My Parents, My Wife and My Lovely Daughters Talah and Daniah.

Declaration:

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: ....

Rae'd Yousef Mohamed Basbous

Date: 24/03/2009

Acknowledgements

First of all, I would like to thank my advisor Assoc. Prof. Dr. Labib Arafeh, who has always guided me into the right directions by his extensive knowledge and experience on the subject; and encouraged, motivated and trusted me while doing this research.

I am thankful to my former director Dr. Mahmoud Omar and the present one Dr. Marwan Darwish for their continuous help and support that they have given to me always with a great politeness and smiling faces.

I owe too much to my wife Shereen for her endless support, understanding, patience, and also for her great help and encouragement.

My appreciations are also to the president of AlQuds Open University, Prof. Younes Amro for his encouragement to complete this thesis. I got the financial support for my study from AlQuds Open University.

I would like to express my special thanks to the staff of the engineering faculty-department of electronic and computer engineering at AlQuds University who taught me during my Master degree study.

I am grateful to Eng. Mansour Nassar and Eng. Salah Alqam from the strategic planning department in Jerusalem District Electricity Company (JDECO) for opening up the load consumption database for this scientific research and also to Mr. Isam Al-Seifi from Palestinian Meteorology Organization (PMO) for providing me the temperature measurements.

I wish to thank Mr. Nader Neirokh, Mr. Alaa Abu Rumaileh, and Dr. Nahidah Al-Kiswani for their help on English and Arabic language structure.

The last but not the least, I would like to thank my parents, my wife and my lovely daughters (Talah and Daniah) who instilled me the free thinking and the joy of making researches.

Jerusalem, March 2009
Rae'd Basbous

Abstract

Short-Term Load Forecasting (STLF) is an important part of the power generation process. For years, it has been achieved by traditional approaches including time series, multiple linear regression, general exponential smoothing, etc; but, new methods based on artificial intelligence emerged recently in literature and started to replace the old ones in the industry. In order to follow the latest developments and to have a modern system, it is aimed to make a research on STLF in Palestine, by Fuzzy Inference System (FIS). For this purpose, new modeling-based methods are explored and proposed to forecast Bier Nabala (as a sample) power electric load one day and one week in advance.

At this study, two kinds of models have been developed, namely the Single Input Single Output, SISO, and Multiple Input Single Output, MISO. These two developed approaches depend on the Fuzzy based techniques including integrated and adaptive Neuro-Fuzzy approaches, and have been compared to represent the STLF models. Real historical data profiles for two years (2006 and 2007) have been used to develop and test these proposed models. These data profiles were provided by Jerusalem District Electric Company (JDECO), and the Palestinian Meteorology Office (PMO). While the provided historical data profile includes the time and the corresponding power load at that time, the weather historical data profile includes humidity, highest temperature, lowest temperature, and wind speed for each day.

A pre-processing stage has been accomplished for the collected data. The bad data and outliers in the collected data identified and removed using an algorithm (*Remove_Outliers*). The Correlation Coefficient (CC) between the datasets before and after applying the *Remove_Outliers* algorithm is 0.998. This indicates that removing the bad data did not affect the data behavior or details. Furthermore, a formatting stage to the time variable to be in real number with fractions has been done.

In order to develop and test the models, we have divided the data using a cross validation algorithm to training and testing datasets (75% of the available historical data profiles have been used for training and 25% for testing). The adequacy of the developed models has been checked using the CC to measure the agreements between the actual and predicted power loads. In addition, two error measures were used namely, Mean Absolute Performance Error (MAPE), and the Root Mean Square Error (RMSE) to indicate the accuracy and the performance of the developed models.

In Developing the SISO models the time parameter was considered as an input, while in MISO models three inputs have been considered namely, Time, High and low Temperatures. For the two kinds of models, the power load was the output.

Different models for SISO and MISO have been developed using the training data, such as, Sugeno FIS with different optimization techniques including Hybrid and Back-propagation optimization techniques, Sugeno model using the Subtractive Clustering, and finally Sugeno cascaded model using Subtractive Clustering and Hybrid optimization technique.

The forecasting performance has been furtherlly improved by the MISO cascaded models, while maintaining all other factors including MFs types and numbers, and cluster radius. This improvement is noticed as an improvement in the obtained CC that results from the MISO cascaded models which ranges between (0.95 and 0.97) and with an average CC value of 0.96 for all the developed models as compared with the forecasting performance of CC that ranges between (0.90 and 0.94) when developing the models using the other optimization techniques; The corresponding MAPE ranges between (0.03 and 0.06) with an average value 0.04, and RMSE ranges between (0.07 and 0.20) with an average value 0.16 compared to average MAPE ranges between (0.05 and 0.07), and average RMSE ranges between (0.18 and 0.27) for the other optimization techniques.

The developed models have been integrated with a stand alone application with Graphical User Interface, GUI. The developed Electric Power Load Forecasting System, EPLFS, can be accessed online to predict the power load.

These preliminary and promising results indicate the suitability and adequacy of the developed models depending on the Fuzzy approach to solve the short term load forecasting using the time and weather variables. Further investigation and works still needed to be applied to furtherlly investigate models with more data, and enhance the developed models by taking other affecting factors on the power load into account such as humidity, wind speed, and a noticeable change on population. In addition, the EPLFS can be upgraded to preprocess the new datasets, fine-tuning the developed models and accordingly predict the power loads online.

Keywords: Artificial Intelligence, Sugeno, Hybrid Optimization, Back-propagation Optimization, STLF, Cross Validation, Outliers.

Table of Contents

CHAPTER FOUR:

CHAPTER FIVE:

CHAPTER SIX:

List of Tables

List of Figures

List of Appendices

Chapter One:

Introduction

1.1 Introduction

Forecasting or predicting is the process of the estimation in unknown situations. In more recent years, forecasting has evolved into the practice of Demand Planning in every day business forecasting for manufacturing companies. Forecasting has applications in many situations, such as, weather forecasting and Meteorology, transport planning, and transportation forecasting, earthquake prediction, water demand, load demand, stock market prediction, and product forecasting.

Load Forecasts (LF) are extremely important for energy suppliers, financial institutions, and other participants in electric energy generation, transmission, distribution, and markets. Load forecasting is an important component of power system operation and planning involving prognosis of the future level of demand to serve as the basis for supply-side and demand side (Khan et al., 2001). Precise load forecasting helps the electric utility to make unit commitment decisions, reduce spinning reserve capacity and schedule device maintenance plan properly (McSharry, 2006).

LF can be divided into three main categories according to (Ho et al., 1990), (Desouky et al., 2000) and (Satish et al, 2004). These categories are Long-Term Load Forecasting, LTLF, Mid-Term Load Forecasting, MTLF, and Short-Term Load Forecasting, STLF.

In spite of the numerous literatures on LF published since 1960s, the research work in this area is still a challenge to the electrical engineering scholars because of its high complexity. How to estimate the future load with the historical data has remained a difficulty up to now, especially for the load forecasting of holidays, days with extreme weather and other anomalous days. With the recent development of new artificial intelligence tools, it is potentially possible to improve the forecasting result (Hwang et al., 2001), (Desoukey et al., 2000) and (Kiartzis et al., 2000).

In this thesis, different modeling techniques for the short term load forecasting problem have been explored. Different measures have been used to check the adequacy of the developed models. These models have been integrated with a stand alone application with GUI. The developed Electric Power Load Forecasting System "EPLFS" is as shown in Fig (1.1); the figure demonstrates the forecasted power loads for a testing datasets. The three lists in the system present the times, forecasted loads, and the actual loads. Three different measures appear in the bottom of the right corner, the Correlation Coefficient (CC), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE).

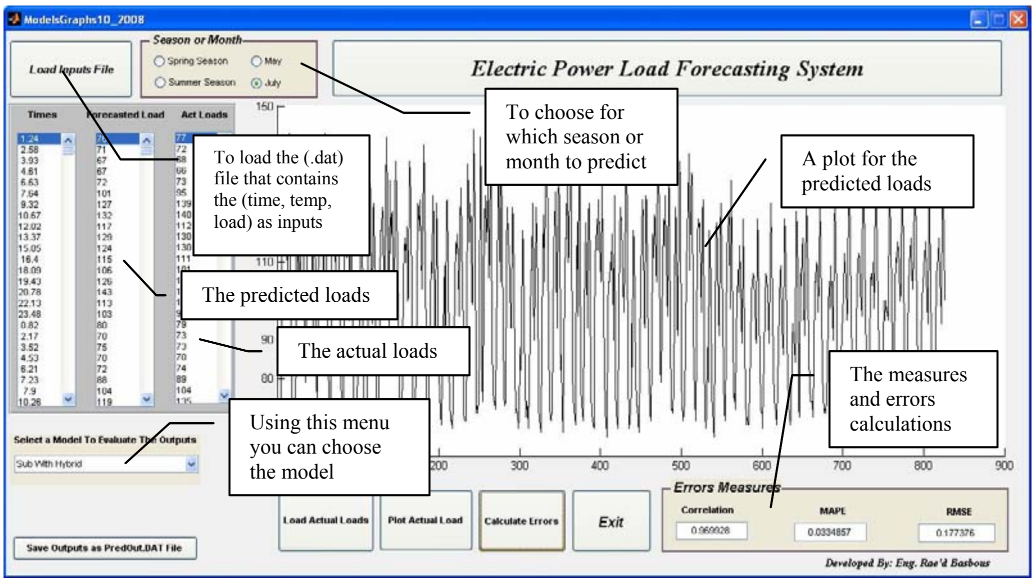


Fig. (1.1): The EPLFS system: Showing a plot for the predicted loads in a certain hours.

Fig 1.2 below illustrates a general developing "training" block diagram of our models. It consists of three main stages. The first stage is pre-processing the input data sets for the system. These datasets include four elements; three of them are inputs; namely, the time, the high temperature, and the low temperature of day and one output (the actual loads). The second stage is concerned with various soft computing models that were developed. The third stage checks the adequacy of the developed models.

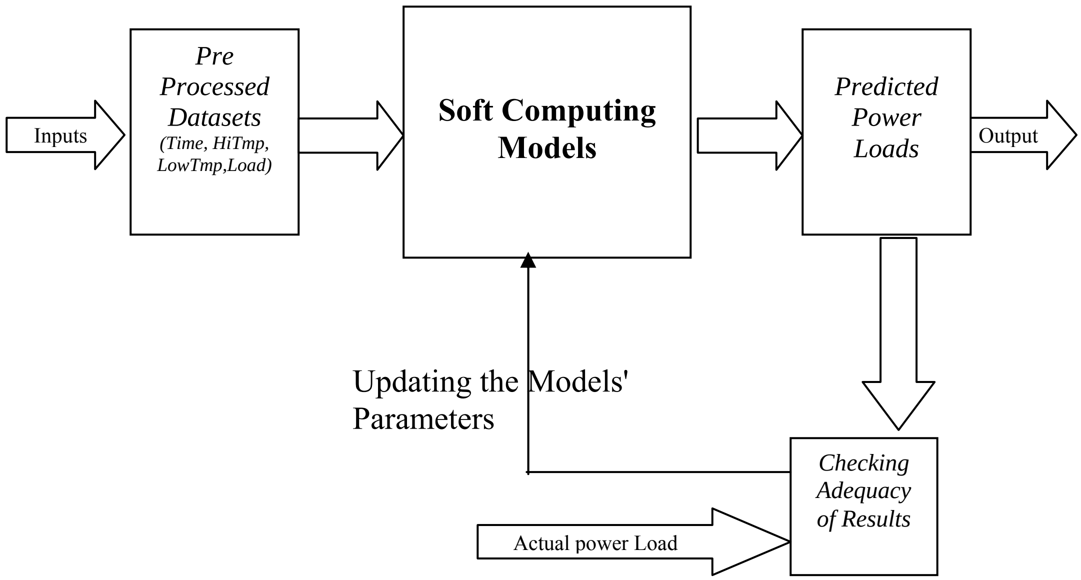


Fig. (1.2): A General Block Diagram for Developing/Training Soft Computing Models

Two kinds of models depending on the Fuzzy based techniques including integrated and adaptive Neuro-Fuzzy approaches have been developed, Single Input Single Output (SISO) models, and Multiple Inputs Single Output (MISO) models. For the SISO models the time has been used as the input for the models and the power load at that time has been used as the output. In the MISO models, three variables (time, High Temperature, and the Low Temperature for that day) have been used as an input for the developed models, and the power load at that time was considered as the model output. Fig (1.3), and Fig (1.4) below show a SISO and MISO model.

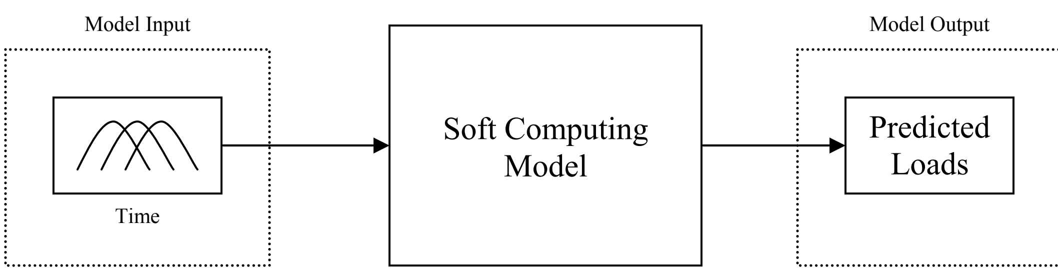


Fig. (1.3): SISO Model Architecture.

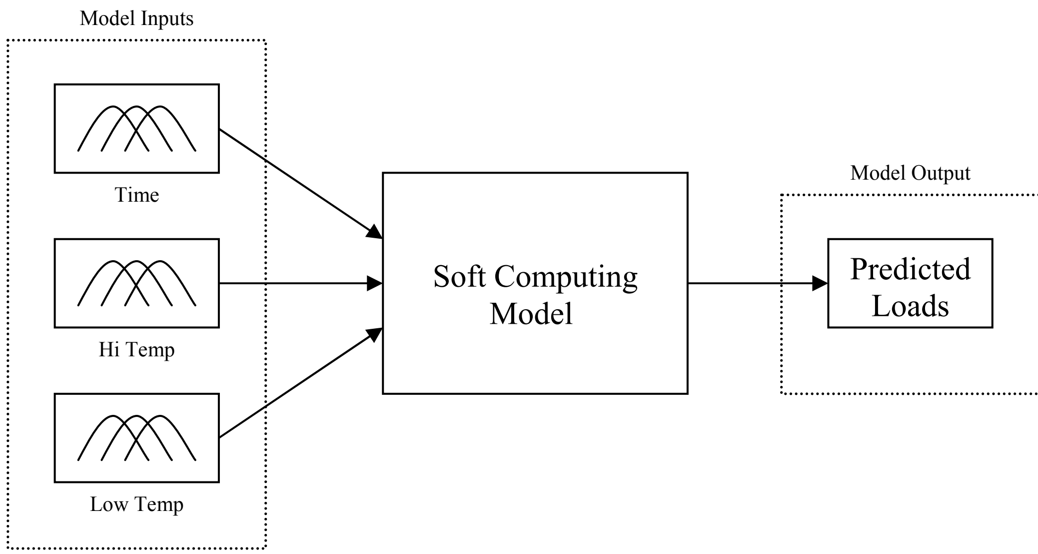


Fig. (1.4): MISO Model Architecture.

1.2 Problem Definition

Power Load forecasting is playing a key role in the power systems. Load forecasting helps an electric utility to make important decisions including decisions on purchasing and generating electric power, load switching, and infrastructure development. Various factors influence the system load behavior, such as weather, time, economy, and random disturbance.

At this study, several models have been explored and developed to represent the STLF problem, by applying integrated and adaptive Neuro-Fuzzy approaches. Real historical data profiles have been used to train/develop and test the new models. These data profiles were provided by Jerusalem District Electric Company (JDECO), and the Palestinian Meteorology Office (PMO). These data profiles include; the time, weather elements, and the power loads.

1.3 Research Objectives

The main aim of this research is to explore the use of different soft computing modeling techniques such as Fuzzy Logic, Neural Networks, as well as different hybrid techniques in the STLF problem. The various tactical objectives include:

1. Collecting a set of actual history power load and weather elements data set to train and test the models.

2. Pre-processing the system inputs before the training stage, to detect and remove the outlier data using existing algorithms to improve the performance of the developed models.
3. Studying the weather factors that affect the electric power load, and their contributions by building and comparing SISO and MISO models.
4. Developing a Sugeno Fuzzy Inference System (FIS) model using the Adaptive Neuro Fuzzy Inference System (ANFIS) with different optimization techniques to forecast the daily electric power load.
5. Developing a fuzzy inference model based on Subtractive Clustering technique to implement a classification and system identification technique.
6. Developing a model by cascading two techniques (Subtractive Clustering and Hybrid optimization) to achieve a more accurate model.
7. Checking the adequacy of the developed models using different metrics.
8. Designing a stand alone application with a graphical user interface (GUI) to load the datasets, evaluate the predicted output using the developed models, plot the actual and predicted load, and calculate several measures including the CC, MAPE and RMSE.

1.4 Rationale

The need to solve highly nonlinear, time variant problems has been growing rapidly as many of today's applications have nonlinear and uncertain behavior which changes with time. Currently no model based method exists that can effectively address complex, nonlinear and time variant problems in a general way. These problems coupled with others (such as problems in decision making, prediction, etc.) have inspired a growing interest in intelligent soft computing techniques including Fuzzy Logic, Neural Networks, Expert Systems, etc. Intelligent Systems, in general, use various combinations of these techniques to address real world complex problems.

The research approaches of short-term load forecasting can be mainly divided into two categories: statistical methods and artificial intelligence ones. In statistical methods, equations can be obtained showing the relationship between load and its relative factors after training the historical data, whereas artificial intelligence methods try to imitate human beings' way of thinking and reasoning to get knowledge from the past experience and forecast the future load.

The statistical category includes multiple linear regression (Papalexopoulos et al., 1990), stochastic time series (Amjady, 2001), general exponential smoothing (Christianse, 1971), state space (Villalba et al., 2000), etc.

Expert systems (Hwan, 2001), Artificial Neural Networks (ANN) (Desouky et al., 2000) and fuzzy inference systems (Kim et al., 2000) belong to the artificial intelligence category. Expert systems try to get the knowledge of experienced operators and express it in an "if…then" rule, but the difficulty is sometimes the experts' knowledge is intuitive and could not easily be expressed. ANN doesn't need the expression of the human experience and aims to establish a network between the input data set and the observed outputs. It is good at dealing with the nonlinear relationship between the load and its relative factors, but the shortcoming lies in over fitting and long training time (Jang et al., 1997).

Fuzzy inference is an extension of expert systems. It constructs an optimal structure of the simplified fuzzy inference that minimizes model errors and the number of the membership functions to grasp nonlinear behavior of short-term loads, yet it still needs the experts' experience to generate the fuzzy rules. Generally artificial intelligence methods are flexible in finding the relationship between load and its relative factors.

Thus we have aimed to explore the use of the adaptive neuro fuzzy inference system models to address the short term load forecasting.

1.5 Contributions

In our research, several models have been developed based on FIS using several optimization techniques such as Hybrid, Back-propagation, Subtractive Clustering, and finally improve the models outcome by cascading the Subtractive Clustering with the Hybrid optimization technique to solve the short term load forecasting for a chosen power line in Beir Nabala village at Jerusalem district. Furthermore, two kinds of these models (SISO and MISO) have been developed to study the effect of the weather parameters on the power load. In addition to that similar works depend on artificial intelligence in the field of short term load forecasting have been mentioned and discussed in our literature survey.

To develop the proposed models, a pre-processing stage has been accomplished for the collected data. The bad data and outliers in the collected data identified and removed using an algorithm (*Remove_Outliers*, Appendix A). Furthermore, a formatting stage to the time variable to be in real number with fractions has been done.

Also, the developed models have been integrated in a stand alone application with a GUI, which can be accessed online through a local area network, or using a web server. Fig.(1.1) above shows a snapshot of the current version of the EPLFS system. Load forecasting can be done using this system, so one can load datasets saved in a text file ".DAT", obtain the forecasted load using the developed models, plot the predicted power loads and the actual ones if known, and evaluate the predicted output by calculating the various measures used including the CC, MAPE and RMSE.

The average CC between the predicted and actual loads for all the models found to be between the values (90.94% and 97.24%). The performance of the EPLFS to predict the power loads for one day and one week ahead has been tested using new unseen datasets from the year 2008. The average CC found equal to 94.12% in case of one day ahead prediction and equal to 93.27% in case of one week ahead prediction.

1.6 Research Conventions

In this research the following conventions have been adopted unless otherwise stated.

- The number of sample load points of per day is 72, i.e. the sampling interval is 20 minutes. The used data sets are for an 8 months from two years 2006 and 2007 (four summer months for July and August and four spring months for April and May), and cross validation algorithm was used to divide them to training and testing data sets.

- The source of data is Jerusalem District Electric Company "JDECO", and the Palestinian Meteorology Office (PMO), for a specific power line in Beir Nabala village.

- The examples and figures are from Birnabala village main power line. The same data have also been employed for the generalization of the models.

- MAPE, RMSE, and CC will be employed to measure the error of the developed models.

- Matlab Version 7.6 (R2008a) has been used to develop the models, and to design our stand alone GUI system.

1.7 Research Structure

The following chapters of this research can be mainly divided into 6 parts; namely, introduction to the load forecasting, introduction to fuzzy inference systems, historical data used, developing the models, results and discussions, and finally the conclusions and future works. The research is organized as follows:

Chapter two gives an introduction to the forecasting and the short-term load forecasting problem. The property of the system load, various forecasting approaches, and the difficulty in forecasting are introduced. Chapter three provides a brief overview of the fuzzy inference system, neuro fuzzy, adaptive neuro fuzzy inference system, and subtractive clustering. In chapter four, the historical data that used, and the pre-processing and manipulation of the data is discussed. Chapter five, presents the developed models that adopted to short-term load forecasting (single input with single output, and multiple inputs with single output), these models include Sugeno models with different optimization techniques, Subtractive clustering, and cascading two techniques. In addition, the error measures that used to compare between the actual and predicted data was discussed in the same chapter. In Chapter six, the results (output) of the developed models that were introduced will be presented and discussed. Chapter seven concludes and highlights the research results obtained, and suggests future work for the research work.

Chapter Two:

Basic Concepts of Load Forecasting

2.1 Introduction

Forecasting is important in many aspects of our lives. As individuals, we try to predict success in our marriages, occupations, and investments. Organizations invest enormous amounts based on forecasts for new products, factories, retail outlets, and contracts with executives. Government agencies need forecasts of the economy, environmental impacts, new sports stadiums, and effects of proposed social programs (Armstrong, 2001).

Forecasting is often confused with planning. Planning concerns what the world should look like, while forecasting is about what it will look like. Planners can use forecasting methods to predict the outcomes for alternative plans. If the forecasted outcomes are not satisfactory, they can revise the plans, and then obtain new forecasts, repeating the process until the forecasted outcomes are satisfactory. Forecasting serves many needs. It can help people and organizations to plan for the future and to make rational decisions. It can help in deliberations about policy variables (Armstrong, 2001).

Forecasting has application in many situations, such as:

- ***Weather forecasting and Meteorology:*** Today, weather forecasts are made by collecting quantitative data about the current state of the atmosphere and using scientific understanding of atmospheric processes to project how the atmosphere will evolve. Weather forecasts become less accurate as the difference in time between the present moment and the time for which the forecast is being made increases (Encarta, 2008).

- ***Transport planning and Transportation forecasting:*** is the process of estimating the number of vehicles or travelers that will use a specific transportation facility in the future. A forecast estimates, for instance, the number of vehicles on a planned freeway or bridge, the ridership on a railway line, the number of passengers patronizing an airport, or the number of ships calling on a seaport. Traffic forecasting begins with the collection of data on current traffic. Together with data on population, employment, trip rates, travel costs, etc., traffic data are used to develop a traffic demand model. Feeding data on future population, employment, etc. into the model results in output for future traffic, typically estimated for each segment of the transportation infrastructure in question, e.g., each roadway segment or each railway station (White, 1979) and (Kholer, 1995).

- ***Product forecasting:*** is the science of predicting the degree of success a new product will enjoy in the marketplace. To do this, the forecasting model must take into account such things as product awareness, distribution, price, fulfilling unmeet needs and competitive alternatives (Thomas, 1993).

- ***Water Demand Forecasting:*** Water demand forecasts are used for several purposes, such as: planning new developments or system expansion; to estimate the size and operation of reservoirs, pumping stations and pipe capacities; and for urban water management issues (e.g. pricing policy, water use restrictions, etc.) (Bougadis et al., 2005).

- ***Economic Forecasts:*** Economic forecasts are a key building block of any budgetary projection. In concert with tax codes and expenditure plans, they determine budgetary goals. As a consequence, any ex-post assessment of fiscal policy will have to consider the discrepancy between projected and actual economic growth (Jonung et al., 2006).

2.2 The Control of Electric Power Production

The total amount of electric power consumed in an electrical power system must be balanced with an equal amount of generated power. There is no efficient way of storing large amounts of electrical energy. To maintain this power balance between production and consumption the power input to the power system must be controlled. By using various methods to forecast future power needs, the electric power production may be controlled. This is called power load adoption (Felix, 2002).

The electric power production is controlled using a long term production program, based on forecasts of future power load. The production programs have to make sure that the energy production is carried out at low cost (production optimization). In addition, there are requirements on electric power availability and also on electric power quality.

2.3 Load Forecasting

Accurate models for electric power load forecasting are essential to the operation and planning of a utility company. Load forecasting helps an electric utility to make important decisions including decisions on purchasing and generating electric power, load switching, and infrastructure development. Load forecasts are extremely important for energy suppliers, financial institutions, and other participants in electric energy generation, transmission, distribution, and markets.

Load forecasts can be divided into three main categories according to (Ho et al., 1990), (Desouky et al., 2000) and (Satish et al, 2004):

1. Long-term load forecasting (LTLF), covering from five to 20 years. LTLF is used by planning engineers and economists to determine the type and size of generating plants that minimize both fixed and variable costs.

2. Mid-term load forecasting (MTLF), ranging from one month to five years. MTLF is used by utilities to purchase enough fuel and from which the changes in electricity tariffs are calculated. The MTLF and LTLF take into account the historical load and weather data, the number of customers in different categories, the appliances in the area and their characteristics including age, the economic and demographic data and their forecasts, the appliance sales data, and other factors.

3. Short-term load forecasting (STLF), over an interval ranging from an hour to a week. For STLF several factors should be considered, such as time factors, weather data, and possible customers' classes. STLF is important for different functions such as unit commitment, economic dispatch, energy transfer scheduling and real time control.

The forecasts for different time horizons are important for different operations within a utility company. The natures of these forecasts are different as well. It is also possible, according to the industry practice, to predict the so-called weather normalized load, which would take place for average annual peak weather conditions or worse than average peak weather conditions for a given area. Weather normalized load is the load calculated for the so-called normal weather conditions which are the average of the weather characteristics for the peak historical loads over a certain period of time. The duration of this period varies from one utility to another.

Most forecasting methods use statistical techniques or artificial intelligence algorithms such as neural networks, fuzzy logic, and expert systems. Two of the methods, so-called end-use and econometric approach are broadly used for medium- and long-term forecasting. A variety of methods, which include the so-called similar day approach, various regression models, time series, neural networks, statistical learning algorithms, fuzzy logic, and expert systems, have been developed for short-term forecasting.

As it can be noticed, a large variety of mathematical methods have been used for load forecasting. The development and improvements of appropriate mathematical tools will lead to the development of more accurate load forecasting techniques. The accuracy of load forecasting depends not only on the load forecasting techniques, but also on the accuracy of forecasted weather scenarios.

2.4 Important Factors for Load Forecasts

The system load is the sum of all the individual demands at all the nodes of the power system (Gross et al., 1987). The objective of load forecasting system is to forecast the future system load. Good understanding of the system characteristics helps to design reasonable forecasting models and select appropriate models in different situations. Various factors influence the system load behavior, which can be mainly classified into the following categories (Gross et al., 1987): Weather, Time, Economy, Random disturbance.

These factors are briefly introduced in the following subsections.

2.4.1 Weather

Weather conditions influence the load. In fact, forecasted weather parameters are the most important factors in short-term load forecasts. The change of the weather causes the change of consumers' comfort feeling and in turn the usage of some appliances such as water heater and air conditioner. Weather-sensitive load also includes appliance of agricultural irrigation due to the need of the cultivated plants. In the areas where summer and winter have great meteorological difference, the load patterns differ greatly.

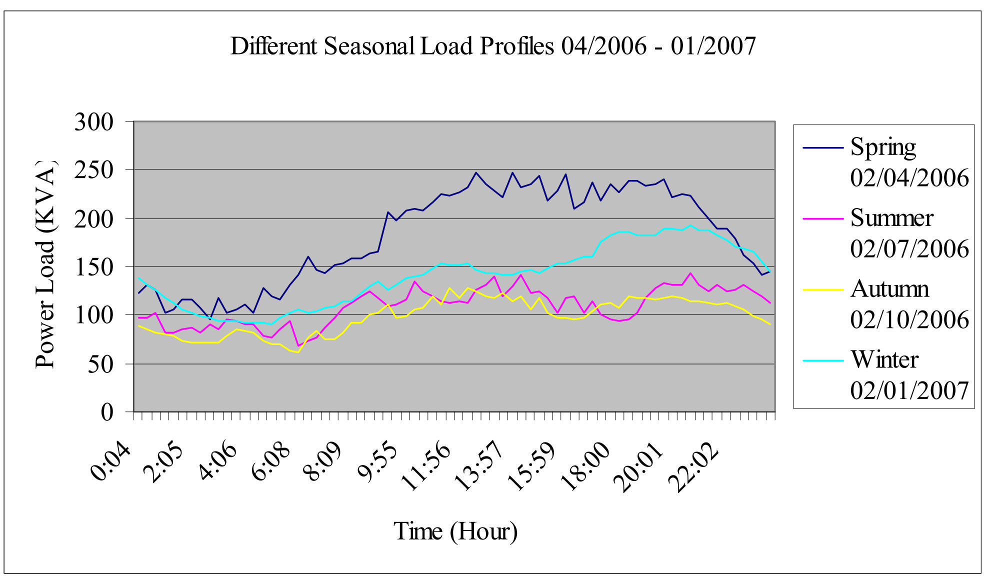


Fig. (2.1): Different Seasonal 24 Hours Load Profiles for Birnabala Village Power Line.

Various weather variables could be considered for load forecasting such as, temperature, humidity, precipitation, wind speed, cloud cover, light intensity and so on. Temperature and humidity are the most commonly used load predictors. An electric load prediction survey published in (Hippert et al., 2001) indicated that of the 22 research reports considered, thirteen made use of temperature only, three made use of temperature and humidity, three utilized additional weather parameters, and three used only load parameters. Fig. (2.1) shows the typical different seasonal 24 hours load profiles for Birnabala Village power line for the year 2006 and 2007 as provided from JDECO, where months 3-5 considered as spring season, 6-8 as summer, 9-11 as autumn, and from 12 to 2 considered as the winter season).

Normally the intraday temperatures are the most important weather variables in terms of their effects on the load; hence they are often selected as the independent variables in STLF. Temperatures of the previous days also affect the load profile. For example, continuous high temperature days might lead to heat buildup and in turn a new system peak. Humidity is also an important factor, because it affects the human being's comfort feeling greatly. People feel hotter in the environment of 35˚C and 70% relative humidity than in the environment of 37˚C and 50% relative humidity. That's why THI (temperature-humidity index) (Rahman, 1990) is sometimes employed as an affecting factor of load forecasting. It is a meaningful topic to select the appropriate weather variables as the inputs of STLF.

Fig. (2.2) shows a simple example to the variations of the high temperatures occurred in the months April and July in the year 2006. The smoothness of the temperatures variations can be noticed in July.

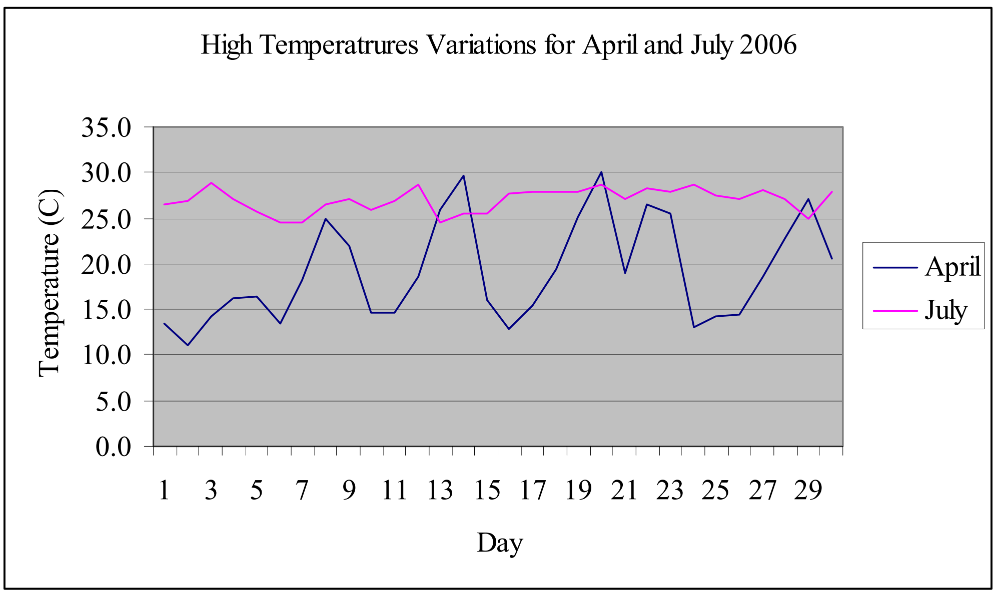


Fig. (2.2): The High Temperatures Variations for the Months April and July.

2.4.2 Time

The time factors influencing the load include the time of the year, the day of the week, and the hour of the day, holiday property, weekday/weekend property and season property. From the observation of the load curves it can be seen that there are certain rules of the load variation with the time point of the day. For example, the typical load curve of the normal winter weekdays (from Thursday to Wednesday) of the Birnabala power line in Jerusalem is shown in Fig.(2.3), with the sample interval of about 25 minutes. The load is low 0:00 to 6:00; it rises from around 6:00 to 12:00 and then becomes stable until around 17:00; then it reaches the peak value between 17:00 and 19:00, after that it descends gradually until 21:00; thereafter it rises a little again until 22:00; it descends again until the end of the day. Actually this load variation with time reflects the arrangement of people's daily life: working time, leisure time and sleeping time.

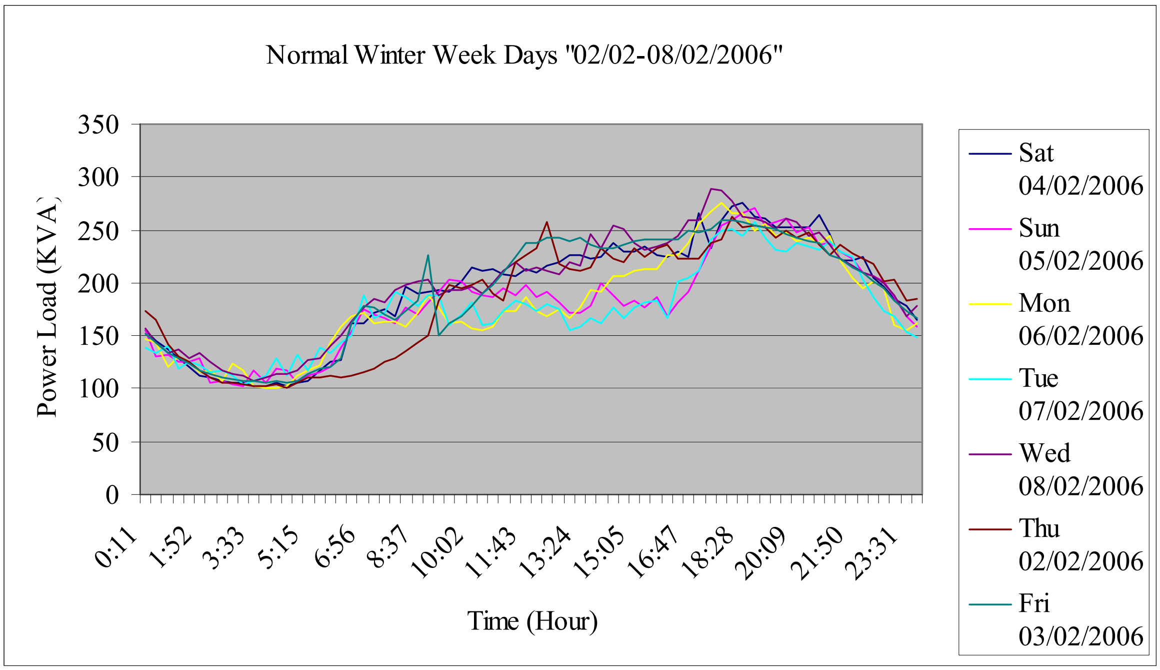


Fig. (2.3): Normal winter weekdays (from Thursday to Wednesday) of the Beir Nabala power line in Jerusalem for the period 02/02/2006-08/02/2006.

There are also some other rules of load variation with time. The weekend or holiday load curve is lower than the weekday curve, due to the decrease of working load. Shifts to and from daylight savings time and start of the school year also contribute to the significant change of the previous load profiles.

Periodicity is another property of the load curve. There is very strong daily, weekly, seasonal and yearly periodicity in the load data. Taking good use of this property can benefit the load forecasting result.

2.4.3 Economy

Electricity is a kind of commodity. The economic situation also influences the utilization of this commodity. Economic factors, such as the degree of industrialization, price of electricity and load management policy have significant impacts on the system load growth/decline trend. With the development of modern electricity markets, the relationship between electricity price and load profile is even stronger. Although time-of-use pricing and demand-side management had arrived before deregulation, the volatility of spot markets and incentives for consumers to adjust loads are potentially of a much greater magnitude. But in JDECO this is not applied, since they have a fix rate for the all period of the day.

2.4.4 Random Disturbance

The modern power system is composed of numerous electricity users. Although it is not possible to predict how each individual user consumes the energy, the amount of the total loads of all the small users shows good statistical rules and in turn, leads to smooth load curves. But the startup and shutdown of the large loads, such as steel mill, synchrotrons and wind tunnels, always lead to an obvious impulse to the load curve. This is a random disturbance, since for the dispatchers, the startup and shutdown time of these users is quite random, i.e. there is no obvious rule of when and how they get power from the grid. When the data from such a load curve are used in load forecasting training, the impulse component of the load adds to the difficulty of load forecasting.

Special events, which are known in advance but whose effect on load is not quite certain, are another source of random disturbance. A typical special event is, for example, a world cup football match, which the dispatchers know for sure will cause increasing usage of television, but cannot best decide the amount of the usage.

2.5 Classification of Developed Load Forecasting Approaches

Over the last few decades a number of forecasting approaches have been developed. Two of the approaches, so-called end-use and econometric approach are broadly used for medium- and long-term forecasting. A variety of approaches, which include the so-called similar day approach, various regression models, time series, neural networks, expert systems, fuzzy logic, and statistical learning algorithms, are used for short-term forecasting. The development, improvements, and investigation of the appropriate mathematical tools will lead to the development of more accurate load forecasting techniques.

Statistical approaches usually require a mathematical model that represents load as function of different factors such as time, weather, and customer class. The two important categories of such mathematical models are: additive models and multiplicative models. They differ in whether the forecast load is the sum (additive) of a number of components or the product (multiplicative) of a number of factors.

2.5.1 Medium and Long-term Load Forecasting Approaches

The end-use modeling, econometric modeling, and their combinations are the most often used approaches for medium- and long-term load forecasting. Descriptions of appliances used by customers, the sizes of the houses, the age of equipment, technology changes, customer behavior, and population dynamics are usually included in the statistical and simulation models based on the so-called end-use approach. In addition, economic factors such as per capita incomes, employment levels, and electricity prices are included in econometric models. These models are often used in combination with the end-use approach (Feinberg et al., 2005).

End-use models. The end-use approach directly estimates energy consumption by using extensive information on end use and end users, such as appliances, the customer use, their age, sizes of houses, and so on. Statistical information about customers along with dynamics of change is the basis for the forecast. End-use models focus on the various uses of electricity in the residential, commercial, and industrial sector. These models are based on the principle that electricity demand is derived from customer's demand for light, cooling, heating, refrigeration, etc. Thus end-use models explain energy demand as a function of the number of appliances in the market (Gellings 1996).

Econometric models. The econometric approach combines economic theory and statistical techniques for forecasting electricity demand. The approach estimates the relationships between energy consumption (dependent variables) and factors influencing consumption (Armstrong, 2001). The relationships are estimated by the least-squares approach method or time series approaches (Feinberg et al., 2005). One of the options in this framework is to aggregate the econometric approach, when consumption in different sectors (residential, commercial, industrial, etc.) is calculated as a function of weather, economic and other variables, and then estimates are assembled using recent historical data. Integration of the econometric approach into the end-use approach introduces behavioral components into the end-use equations.

The end-use and econometric approaches require a large amount of information relevant to appliances, customers, economics, etc. Their application is complicated and requires human participation. In addition such information is often not available regarding particular customers and a utility keeps and supports a profile of an "average" customer or average customers for different type of customers. The problem arises if the utility wants to conduct next-year forecasts for sub-areas, which are often called load pockets (Feinberg et al., 2005). In this case, the amount of the work that should be performed increases proportionally with the number of load pockets. In addition, end-use profiles and econometric data for different load pockets are typically different.

2.5.2 Short-term Load Forecasting Approaches

The research approaches of short-term load forecasting can be mainly divided into two categories: *statistical* approaches and *artificial intelligence approaches* (Yang, 2006). In the statistical approaches, equations can be obtained showing the relationship between load and its relative factors after training the historical data (Yang, 2006), while artificial intelligence approaches try to imitate human beings' way of thinking and reasoning to get knowledge from the past experience and forecast the future load.

The statistical category includes multiple linear regression (Papalexopoulos et al., 1990), general exponential smoothing (Christianse, 1971), state space (Villalba et al., 2000), etc. Recently Support Vector Regression (SVR) (Yang et al., 2004) and (Li, 2002), which is a very promising statistical learning method, has also been applied to short-term load forecasting. Expert system (Hwan, 2001), Artificial Neural Network (ANN) (Desouky et al., 2000) and fuzzy inference approaches (Kim et al., 2000) belong to the artificial intelligence category.

Expert systems try to get the knowledge of experienced operators and express it in an "if…then" rule. Artificial neural network doesn't need the expression of the human experience and aims to establish a network between the input data set and the observed outputs. It is good at dealing with the nonlinear relationship between the load and its relative factors, but the shortcoming lies in over fitting and long training time (Jang et al., 1997). Fuzzy inference is an extension of expert systems. It constructs an optimal structure of the simplified fuzzy inference that minimizes model errors and the number of the membership functions to grasp nonlinear behavior of short-term loads, yet it still needs the experts' experience to generate the fuzzy rules. Generally artificial intelligence approaches are flexible in finding the relationship between load and its relative factors, especially for the anomalous load forecasting.

Hereafter, some of the main STLF approaches are introduced.

2.5.2.1 Artificial Neural Networks (ANN)

The use of ANN has been a widely studied load forecasting technique since 1990 (Peng et al., 1992). ANNs are essentially non-linear circuits that have the demonstrated capability to do non-linear curve fitting.

The outputs of an artificial neural network are some linear or non-linear mathematical function of its inputs. The inputs may be the outputs of other network elements as well as actual network inputs. In practice network elements are arranged in a relatively small number of connected layers of elements between network inputs and outputs. Feedback paths are sometimes used.

In applying a neural network to load forecasting, one must select one of a number of architectures (e.g. Hopfield, back propagation, Boltzmann machine), the number and connectivity of layers and elements, use of bi-directional or uni-directional links and the number format (e.g. binary or continuous) to be used by inputs and outputs (Yang, 1998).

Learning of ANN's may be performed by unsupervised techniques that construct internal models which capture and extract regularities in their input data without receiving any information a priori. The supervised techniques (which require information a priori about the input and desired output data) and the reinforcement ones (which require a single scalar evaluation of the produced output) are also used. The supervised Backpropagation-based ANN functions by accepting the inputs, processing them, producing an output, comparing this output with the desired output, and adjusting the weights to minimize the error and produce better output (Arafeh et al., 1999).

Bakirtzis et al. (Bakirtzis, 1996) developed an ANN based short-term load forecasting model for the Energy Control Center of the Greek Public Power Corporation. In the development they used a fully connected three-layer feed forward ANN and a back propagation algorithm was used for training. Input variables include historical hourly load data, temperature, and the day of week. The model can forecast load profiles from one to seven days. The average error result for one-day ahead forecast of normal days is 0.0224, while it is 0.0356 for the holidays. Also Papalexopoulos et al. (Papalexopoulos et al., 1994) developed and implemented a multi-layered feed forward ANN for short-term system load forecasting. In the model three types of variables are used as inputs to the neural networks: seasonal related inputs, weather related inputs, and historical loads. The average absolute error result is found equal to 0.0195.

Khotanzad (Khotanzad et al., 1997) described a load forecasting system known as ANNSTLF. It is based on multiple ANN strategy that captures various trends in the data. In the development they used a multilayer perceptron trained with an error back propagation algorithm. ANNSTLF can consider the effect of temperature and relative humidity on the load. It also contains forecasters that can generate the hourly temperature and relative humidity forecasts needed by the system. The average MAPE for one day ahead is found 0.0263. An improvement of the above system was described in (Khotanzad et al., 1998). In the new generation, ANNSTLF includes two ANN forecasters: one predicts the base load and the other forecasts the change in load. The final forecast is computed by adaptive combination of these forecasts. The effect of humidity and wind speed are considered through a linear transformation of temperature. The average MAPE in the new generation found to be 0.0205. At the time it was reported in (Khotanzad et al., 1997), ANNSTLF was being used by 35 utilities across the USA and Canada. Chen et al. (Chen et al., 2004) also developed a three layer fully connected feed forward neural network and a back propagation algorithm was used as the training approach. Their ANN though considers electricity price as one of the main characteristics of the system load. The average MAPE for their model when taking the effect of the temperature is 0.0363. Many published studies use artificial neural networks in conjunction with other forecasting techniques such as time series (Chow et al., 1996).

2.5.2.2 Expert Systems

Rule-based forecasting makes use of rules, to do accurate forecasting. Expert systems incorporate rules and procedures used by human experts in the field of interest into software that is then able to automatically make forecasts without human assistance.

Ho et al. (Ho et al., 1990) proposed a knowledge-based expert system for the short-term load forecasting of the Taiwan power system. Operators' knowledge and the hourly observation of system load over the past five years are employed to establish eleven day-types. Weather parameters were also considered. The average mean absolute error obtained is 0.0252. Rahman and Hazim (Rahman et al., 1996) developed a site-independent technique for short-term load forecasting. Knowledge about the load and the factors affecting it is extracted and represented in a parameterized rule base. This rule-based system is complemented by a parameter database that varies from site to site. The technique is tested in different sites in the United States with low forecasting errors. The load model, the rules and the parameters presented in the paper have been designed using no specific knowledge about any particular site. Results improve if operators at a particular site are consulted. The average absolute error found equal to 0.0218.

2.5.2.3 Fuzzy Logic

Fuzzy logic is a generalization of the usual Boolean logic used for digital circuit design. An input under Boolean logic takes on a value of “True” or “False”. Under fuzzy logic an input is associated with certain qualitative ranges. For instance the temperature of a day may be “low”, “medium” or “high”. Fuzzy logic allows one to logically deduce outputs from fuzzy inputs. In this sense fuzzy logic is one of a number of techniques for mapping inputs to outputs. Among the advantages of the use of fuzzy logic is the absence of a need for a mathematical model mapping inputs to outputs and the absence of a need for precise inputs. With such generic conditioning rules, properly designed fuzzy logic systems can be very robust when used for forecasting. Of course in many situations an exact output is needed. After the logical processing of fuzzy inputs, a “defuzzification” can be used to produce such precise outputs. (Kiartzis et al., 2000) and (Miranda et al., 2000) describe applications of fuzzy logic to load forecasting. The average percentage error obtained in (Kiartzis et al., 2000) is 0.0245.

2.5.2.4 Wavelets

Recently, wavelet theory has received wide attention in applications to analysis of transient disturbances and signal compression in power systems. With characteristics similar to ban-filters, the wavelet transform decomposes a signal into scales of signals at different levels of resolution (Huang et al., 2001). Along with timing information, each scale of the signal denotes the different frequency content of the original signal. Also, through the inverse wavelet transform, the original signal can be recovered from the decomposed scales of signals (Huang et al., 2001).

Yu et al. (Yu et al., 2000) proposed a novel wavelet transform–based approach for short-time load forecasting of weather-sensitive loads. In this approach, Daubechies wavelet transforms are adopted to predict short-term loads, and the numerical results reveal that certain wavelet components can effectively be used to identify the load characteristics in electric power systems. The wavelet coefficients associated with certain frequency and time localization are adjusted using the conventional multiple regression method and then reconstructed in order to forecast the final loads through a three-scale synthesis technique. The outcome of the study clearly indicates that the proposed wavelet transform approach can be used as an attractive and effective means for short-term load forecasting. The average mean percentage error obtained is 0.020.

2.6 Limitations for the STLF

Several difficulties exist in the short-term load forecasting. Some of these are briefly introduced in the next subsections.

2.6.1 Precise Hypothesis of the Input-output Relationship

Most of the STLF approaches hypothesize a regression function (or a network structure, e.g. in ANN) to represent the relationship between the input and output variables. How to hypothesize the regression form or the network structure is a major difficulty because it needs detailed a prior knowledge of the problem. If the regression form or the network structure were improperly selected, the prediction result would be unsatisfactory. For example, when a problem itself is a quadratic, the prediction result will be very poor if a linear input-output relationship is supposed.

Another similar problem is parameter selection: not only the form of the regression function (or the network structure), but also the parameters of it should be well selected to get a good prediction. Moreover, it is always difficult to select the input variables. Too many or too few input variables would decrease the accuracy of prediction. It should be decided which variables are influential and which are trivial for a certain situation. Trivial ones that do not affect the load behavior should be abandoned.

Because it is hard to represent the input-output relationship in one function, the mode recognition tool, clustering, has been introduced to STLF (Erkmen et al. 1997). It divides the sample data into several clusters. Each cluster has a unique function or network structure to represent the input and output relationship. This approach tends to have better forecasting results because it reveals the system property more precisely. But a prior knowledge is still required to do the clustering and determine the regression form (or network structure) for every cluster.

2.6.2 Generalization of Experts' Experience

Many experienced working staff in power grids is good at manual load forecasting. They are even always better than the computer forecasting. So it is very natural to use expert systems and fuzzy inference for load forecasting. But transforming the experts' experience to a rule database is a difficult task, since the experts' forecasting is often intuitive.

2.6.3 The Forecasting of Special Days

Loads of special days are also not easy to be predicted precisely, due to the dissimilar load behavior compared with those of ordinary days during the year, as well as the lack of sufficient samples. These days include public holidays, consecutive holidays, days preceding and following the holidays, days with extreme weather or sudden weather change and special event days. Although the sample number can be greatly enhanced by including the days that are far away from the target day, e.g. the past 5 years historical data can be employed rather than only one or two years, the load growth through the years might lead to dissimilarity of two sample days. From the experimental results it is found that days with sudden weather change are extremely hard to forecast. This sort of day has two kinds of properties: the property of the previous neighboring days and the property of the previous similar days. How to combine these two properties is a challenging task.

2.6.4 Inaccurate or Incomplete Forecasted Weather Data

As weather is a key factor that influences the forecasting result, it is employed in many models. Although the technique of weather forecasting, like the load forecasting, has been improved in the past several decades, sometimes it is still not accurate enough. The inaccurate weather report data employed in the STLF would cause large error.

Another problem is, sometimes the detailed forecasted weather data cannot be provided. The normal one day ahead weather report information includes highest temperature, lowest temperature, average humidity, precipitation probability, maximum wind speed of the day, weather condition of two period of the day (morning, and evening). Usually the period of the load forecasting is 24 hours. If the forecasted weather data of these 24 hours can be known in advance, it would greatly increase the precision. However, normal weather reports do not provide such detailed information, especially when the lead time is long.

Chapter Three:

Introduction to Soft Computing

3.1 Introduction

Soft computing is "an emerging approach to computing, which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision" (Zadeh, 1994, 1997). It, in general, is a collection of computing tools and techniques, shared by closely related disciplines that include fuzzy logic, artificial neural nets, genetic algorithms, belief calculus, and some aspects of machine learning like inductive logic programming (Jang et al., 1997). These tools are used independently as well as jointly depending on the type of the domain of applications. It is now realized that complex real-world problems require intelligent systems that combine knowledge, techniques, and methodologies from various sources. These intelligent systems are supposed to possess humanlike expertise within a specific domain, adapt themselves and learn to do better in changing environments, and explain how they make decisions or take actions.

The real-world decision-making is too much complex, uncertain and imprecise to lend itself to precise, prescriptive analysis. It is this realization that underlies the rapidly growing shift from conventional techniques of decision analysis to technologies based on fuzzy logic. It was originally proposed as a means for representing uncertainty and formalizing qualitative concepts that have no precise boundaries. So far, engineering applications of fuzzy logic have gained much more attention than business and finance applications, but an even larger potential exists in the latter fields.

Fuzzy logic (FL) is an excellent means to combine Artificial Intelligence methods. The advantage of fuzziness dealing with imprecision fit ideally into decision systems; the vagueness and uncertainty of human expressions is well modeled in the fuzzy sets, and a pseudo-verbal representation, similar to an expert's formulation, can be achieved. Fuzzy logic avoids the abrupt change from one discrete output state to another when the input is changed only marginally. This is achieved by a quantization of variables into membership functions.

Neural Networks (NN) originated in an attempt to build mathematical models of elementary processing units in the brain and the flow of signals between these processing units. After a period of stagnation, these formal models have become increasingly popular, with the discovery of efficient algorithms capable of fitting them to data sets. Since then, neural nets have been applied to build computerized architectures that can approximate nonlinear functions of several variables, and classify objects. A neural net is nothing more than a sophisticated black box nonlinear model that can be trained on data.

Genetic Algorithms (GA) are stochastic combinatorial approximation procedures that have been inspired by a biological analogy: the mutation and cross-over of chromosomes in genetics. They belong to a large class of meta-heuristics that are used to avoid local minima in heuristic search methods. The particularity of genetic algorithm is to try to improve several solutions to a problem in parallel, mixing random jumps and crossing.

A detailed discussion for these approaches is presented in the following sections.

3.2 Fuzzy Expert Systems (FES)

Expert systems were designed to reason through knowledge to solve problems using methods that humans use. Expert systems use heuristic knowledge rather than numbers to control the process of solving the problem. Expert systems have their knowledge encoded and maintained separately from the computer program, which uses that knowledge to solve the problem. Expert systems are capable of explaining how a particular conclusion was reached, and why requested information is needed.

A Fuzzy Expert System (FES) is an expert system that utilizes fuzzy sets and fuzzy logic to overcome some of the problems, which occur when the data provided by the user are vague or incomplete. The power of fuzzy set theory comes from the ability to describe linguistically a particular phenomenon or process, and then to represent that description with a small number of very flexible rules. In a fuzzy system, the knowledge is contained both in its rules and in fuzzy sets, which hold general description of the properties of the phenomenon under consideration. The structure of the FES is shown in Fig.(3.1).

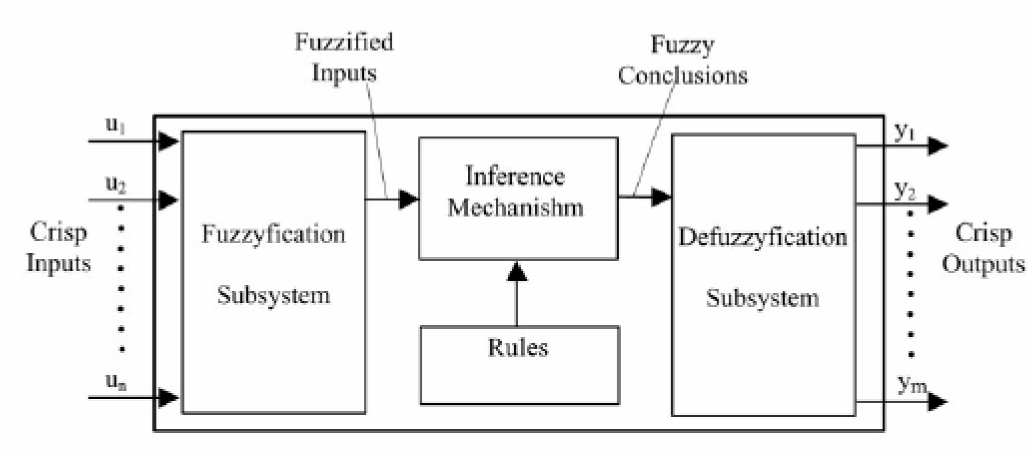


Fig.(3.1): Structure of the Fuzzy Expert System (Llata et al., 2001)

One of the major differences between a FES and another expert system is that the first can infer multiple conclusions. In fact it provides all possible solutions whose truth is above a certain threshold, and the user or the application program can then choose the appropriate solution depending on the particular situation. This fact adds flexibility to the system and makes it more powerful. Fuzzy expert systems use fuzzy data, fuzzy rules, and fuzzy inference, in addition to the standard ones implemented in the ordinary expert systems.

In building FES, the crucial steps are the fuzzification (transforming the crisp inputs into degrees of match with linguistic values) and the construction of blocks of fuzzy rules. These steps can be handled in two different ways. The first is by using information obtained through interviews to the experts of the problem. The second is by using methods of machine-learning, neural networks and genetic algorithms to learn membership functions and fuzzy rules. The two approaches are quite different. The first does not use the past history of the problem, but it relies on the experience of experts who have worked in the field for years; whereas, the second is based only on past data and project into the future the same structure of the past.

3.3 Fuzzy Inference System

Fuzzy Inference Systems (FISs) are also known as fuzzy rule-based systems, fuzzy model, fuzzy expert system, and fuzzy associative memory. This is a major unit of a fuzzy logic system. The decision-making is an important part in the entire system. The FIS formulates suitable rules and based upon the rules the decision is made. This is mainly based on the concepts of the fuzzy set theory, fuzzy IF–THEN rules, and fuzzy reasoning. FIS uses "IF. . . THEN. . . " statements, and the connectors present in the rule statement are "OR" or "AND" to make the necessary decision rules. The basic FIS can take either fuzzy inputs or crisp inputs, but the outputs it produces are almost always fuzzy sets. When the FIS is used as a controller, it is necessary to have a crisp output. Therefore in this case defuzzification method is adopted to best extract a crisp value that best represents a fuzzy set.

3.3.1 Construction and Working of Inference System

Fuzzy inference system consists of a fuzzification interface, a rule base, a database, a decision-making unit, and finally a defuzzification interface. A FIS with five functional block described in Fig. (3.2). The function of each block is as follows:

- a *rule base* containing a number of fuzzy IF–THEN rules;
- a *database* which defines the membership functions of the fuzzy sets used in the fuzzy rules;
- a *decision-making unit* which performs the inference operations on the rules;
- a *fuzzification interface* which transforms the crisp inputs into degrees of match with linguistic values;
- a *defuzzification interface* which transforms the fuzzy results of the inference into a crisp output.

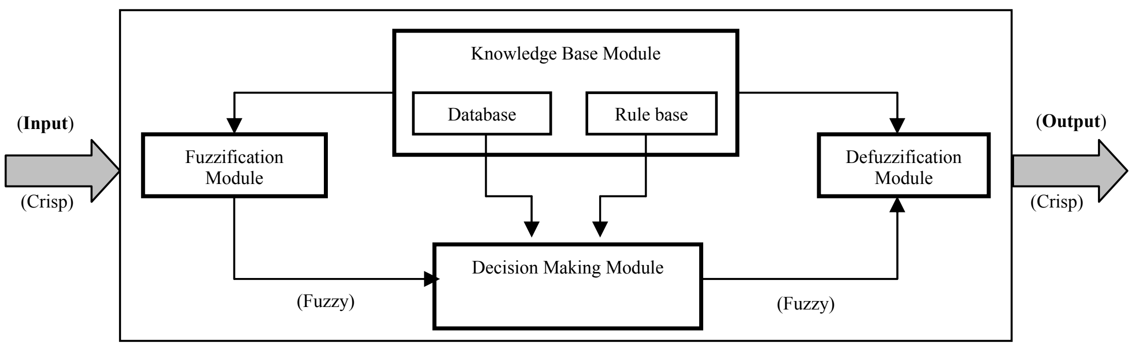


Fig.(3.2): Fussy Inference System (Jang, 1993)

The FIS working as follows: It compares the input variables with the membership functions on the antecedent part to obtain the membership values of each linguistic label. Then it combines the membership values on the premise part to get *firing strength* (*weight*) of each rule. After that it generates the qualified consequents or each rule depending on the firing strength. Finally it aggregates the qualified consequents to produce a crisp output.

3.4 Fuzzy Inference Methods

The most important two types of fuzzy inference method are Mamdani's fuzzy inference method, which is the most commonly seen inference method. This method was introduced by Mamdani and Assilian in 1975 (Mamdani et al, 1975). Another well-known inference method is the so-called Sugeno or Takagi–Sugeno–Kang method of fuzzy inference process. This method was introduced by Sugeno et al. in 1985 (Sugeno et al., 1985). This method is also called as TS method. The main difference between the two methods lies in the consequent of fuzzy rules. Mamdani fuzzy systems use fuzzy sets as rule consequent whereas TS fuzzy systems employ linear functions of input variables as rule consequent. All the existing results on fuzzy systems as universal approximators deal with Mamdani fuzzy systems only and no result is available for TS fuzzy systems with linear rule consequent.

3.4.1 Mamdani's Fuzzy Inference Method

Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology. Mamdani's method was among the first control systems built using fuzzy set theory. It was proposed by (Mamdani et al., 1975) as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Mamdani's effort was based on (Zadeh's, 1973) paper on fuzzy algorithms for complex systems and decision processes. Mamdani type inference, as defined it for the Fuzzy Logic Toolbox, expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification.

It is possible, and in many cases much more efficient, to use a single spike as the output membership function rather than a distributed fuzzy set. This is sometimes known as a *singleton* output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required by the more general Mamdani method, which finds the centroid of a two-dimensional function. Rather than integrating across the two-dimensional function to find the centroid, the weighted average of a few data points. Sugeno type systems support this type of model. In general, Sugeno type systems can be used to model any inference system in which the output membership functions are either linear or constant.

To compute the output of a Mamdani inference system given the inputs, six steps has to be followed:

1. Determining a set of fuzzy rules
2. Fuzzifying the inputs using the input membership functions
3. Combining the fuzzified inputs according to the fuzzy rules to establish a rule strength
4. Finding the consequence of the rule by combining the rule strength and the output membership function
5. Combining the consequences to get an output distribution
6. Defuzzifying the output distribution (this step is only if a crisp output (class) is needed).

3.4.2 Takagi–Sugeno Fuzzy Method (TS Method)

The Sugeno fuzzy model was proposed by Takagi Sugeno, and Kang (Sugeno et al., 1985) in an effort to formalize a system approach to generating fuzzy rules from an input–output data set. Sugeno fuzzy model is also known as Takagi- Sugeno (TS) model. Fuzzy rule systems are receiving more and more attention in the recent years. The main difference between TS models and Mamdani models lies in fact that the consequent part of TS fuzzy rules is real-valued function of the input variables instead of a fuzzy set. A typical fuzzy rule in a Sugeno fuzzy model has the format (Arafeh et al., 1999):

$$IF\ x\ is\ A\ and\ y\ is\ B\ THEN\ z = f(x, y)\ , \tag{3.1}$$

Where *AB* are fuzzy sets in the antecedent; $Z = f(x, y)$ is a crisp function in the consequent. Usually $f(x, y)$ is a polynomial in the input variables x and y, but it can be any other functions that can appropriately describe the output of the system within the fuzzy region specified by the antecedent of the rule. When $f(x, y)$ is a first-order polynomial, we have the *first-order* Sugeno fuzzy model. When f is a constant, we then have the *zero-order* Sugeno fuzzy model, which can be viewed as a special case of the Mamdani FIS where each rule's consequent is specified by a fuzzy singleton. Moreover, a zero-order Sugeno fuzzy model is functionally equivalent to a radial basis function network under certain minor constraints.

TS fuzzy rule systems have the following merits over the Mamdani fuzzy rule system (Jin, 2003):

1. TS rule systems are more suitable for several of learning algorithms.
2. TS rule systems have stronger representative power and therefore are capable of dealing with complex systems.

For STLF a typical rule in a MISO Sugeno fuzzy model with three inputs (Time, HiTemp, LowTemp) and one output (PowerLoad), has the form (Arafeh et al., 1999):

$$\begin{gathered} If\ Time\ is\ Time_j\ and\ Hi\text{-}Temp\ is\ HiTemp_k\ and\ Low\text{-}Temp\ is\ LowTemp_l,\ then \\ PowerLoad = p_i\ Time_j + q_i\ HiTemp_k + r_i LowTemp_l + s_i\ , \end{gathered} \tag{3.2}$$

Where ($_j$) represent the time input MF, ($_k$) represent the high temperature input MF, and ($_l$) represent the low temperature input MF. The terms p_i, q_i, r_i, s_i, indicate the consequent parameters. For a zero-order Sugeno model, the output level *PowerLoad* is a constant. The output level $PowerLoad_l$ of each rule is weighted by the firing strength w_i of the rule. For example, for an AND rule with Time = $Time_j$ and Hi-Temp = $HiTemp_k$ *and Low-Temp*= $LowTemp_l$, the firing strength is (MahWorks, 2008):

$$w_i = AndMethod\ (F_1(Time_j), F_2(HiTemp_k), F_3(LowTemp_l))\ , \tag{3.3}$$

Where $F_{1,2,3}$ *(.)* are the membership functions for Time ,Hi-Temp, and Low-Temp. The final output of the system is the weighted average of all rule outputs, computed as (MathWorks, 2008):

$$Final\ Output = \frac{\sum_{i=1}^{N} w_i\ PowerLoad_i}{\sum_{i=1}^{N} w_i}\ , \tag{3.4}$$

Fig.(3.3) shows how a Sugeno rule operates.

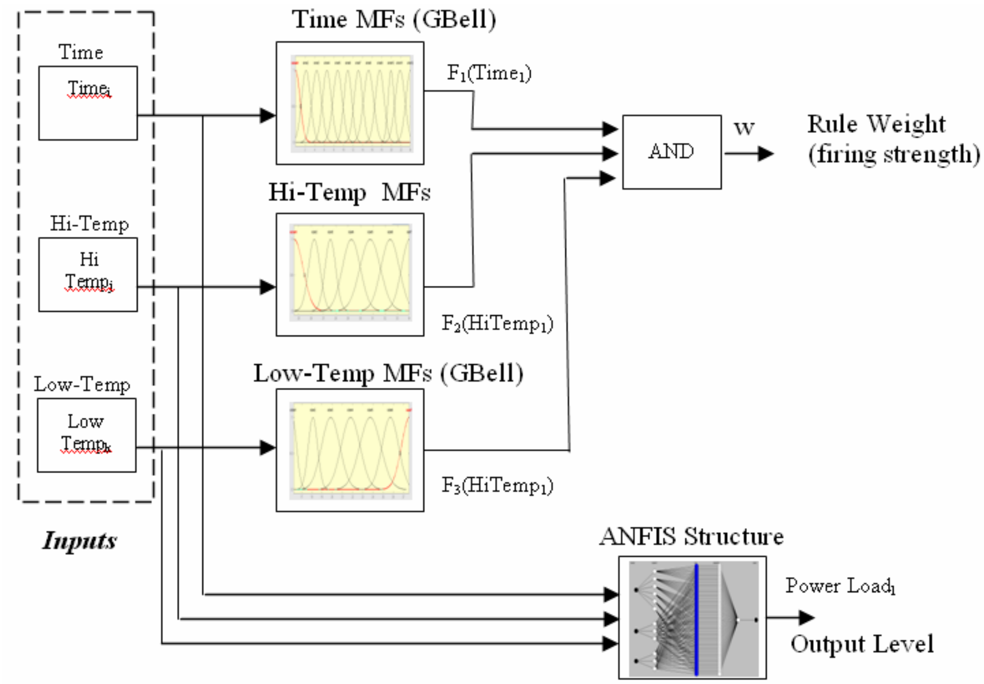


Fig.(3.3): Operation of a Sugeno Rule. *(MathWorks, 2008)*

3.5 Artificial Neural Networks

Artificial Neural Networks (ANNs) mimic biological information processing mechanisms. They are typically designed to perform a nonlinear mapping from a set of inputs to a set of outputs. ANNs are developed to try to achieve biological system type performance using a dense interconnection of simple processing elements analogous to biological neurons. ANNs are information driven rather than data driven. They are non-programmed adaptive information processing systems that can autonomously develop operational capabilities in response to an information environment. ANNs learn from experience and generalize from previous examples. They modify their behavior in response to the environment, and are ideal in cases where the required mapping algorithm is not known and tolerance to faulty input information is required.

ANNs contain electronic processing elements (PEs) connected in a particular fashion. The behavior of the trained ANN depends on the weights, which are also referred to as strengths of the connections between the PEs. ANNs offer certain advantages over conventional electronic processing techniques. These advantages are the generalization capability, parallelism, distributed memory, redundancy, and learning. Artificial neural networks are being applied to a wide variety of automation problems including adaptive control, optimization, medical diagnosis, decision making, as well as information and signal processing, including speech processing.

The first significant paper on artificial neural networks is generally considered to be that of McCullock and Pitts (McCullock et al., 1943) in 1943. This paper outlined some concepts concerning how biological neurons could be expected to operate. The neuron models proposed were modeled by simple arrangements of hardware that attempted to mimic the performance of the single neural cell. In 1949 Hebb (Hebb, 1949) formed the basis of 'Hebbian learning' which is now regarded as an important part of ANN theory. The basic concept underlying 'Hebbian learning' is the principle that every time a neural connection is used, the pathway is strengthened.

John Hopfield (Hopfield, 1982) produced a paper in 1982 that showed that the ANN had potential for successful operation, and proposed how it could be developed. This paper was timely as it marked a second beginning for the ANN. While Hopfield is the name frequently associated with the resurgence of interest in ANN it probably represented the culmination of the work of many people in the field. From this time onward the field of neural computing began to expand and now there is world-wide enthusiasm as well as a growing number of important practical applications.

The basic learning rule of the adaptive network is the well-known steepest descent method, in which the gradient vector is derived by successive invocations of the chain rule. This method for systematic calculation of the gradient vector was proposed independently several times, by Bryson and Ho (Bryson et al., 1969), Werbos (Werbos, 1974), and Parker (Parker, 1982). In 1986, Rumelhart et al. (Rumelhart et al., 1986) used the same procedure to find the gradient in a multilayer neural network. Their procedure was called the backpropagation learning rule, a name which is now widely known because the work of Rumelhart et al. inspired enormous interest in research on neural networks.

Today there are two classes of ANN paradigm, supervised and unsupervised. The multilayer back-propagation network (MLBPN) is the most popular example of a supervised network. It is a very powerful technique for constructing nonlinear transfer functions between several continuous valued inputs and one or more continuously valued outputs. The network basically uses multilayer perceptron architecture and gets its name from the manner in which it processes errors during training.

Adaptive Resonance Theory (ART) is an example of an unsupervised or self-organizing network and was proposed by Carpenter and Grossberg (Carpenter et al., 1987). Its architecture is highly adaptive and evolved from the simpler adaptive pattern recognition networks known as the competitive learning models. Kohonen's Learning vector quantiser (Kohonen, 1989) is another popular unsupervised neural network that learns to form activity bubbles through the actions of competition and cooperation when the feature vectors are presented to the network. A feature of biological neurons, such as those in the central nervous system, is their rich interconnections and abundance of recurrent signal paths. The collective behavior of such networks is highly dependent upon the activity of each individual component. This is in contrast to feed forward networks where each neuron essentially operates independent of other neurons in the network.

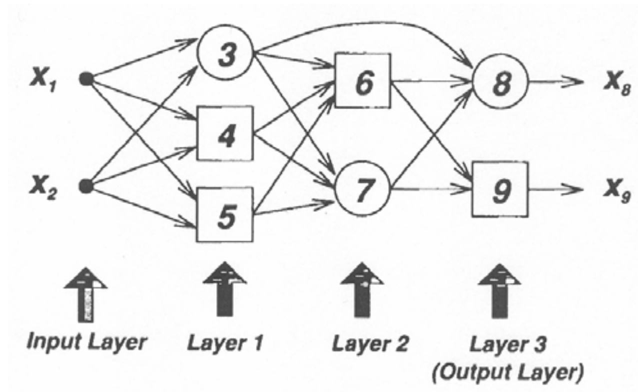


Fig.(3.4): A Feed Forward Adaptive Network in Layered Representation (Jang et al., 1997).

Fig.(3.4) represents an adaptive network, which is a network structure whose overall input-output behavior is determined by a collection of modifiable parameters. Specifically, the configuration of an adaptive network is composed of a set of nodes connected by directed links, where each node performs a static ***node function*** on its incoming signals to generate a single ***node output*** and each link specifies the direction of signal flow from one node to another. Usually a node function is a parameterized function with modifiable parameters; by changing these parameters, we change the node function as well as the overall behavior of the adaptive network is a network structure whose overall input-output behavior is determined by a collection of modifiable parameters.

3.5.1 Back-propagation Learning Algorithm

Back-propagation learning algorithm, or propagation of error, is a common method of teaching artificial neural networks how to perform a given task. It is a very powerful method to adjust the weights of the neural network It was first described by Paul Werbos in 1971 which he published in his doctoral thesis (Werbos, 1974), but it wasn't until 1986, through the work of (Rumelhart *et al.* 1986), when Rumelhart et al, rediscovered this technique, that it gained recognition, and it led to a "renaissance" in the field of artificial neural network research.

Back-propagation is a supervised learning method, and is an implementation of the Delta rule. It requires a teacher that knows, or can calculate, the desired output for any given input. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop) (Karayiannis, 1996). The term is an abbreviation for "backwards propagation of errors". Back-propagation requires that the activation function used by the artificial neurons (or "nodes") is differentiable.

Back-propagation is an iterative procedure which has three steps during each iteration as mentioned by Chen et al. (Chen, 2001):

1. *Forward*: The outputs are calculated for given inputs.
2. *Backward*: The errors at the output layer are propagated backwards toward the input layer, with the partial derivatives of the performance with respect to the weights and biases calculated in each layer.
3. *Weight adjustment*: A multivariate nonlinear numeric optimization algorithm finds the weights that minimize the error based on the gradient.

Training stops when the performance has been minimized to the goal, the performance gradient falls below a minimum gradient, the maximum number of epochs is reached, or the maximum amount of time has been exceeded.

3.6 Adaptive Neuro Fuzzy Inference System

Adaptive Neuro Fuzzy Inference System or semantically equivalently, Adaptive-network-based-fuzzy inference system (ANFIS) is considered to be an adaptive network which is very similar to neural networks, using a given input/output data set, ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are adjusted using either a backpropagation algorithm alone, or in combination with a least squares method. This allows the fuzzy systems to learn from the data they are modeling. The purpose of ANFIS is to integrate the best features of Fuzzy Systems and Neural networks. From Fuzzy Systems it is a representation of prior knowledge into a set of constraints to reduce the optimization search space. From Neural networks it is an adaptation of back propagation to structured network to automate the parametric tuning.

In fuzzy modeling, the membership functions and rule base are generally determined by trial-and-error approaches. Although this approach is straightforward, the determination of best fitting boundaries of membership functions and number of rules are very difficult. In order to calibrate the membership functions and rule base in fuzzy modeling, the neural networks have been employed by researchers (Jang, 1993), (Rong et al., 1996), (Ouyang et al., 2000), (Rojas et al., 2000), and (Guler et al., 2004). This system has been called fuzzy neural, neuro-fuzzy or adaptive network based system. The key properties of neuro-fuzzy systems are the accurate learning and adaptive capabilities of the neural networks, together with the generalization and fast-learning capabilities of fuzzy logic systems.

The basic structure of the type of fuzzy inference system is a model that maps input characteristics to input membership functions, input membership function to rules, rules to a set of output characteristics, output characteristics to output membership functions, and the output membership function to a single-valued output or a decision associated with the output.

3.6.1 ANFIS architecture

ANFIS architecture consists of five layers: fuzzy layer, product layer, normalized layer, de-fuzzy layer, and summation layer. In the fuzzy layer, the crisp input values are converted to the fuzzy values by the MFs. After, in the product layer, “and” operation is performed between the fuzzy values by using production so as to determine the weighting factor of each rule. Then, the normalized weighting factors are calculated in the normalized layer. In the de-fuzzy layer, the output rules are constructed. Finally, each rule is weighted by own normalized weighting factor and the output of the ANFIS is calculated by summing of all rule outputs in the summation layer.

Fig. (3.5) show the ANFIS architecture, two fuzzy if–then rules based on a first order Sugeno model are considered (Jang et al., 1997):

Rule 1: If (x is A_1) and (y is B_1) then ($f_1 = p_1x + q_1y + r_1$), (3.5)
Rule 2: If (x is A_2) and (y is B_2) then ($f_2 = p_2x + q_2y + r_2$), (3.6)

where x and y are the inputs, A*i* and B*i* are the fuzzy sets, f_i are the outputs within the fuzzy region specified by the fuzzy rule, p_i; q_i and r_i are the design parameters that are determined during the training process. The ANFIS architecture to implement these two rules is shown in Fig.(3.5), in which a circle indicates a fixed node, whereas a square indicates an adaptive node.

In the first layer, all the nodes are adaptive nodes. The outputs of layer 1 are the fuzzy membership grade of the inputs. In the second layer, the nodes are fixed nodes. They are labeled with M, indicating that they perform as a simple multiplier. In the third layer, the nodes are also fixed nodes. They are labeled with N, indicating that they play a normalization role to the firing strengths from the previous layer. In the fourth layer, the nodes are adaptive nodes. The output of each node in this layer is simply the product of the normalized firing strength and a first-order polynomial (for a first-order Sugeno model). In the fifth layer, there is only one single fixed node labeled with S. This node performs the summation of all incoming signals.

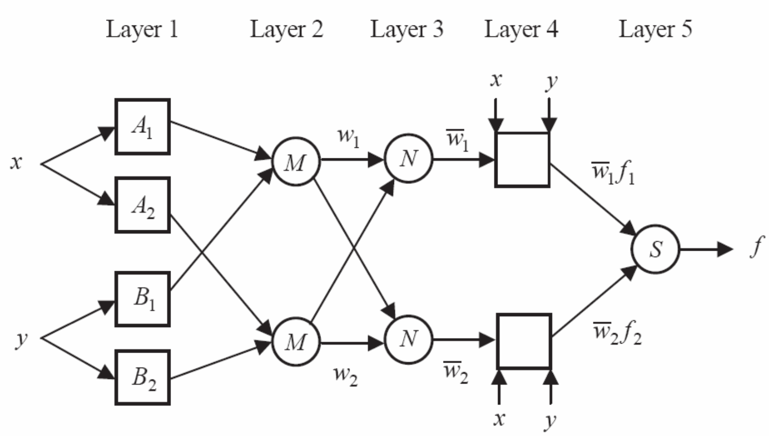


Fig.(3.5) : ANFIS Architecture (Lin et al., 2007).

3.6.2 Least-Squares Optimization Algorithm.

The Least-Squares (LSQ) optimization algorithm (Levenberg, 1944), (Marquardt, 1963), and (Dennis, 1977) is a mathematical optimization technique that attempts to find a function which closely approximates a given dataset. It tries to minimize the sum of the squares of the ordinate differences between points generated by the function and corresponding points in the dataset. The primary goal of the LSQ algorithm is to find a good estimator, which is unbiased and has a minimum variance.

3.6.3 Hybrid Learning Algorithm.

The Hybrid Learning (HL) algorithm (Jang, 1993) and (Jang et al., 1997), which combines the Gradient Descent (GD) and the LSQ algorithms, is one of the widely used algorithm in the literature to identify the parameters of the ANFIS. In the HL algorithm procedure, there are two passes which are called forward pass and backward pass. In the forward pass, functional signals go forward until the de-fuzzy layer and the consequent parameters are identified by the LSQ algorithm. In the backward pass, the error rates propagate backward and the premise parameters are updated by the GD algorithm.

3.7 Data Clustering

Data Clustering is considered an interesting approach for finding similarities in data and putting similar data into groups. Clustering partitions a data set into several groups such that the similarity within a group is larger than that among groups (Jang et al., 1997). The idea of data grouping, or clustering, is simple in its nature and is close to the human way of thinking; whenever we are presented with a large amount of data, we usually tend to summarize this huge number of data into a small number of groups or categories in order to further facilitate its analysis.

Moreover, most of the data collected in many problems seem to have some inherent properties that lend themselves to natural groupings. Nevertheless, finding these groupings or trying to categorize the data is not a simple task for humans unless the data is of low dimensionality (two or three dimensions at maximum.) This is why some methods in soft computing have been proposed to solve this kind of problem. Those methods are called “Data Clustering Methods”.

Clustering algorithms are used extensively not only to organize and categorize data, but are also useful for data compression and model construction. By finding similarities in data, one can represent similar data with fewer symbols for example. Also if we can find groups of data, we can build a model of the problem based on those groupings. Another reason for clustering is to discover relevance knowledge in data. Francisco Azuaje et al. (Azuaje et al., 2000) implemented a Case Based Reasoning (CBR) system based on a Growing Cell Structure

(GCS) model. Data can be stored in a knowledge base that is indexed or categorized by cases; this is what is called a Case Base. Each group of cases is assigned to a certain category. Using a Growing Cell Structure (GCS) data can be added or removed based on the learning scheme used. Later when a query is presented to the model, the system retrieves the most relevant cases from the case base depending on how close those cases are to the query.

Four of the most representative off-line clustering techniques frequently used in

1. K-means (or Hard C-means) Clustering,
2. Fuzzy C-means Clustering,
3. Mountain Clustering, and
4. Subtractive Clustering.

The first technique is *K-means* clustering (Hartigan et al., 1979) (or *Hard C-means* clustering, as compared to *Fuzzy C-means* clustering). This technique has been applied to a variety of areas, including image and speech data compression, (Lin et al., 1996) and (Tsoukalas et al., 1997) data preprocessing for system modeling using radial basis function networks, and task decomposition in heterogeneous neural network architectures (Nauck et al., 1997). This algorithm relies on finding cluster centers by trying to minimize a cost function of dissimilarity (or distance) measure.

The second technique is *Fuzzy C-means* clustering, which was proposed by Bezdek in 1973 (Jang et al., 1997) as an improvement over earlier Hard C-means clustering. In this technique each data point belongs to a cluster to a degree specified by a membership grade. As in K-means clustering, Fuzzy C-means clustering relies on minimizing a cost function of dissimilarity measure.

The third technique is *Mountain* clustering, proposed by Yager and Filev (Jang et al., 1997). This technique calculates a mountain function (density function) at every possible position in the data space, and chooses the position with the greatest density value as the center of the first cluster. It then destructs the effect of the first cluster mountain function and finds the second cluster center. This process is repeated until the desired number of clusters has been found.

The fourth technique is *Subtractive* clustering, proposed by Chiu (Jang et al., 1997). This technique is similar to mountain clustering, except that instead of calculating the density function at every possible position in the data space, it uses the positions of the data points to calculate the density function, thus reducing the number of calculations significantly.

3.7.1 Subtractive Clustering

Subtractive clustering solves the problem of mountain clustering, which its computation grows exponentially with the dimension of the problem, by using data points as the candidates for cluster centers, instead of grid points as in mountain clustering. This means that the computation is now proportional to the problem size instead of the problem dimension. However, the actual cluster centers are not necessarily located at one of the data points, but in most cases it is a good approximation, especially with the reduced computation this approach introduces.

Consider a collection of n data points $\{x_1,\ldots,x_n\}$ in an M-dimensional space. Since each data point is a candidate for cluster centers, a density measure at data point x_i is defined as (Guldemair et al., 2006):

$$D_i = \sum_{j=1}^{n} \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{\left(r_a / 2 \right)^2} \right), \tag{3.6}$$

where r_a is a positive constant representing a neighborhood radius. Hence, a data point will have a high density value if it has many neighboring data points. The first cluster center x_{c1} is chosen as the point having the largest density value D_{c1}. Next, the density measure of each data point x_i is revised as follows (Guldemir et al., 2006):

$$D_i = D_i - D_{c1} \ \exp\left(-\frac{\left\| x_i - {x_{c1}}^2 \right\|}{\left(r_b / 2 \right)^2} \right), \tag{3.7}$$

where r_b is a positive constant which defines a neighborhood that has measurable reductions in density measure. Therefore, the data points near the first cluster center x_{c1} will have significantly reduced density measure. After revising the density function, the next cluster center is selected as the point having the greatest density value. This process continues until a sufficient number of clusters is attainted. The constant r_b is normally larger than r_a to prevent closely spaced cluster centers; generally r_b is equal to 1.5 $r_{a,}$ as suggested in (Tsoukalas et al , 1997).

After the density measure for each data point is revised, the next cluster center x_{c2} is selected and all of the density measures for data points are revised again. This process is repeated until a sufficient number of cluster centers are generated.

Chapter Four:

Historical Data Profiles and Preprocessing

4.1 Introduction

As mentioned in chapter two and three, developing any supervised-based soft computing model needs pairs of data (inputs and output), and in order to have a reliable STLF models that best represents the trends of these input and output data, we need reasonable actual sets of data composed of the electric power load as an output for a certain time during a day with known weather conditions as an inputs for a specific line that provides a chosen area. Many weather variables are considered in building the load forecasting model. The selected variables, however, may not be the same for all seasons, load shapes or day types. In some cases more and different variables may be used because these are more significant than others (Rahman, 1990). Khotanzad (Khotanzad et al., 1996) has reported that, temperature is the primary weather variable affecting the load and if the hourly weather forecasts are unavailable, we can still train the models using the daily high and low values of weather temperatures.

The selection of the training datasets from the available data significantly affects the forecasting results, and to achieve a reliable and a more comprehensive approach to load forecasting, the days which have similar load and historical temperature values should be chosen to train (develop) the models (Peng et al., 1992). Peng (Peng et al., 1990) developed a "moving window" method was presented to select the training datasets. In that method, the immediate previous two weeks of data were selected for training and the load was forecasted for the present week. The results using this technique showed relatively large average errors due to the complications from holidays and unusual events. Park (Park et al., 1990), on the other hand, selected the training cases for a 6-day period in the same time frame and performed daily forecasts using very limited simulations. Further more, Khotanzad (Khotanzad et al., 1998) three years of historical hourly load and weather data required to train the model.

A technique based on the application of knowledge-based systems to the problem of load forecasting was introduced by Rahman and Bhatnagar (Rahman et al., 1988), and (Bhatnagar et al., 1986). They developed an algorithm for the six-hour load forecast and another algorithm for the 24-hour load forecast. Such algorithms were developed "based on the logical and syntactical relationships between the weather and prevailing daily load shapes" (Rahman et al., 1988).

To avoid over-training problem (trained model fits the training samples perfectly "error reduced to 0" but it does not give accurate outputs for inputs not in the training set), the cross-validation method should be used (Khotanzad et al., 1998). They have been divided the datasets into two sets. For instance, if three years of data is available, it is divided into a two-year and a one-year set. The first set is used to train the model and the second set is used to test the trained model.

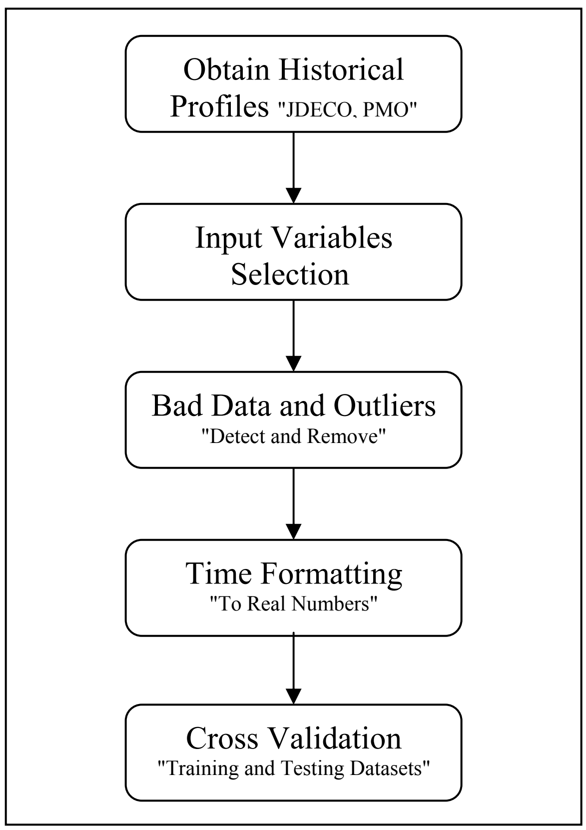


Fig. (4.1): Data Preprocessing Procedure Stages.

Many factors affect the success of model training on a given task. The representation and quality of the instance data is first and foremost (Pyle, 1999). If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. It is well known that data preparation and filtering steps take considerable amount of processing time in model training. Data pre-processing includes data cleaning, normalization, transformation and selection, etc. The product of data pre-processing is the final training set. Kotsiantis (Kotsiantis et al., 2006) present a well know algorithm for each step of data pre-processing.

As mentioned in the previous chapters, the sources of the available datasets profiles are Jerusalem District Electric Company (JDECO) and Palestinian Meteorology Office (PMO). Pre-processing stages have been accomplished for the collected data. These stages are as shown in Fig.(4.1). They are, obtaining historical profiles, input variables selection, bad data and outliers detecting and removing, time formatting, and cross validation. In the obtaining historical profiles stage, an actual historical load and weather elements profiles for two years (2006 and 2007) have been obtained from (JDECO) and (PMO). In the second stage, the time, power load, and temperature elements (low and high temperatures) have been selected to train the models. The reason for choosing these parameters as inputs and their effects on the predicted load will be discussed in details on the following chapter. Fig (4.2), shows the effects of the temperature parameters on the predicted load.

After that an existing algorithm for detecting and removing the outliers from the historical profiles has been used, and a manual procedure has been followed for detecting and removing the bad data such as the zero loads. Time formatting in the fourth stage is necessary since the input to the models should be a real number format and not in a time format (hour: minutes) as provided to us. Finally, a cross validation technique has been applied to divide the datasets into training and testing datasets.

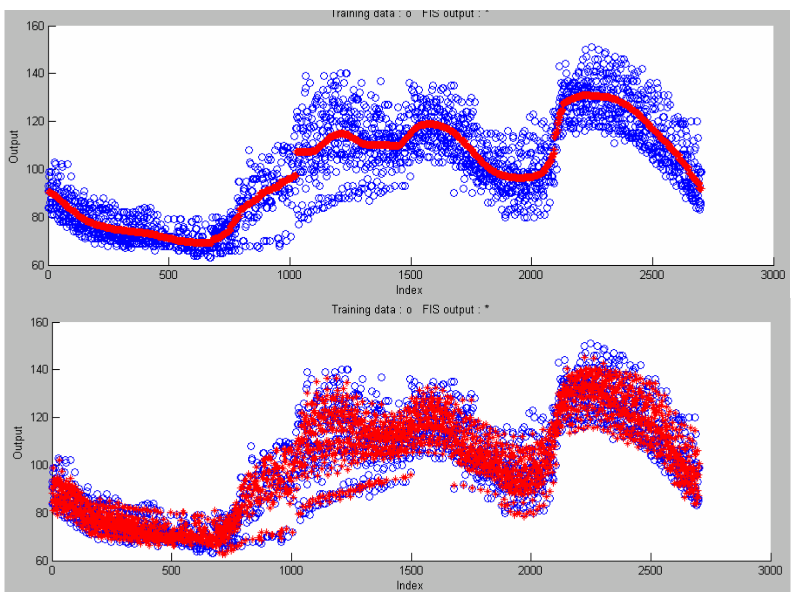
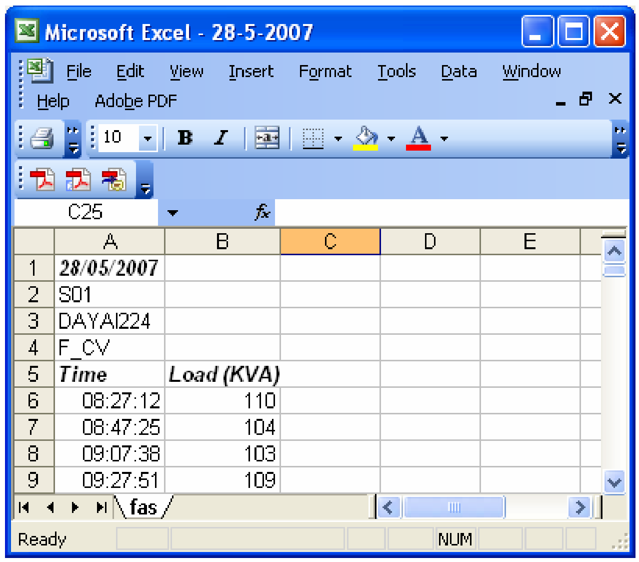


Fig. (4.2): The Effect of the Temperature Parameters on the Predicted Load. The Upper Plot Represent the Output from a SISO Sugeno Model with Hybrid Optimization. The Lower One Represent the Output from a MISO Sugeno Model with Hybrid Optimization.

A detailed discussion for these stages is presented in the following sections.

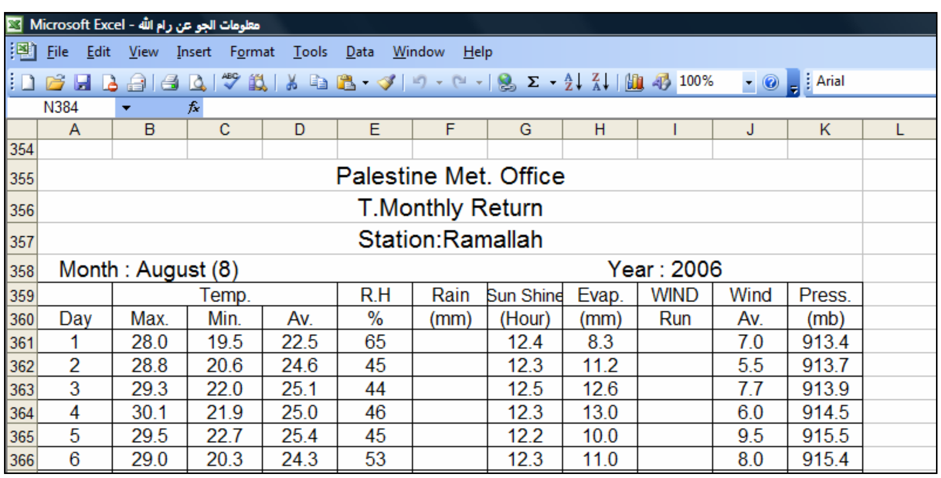
4.2 Historical Data Profiles

Real JDECO for a power line in Bier Nabala village and PMO historical data profiles for two years (2006 and 2007) collected and used to develop and test our models. The JDECO data includes the time (a reading at every 20 minutes approximately) and the corresponding power load at that time. Furthermore, the weather history data, provided by PMO, including humidity, highest temperature, lowest temperature, and wind speed for each day. Fig. (4.3) and (4.4) show a sample for the provided data profiles. The provided power loads and weather elements actual profiles are in Excel format (.xls).



	A	B	C	D	E
1	*28/05/2007*				
2	S01				
3	DAYAI224				
4	F_CV				
5	*Time*	*Load (KVA)*			
6	08:27:12	110			
7	08:47:25	104			
8	09:07:38	103			
9	09:27:51	109			

Fig. (4.3): A Sample for Actual Power Loads Profile as Provided by JDECO



Palestine Met. Office
T.Monthly Return
Station:Ramallah
Month : August (8) Year : 2006

Day	Temp. Max.	Temp. Min.	Temp. Av.	R.H %	Rain (mm)	Sun Shine (Hour)	Evap. (mm)	WIND Run	Wind Av.	Press. (mb)
1	28.0	19.5	22.5	65		12.4	8.3		7.0	913.4
2	28.8	20.6	24.6	45		12.3	11.2		5.5	913.7
3	29.3	22.0	25.1	44		12.5	12.6		7.7	913.9
4	30.1	21.9	25.0	46		12.3	13.0		6.0	914.5
5	29.5	22.7	25.4	45		12.2	10.0		9.5	915.5
6	29.0	20.3	24.3	53		12.3	11.0		8.0	915.4

Fig. (4.4): A Sample for Actual Weather Elements Profile as Provided by PMO.

4.3 Input Variables Selection

As mentioned above, and in the literature survey, three main inputs (Time (T), High Temperature (HT), and Low Temperature (LT)) and one output (Power Load (PL)) have been considered in building our models. For SISO models the time has been adopted as an input, and the forecasted power load as an output. To build the MISO models, three inputs (T, HT, LT), and one output (PL) have been adopted. Fig. (1.3) and (1.4) in chapter one represent the input and the output for the SISO and MISO models.

4.5 Bad Data and Outliers

The obtained training data are always subject to noise or maybe outliers (Chuang et al., 2004). Outliers are ''an observation or set of observations which appears to be inconsistent with the remainder of that set of data'' (Barnett et al, 1994). The existence of bad data in historical load curve affects the precision of load forecasting result. There are two kinds of bad data in the daily load curve: false channel bad data and abnormal event bad data. False channel bad data are due to the measurement and transmission mistakes, and they are far from their real physical values. Abnormal event bad data come from some unexpected sudden incidents, such as short circuit and equipment overhaul, which cause unnatural sudden changes of the load curve trend. According to the continuous time of the bad data appearance they can be put into two categories: long-last bad data and short-period bad data. Fig. (4.5) and Fig. (4.6) show these two kinds of bad data.

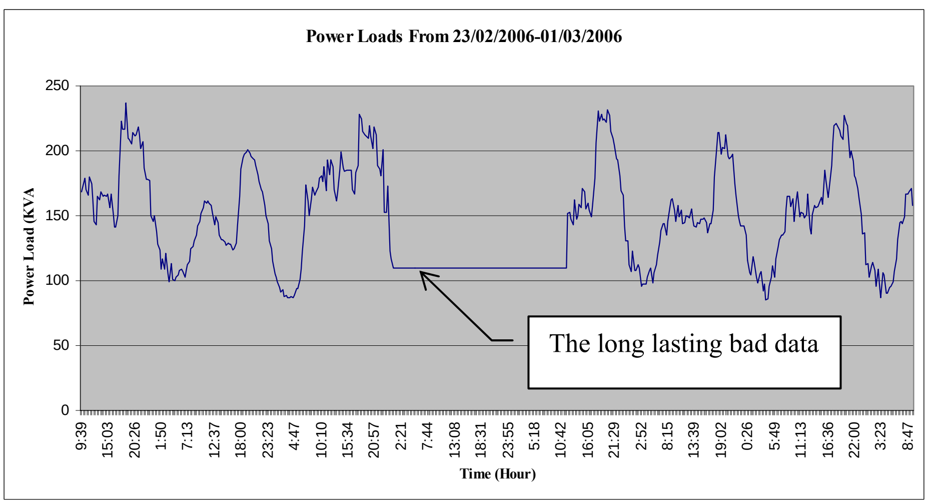


Fig. (4.5): Long-lasting Bad Data (23/02/2006-01/03/2006).

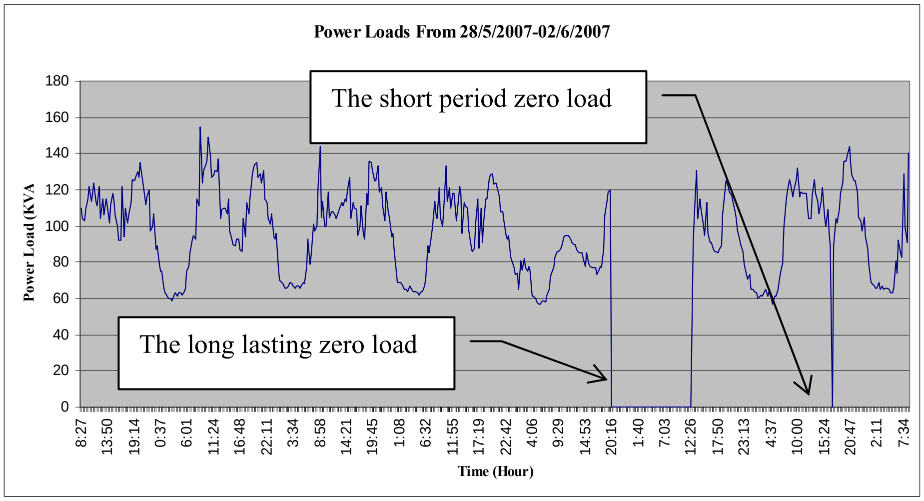


Fig. (4.6): Long-lasting and Short-period Bad Data "Zero Loads" (28/05/2007-02/06/2007).

A pre-processing stage was accomplished for the collected data. The bad data such as zero loads for a long period had been removed manually. The load profiles have been plotted for each month and then we have been scanned the plot searching for the zero loads. In addition, the outliers in the collected data have been identified and removed using an existing algorithm to neglect it in the training stage. The basic idea in detecting the outliers using this algorithm is by calculating the mean and the standard deviation. The point considered to be an outlier if it has a larger absolute difference between them and the mean divided by the standard deviation than lambda. Lambda is a factor and calculated based on the equations in (Yu et al., 2004). This algorithm has been chosen after trying many algorithms, but this one has been give us the best results for a huge datasets as in our case. For more details about the algorithm see Appendix A.

Fig. (4.7) depicts a sample for detecting the outliers for (two months 7/2006, 7/2007) which was used to develop our July models before removing the outliers. The figure shows the original signal for the first two days of the datasets (01-02/07/2006). Three clear outliers which have been detected and removed were circled.

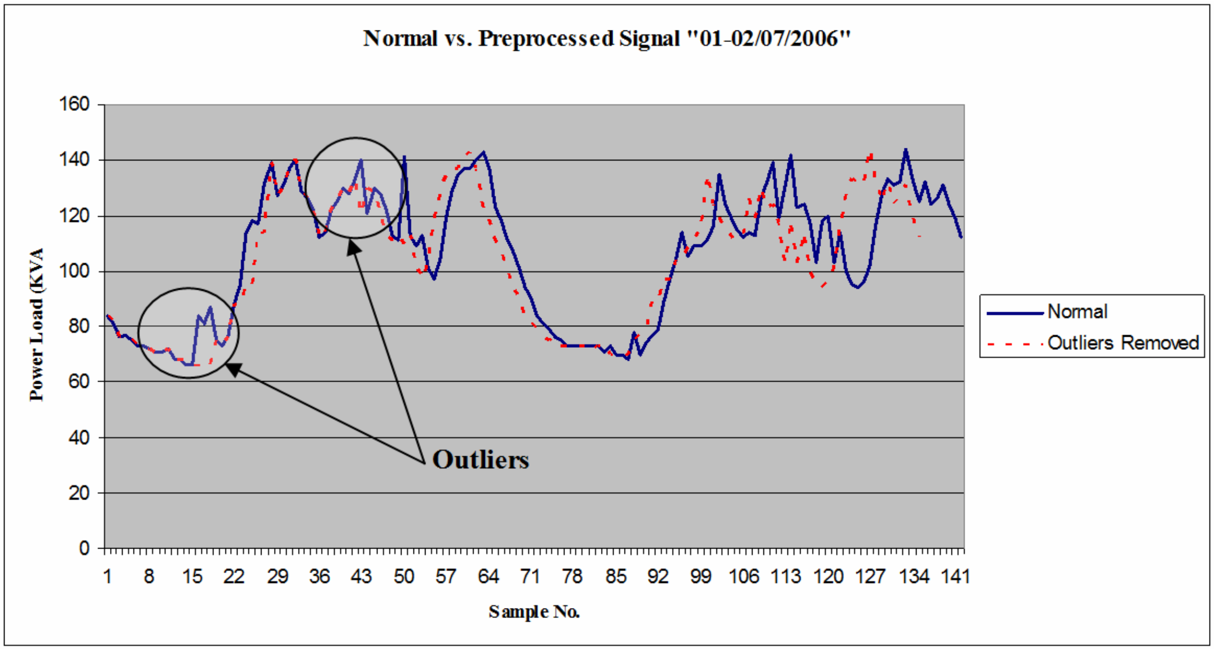


Fig. (4.7): The Original Dataset Before and After Removing the Outliers.

A correlation measures have been applied between the two signals to ensure that there is no loss in signal details after removing the outliers. The correlation found equal to 0.998 which shows that there is no crucial loss in the signal details.

4.6 Time Formatting

To build the models, the inputs and the output for the models should be in real numbers format. The provided historical power profiles from JDECO containing the time in time format (hour: minutes: seconds), so a conversion to real numbers (ranging from 00.00 to 24.00) needed. To do that, we have been converted each hour and minute to a real number of the form (1.0 to 24.0). For example if we have the time 12:30 it is converted to 12.50 and 23:20 it is converted to 23.33 and so on.

4.7 Cross Validation

The basic idea used in the selection of training (Tr) and testing (Ts) datasets can be expressed as the following.

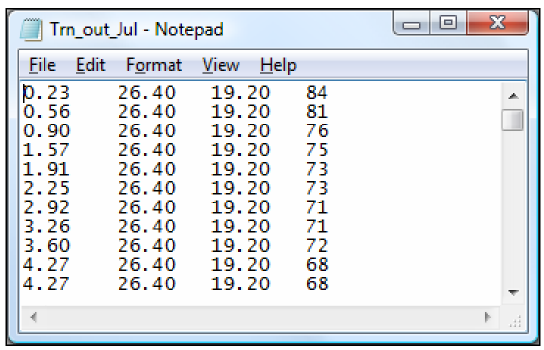
1. Prepare a complete historical (power and weather) profiles. For the obtained profiles we have complete ones for the months April, May, July and August from the years 2006 and 2007.

2. A cross validation algorithm to avoid the over training problem (Khotanzad et al., 1998) has been developed to distribute the available datasets into Tr and Ts datasets.

The developed algorithm works by initiating two matrices (the first one used to store the training datasets and the second one to store the testing datasets). Then the algorithm scans the available datasets profile and selects recursively three elements for training and moving them to Tr datasets matrix, and then moving the fourth one to the testing datasets matrix. The algorithm repeats the process until it has been reached the end of the profile (see Appendix A).

3. Save the files which containing the sets for training and testing in a DAT format, where in training the SISO FIS models the first column (the time) is considered as the input, and the last column of the file is considered as the desired (Actual) output of the model. In training the MISO models the first three columns (the time, HT, and LT) are considered as the inputs for the model, and the fourth column is considered as the desired (Actual) output. The same thing should be considered for the testing datasets.

Fig. (4.8) represent a sample for a training dataset file (.dat). The time is in real number format (0.0 – 24.0), the second and third columns are the HT and LT (in C◦), while the last column represent the actual power load (in KVA).



0.23	26.40	19.20	84
0.56	26.40	19.20	81
0.90	26.40	19.20	76
1.57	26.40	19.20	75
1.91	26.40	19.20	73
2.25	26.40	19.20	73
2.92	26.40	19.20	71
3.26	26.40	19.20	71
3.60	26.40	19.20	72
4.27	26.40	19.20	68
4.27	26.40	19.20	68

Fig. (4.8): A Sample from a Training Dataset File for July MISO model.

Table (4.1), shows a summery of the available datasets which have been used to develop the models from the years 2006 and 2007. In addition, summery for the ranges of the used datasets in developing our models are shown on Table (4.2).

Table (4.1): Summery of the Available Datasets for the Years 2006 and 2007.

Season	Month	Total No. of Datasets	No. of Training Datasets (Tr)	No. of Testing Datasets (Ts)
Spring	April	4008	3006	1002
	May	4074	3056	1018
	Total	8082	6062	2020
Summer	July	3634	2734	900
	August	4074	3056	1018
	Total	7708	5790	1918

Table (4.2): Available Datasets Profiles Ranges for the Developed Models.

Month/Season	Time	Hi Temperature (C°)	Low Temperature (C°)
May	[0.00 – 24.00]	[17.0 – 35.0]	[09.2 – 25.0]
July	[0.01 – 24.00]	[24.6 – 34.0]	[16.2 – 27.6]
Spring	[0.00 – 24.00]	[08.9 – 35.0]	[05.8 – 25.0]
Summer	[0.01 – 24.00]	[24.6 – 34.0]	[16.0 – 27.6]

4.8 Extra Testing Datasets from the Year 2008

In order to test our developed models using new unseen datasets, new profiles have been obtained for the year 2008 from JDECO and PMO for using the developed models to predict the power load for one day and one week. A small two samples selected to test the developed models. The first one is for one week from July (01-07/07/2008) to test the general July and Summer MISO models. The second datasets are for one week from May (0-07/05/2008) to test the general May and Spring MISO models.

Chapter Five:

Development of Models and EPLFS System

5.1 Introduction

In system modeling and identification, the important steps are to identify structure and parameters of the system based on the available data. The structure identification itself can be considered as two types, identification of the input variables of the model and the input–output relation. Most of the modeling approaches consider the input variables as a known priori and hence only the input and output relation has to be found (Satish et al, 2004).

When a number of possible combinations of input variables exist, the decisive input variables that affect the output should be chosen. Thus, the results in Table (5.1) show that the input variable temperatures have a considerable effect on load forecasting. The results in the table are for two kinds of models, Single Input Single Output (SISO) and Multiple Inputs Single Output (MISO) Sugeno models with hybrid optimization. The adequacy of the two developed models has been checked using the *Correlation Coefficient* (CC) and two error measures; *Mean Absolute Performance Error* (MAPE), and the *Root Mean Square Error* (RMSE).

Table (5.1): Temperature Effect on Load Forecasting.

Type of Model	No. of MFs	Training Dataset Errors			Testing Dataset Errors		
		CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
MISO	12 7 7	0.9815	0.0257	0.0769	0.9719	0.0324	0.1715
SISO	12	0.9106	0.0667	0.1662	0.9085	0.0656	0.3045

The models are to be trained with the historical data before testing them. The first step for training a model is obtaining an accurate historical data. In addition, data should be chosen that is relevant to the model. Several models have been developed such as, Sugeno with different optimization techniques, Subtractive Clustering, Subtractive Clustering with Hybrid optimization technique.

As it was mentioned in the previous chapter, real JDECO's for a power line in Bier Nabala village and PMO historical data profiles for two years (2006 and 2007) collected and used to develop and test our models. Using these historical datasets, different SISO and MISO models have been developed to predict the load for one day or one weak ahead in a specific month or season. In monthly case, several models with different optimization techniques have been developed for May and July. In Seasonal case, several models with different optimization techniques have been developed for Spring and Summer.

In order to deeply study the effect of the temperature in the Short Term Load Forecasting (STLF), two kinds of models have been developed, SISO and MISO models. For the SISO models that shown in Fig. (5.1) the time has been used as the input for the models and the power load at that time has been used as the output. In the MISO models which shown in Fig. (5.2), three variables (Time (T), High Temperature (HT), and the Low Temperature (LT) for that day) have been used as an input for the developed models, and the power load at that time was considered as the model output.

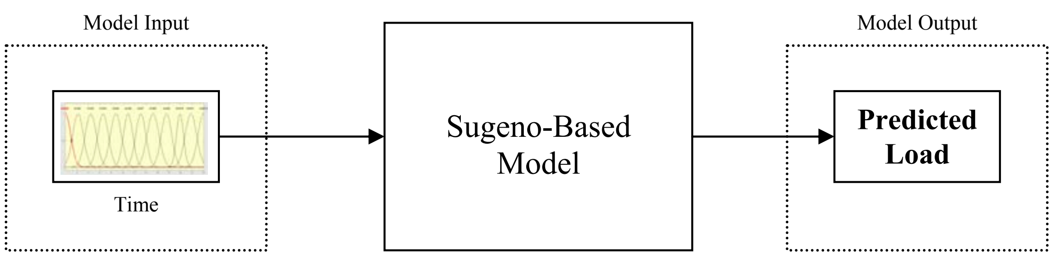


Fig. (5.1): SISO Sugeno FIS Model Architecture.

As mentioned in chapter three for the SISO Sugeno Fuzzy model, a typical rule with one input (Time) and one output (PowerLoad), has the form (Arafeh et al., 1999):

$$\text{If } Time \text{ is } Time_j \text{ then } PowerLoad = p_i\, Time_j + q_i \,, \tag{5.1}$$

Where ($_j$) represents the time input MF, and the terms p_i, q_i, indicate the consequent parameters. For a zero-order Sugeno model, the output level *PowerLoad* is a constant.

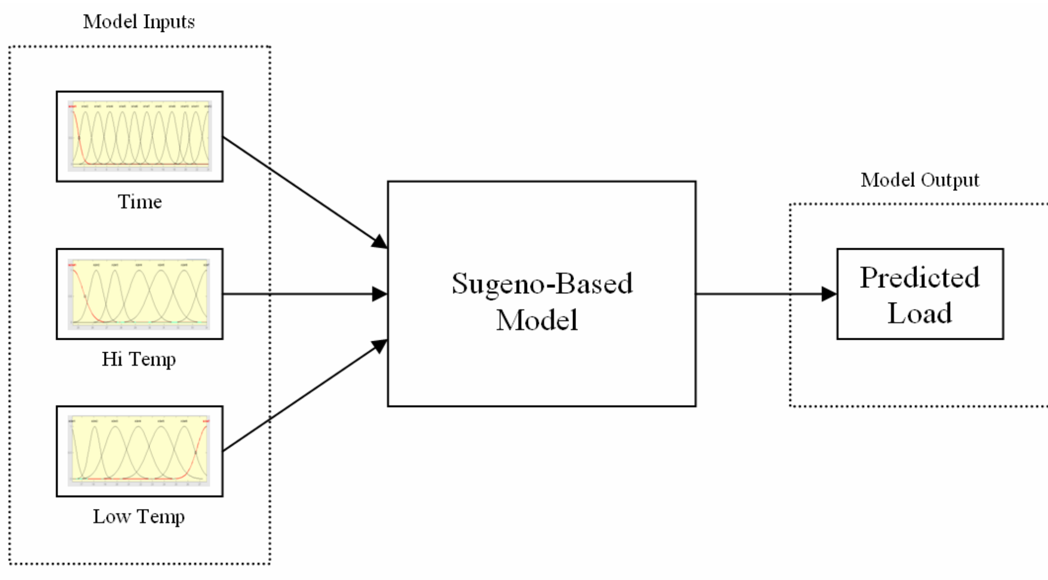


Fig. (5.2): MISO Sugeno FIS Model Architecture.

Whereas; For the STLF MISO Sugeno Fuzzy model a typical rule with three inputs (Time, HiTemp, LowTemp) and one output (PowerLoad), has the form (Arafeh et al., 1999):

If Time is $Time_j$ *and HiTemp is* $HiTemp_k$ *and LowTemp is* $LowTemp_l$*, then*
$PowerLoad = p_i\ Time_j + q_i\ HiTemp_k + r_i LowTemp_l + s_i$, (5.2)

Where (j) represent the time input MF, (k) represent the high temperature input MF, and (l) represent the low temperature input MF. The terms p_i, q_i, r_i, s_i, indicate the consequent parameters. For a zero-order Sugeno model, the output level *PowerLoad* is a constant. The output level $PowerLoad_l$ of each rule is weighted by the firing strength w_i of the rule. For example, for an AND rule with Time = $Time_j$ and Hi-Temp = $HiTemp_k$ *and Low-Temp*= $LowTemp_l$, the firing strength is (MathWorks, 2008):

$$w_i = AndMethod\ (F_1(Time_j),\ F_2(HiTemp_k), F(LowTemp_l))\ , \qquad (5.3)$$

Where $F_{1,2,3}$ *(.)* are the membership functions for Time ,Hi-Temp, and Low-Temp. The final output of the system is the weighted average of all rule outputs, computed as (MathWorks, 2008):

$$Final\ Output = \frac{\sum_{i=1}^{N} w_i\ \mathrm{PowerLoad}_i}{\sum_{i=1}^{N} w_i}\ , \qquad (5.4)$$

Fig. (5.3) below illustrates a general developing "training" block diagram of our models. It consists of three main stages. These stages can be summarized as follow:

1. The first stage is pre-processing the input datasets for the system. These datasets include four elements; three of them are inputs; namely, the time, the high temperature, and the low temperature of the day and one output (the actual loads).

2. The second stage is concerned with various soft computing models that have been developed. SISO/MISO models will be developed using different techniques (Hybrid and Back-propagation optimization techniques, Subtractive Clustering and by cascading two models). The same datasets (Training and Testing) have been used in developing all the models.

3. The third stage checks the adequacy of the developed models to demonstrate their performance.

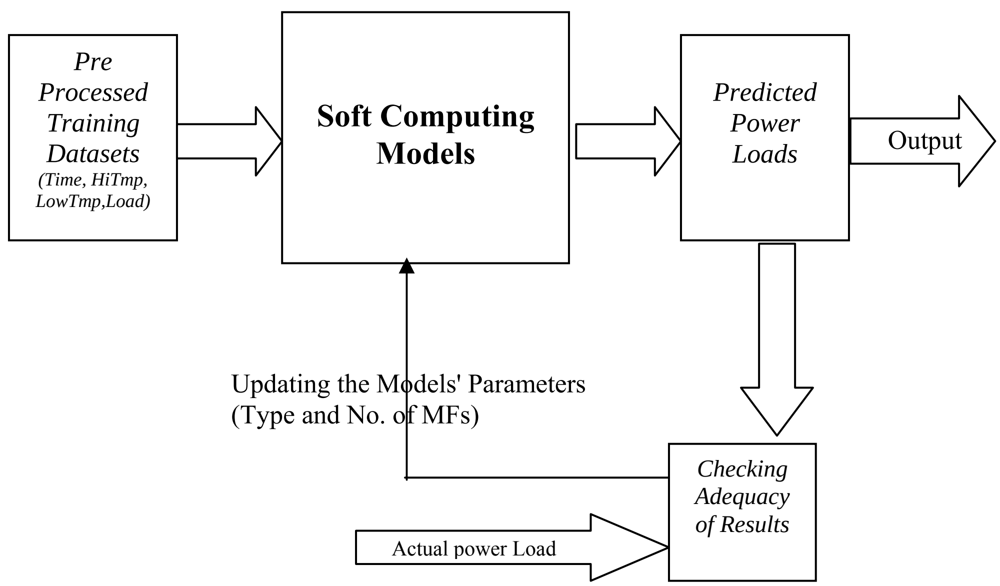


Fig. (5.3): A General Block Diagram for Developing/Training Soft Computing Models

Three measures according to (Khotanzad et al., 1998), (Arafeh et al., 1999), (Desouky et al, 2000), (Hwang et al., 2001), (Chen et al., 2001), (Chen et al., 2004), (McSharry, 2006), (Oriqat, 2007) have been used to effectively check the adequacy of results, and these measures illustrated bellow:

- The CC measure between actual and predicted power loads. It indicates the strength and direction of a linear relationship between the forecasted and actual loads and calculated by (Arafeh et al, 1999):

$$CC_{xy=}\sqrt{1-\frac{\sum_{i=1}^{N}(y_i-x_i))^2}{\sum_{i=1}^{N}(y_i-y)^2}} \quad , \tag{5.5}$$

where y_i : is the i[th] actual data,

y : is the average of all actual data,

x_i : is the i[th] predicted data.

N : is the number of data points under consideration.

- The Mean Absolute Percentage Error (MAPE), which has been traditionally used to measure accuracy in load forecasting (McSharry, 2006). It captures the proportionality between the forecast error and the actual load. The MAPE is calculated by (McSharry, 2006):

$$\text{MAPE}= \sum_{i=1}^{N} \left| \frac{y_i - x_i}{y_i} \right| * \frac{100}{N} \% \quad , \tag{5.6}$$

- The Root Mean Square Error (RMSE), which is used to evaluate the error (differences) between the forecasted and actual loads. The general form of the RMSE equation for the actual power loads (Y) and the predicted ones (X) in given by (Oriqat, 2007):

$$\text{RMSE}= \frac{\sqrt{\sum_{i=1}^{N} (y_i - x_i)^2}}{(N-1)} \quad , \tag{5.7}$$

5.2 SISO and MISO Sugeno Models with Hybrid Optimization Technique

As mentioned in chapter three the hybrid learning algorithm (Jang, 1993) and (Jang et al., 1997), which combines the Gradient Descent and the Least-Squares algorithms, is one of the widely used algorithm in the literature to identify the parameters of the ANFIS.

As shown in Fig (5.1) and Fig (5.2) above, in the SISO model the time has been used as the input for the model and the power load at that time has been used as the output. In the MISO model, three variables (time, High Temperature, and the Low Temperature for that day) have been used as an input for the developed models, and the power load at that time was considered as the model output.

In order to obtain the best results from the developed models, the model parameters (type and number of membership functions) need to be updated and determined manually to be fixed for the all models as shown in Fig. (5.3). The predicted loads will be measured against the actual marks and the system parameters will be altered to find the best outcome. We have been tried several Membership Functions (MFs) including: Gaussian Curve, Generalized Bell, Trapezoidal and Triangular. Table (5.2) lists the results that obtained form a July SISO model with different MFs and the same parameters (No. of MFs, hybrid optimization technique, training datasets and testing datasets).

Table (5.2): Results for a July SISO Model with Different Types of MFs.

Type of MF	No. of MFs	Training Dataset Errors			Testing Dataset Errors		
		CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
Gaussian Curve	12	0.9104	0.0667	0.1664	0.9094	0.0654	0.3032
Trapezoidal		0.9100	0.0669	0.1667	0.9088	0.0656	0.3041
Triangular		0.9094	0.0672	0.1672	0.9079	0.0661	0.3055
Generalized Bell		0.9106	0.0667	0.1662	0.9085	0.0656	0.3045

As listed in Table (5.2), there is no major difference between the outputs (predicted loads) regarding the type of the membership function. However, the MF that produced the best results is found to be the Generalized Bell (GBell) which has the following equation (Mathworks, 2008):

$$f(x,a,b,c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \quad , \tag{5.8}$$

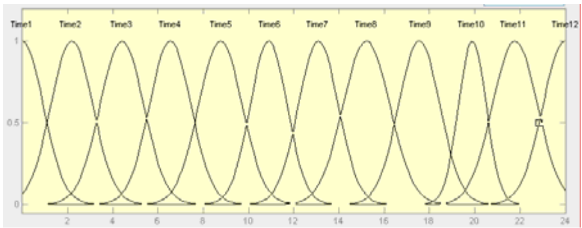
Where, the parameter b is usually positive, the parameter c locates the center of the curve and the parameters (a, b) vary the width of the curve. Accordingly, we will fix the MF of the type GBell to be used throughout the developed models.

Table (5.3) lists the results that obtained form a July MISO model with different number of MFs and the same parameters (Type of MFs (GBell), hybrid optimization technique, training datasets and testing datasets).

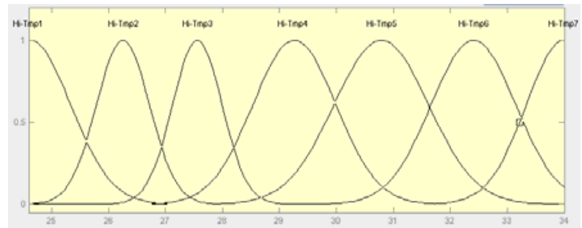
Table (5.3): Results for a July MISO Model with Different Number of MFs.

No. of Inputs MFs			Training Dataset Errors			Testing Dataset Errors		
T	HT	LT	*Corr.*	*MAPE*	*RMSE*	*Corr.*	*MAPE*	*RMSE*
6	6	6	0.9252	0.0593	0.1526	0.9307	0.0570	0.2666
8	6	6	0.9511	0.0466	0.1242	0.9528	0.0449	0.2212
12	6	6	0.9521	0.0452	0.1229	0.9511	0.0448	0.2250
12	***7***	***7***	***0.9815***	***0.0257***	***0.0769***	***0.9719***	***0.0324***	***0.1715***
12	8	8	0.9561	0.0428	0.1179	0.9548	0.0440	0.2167

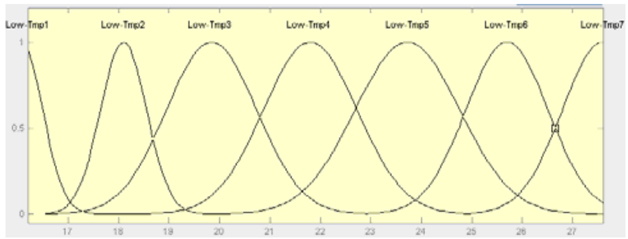
It is clear from the table above that the best results obtained when 12 MFs have been used for the time input, and 7 MFs for the temperature inputs (High and Low). According to the results shown in Table (5.3), we will fix the number of the MF's in the proposed models to 12 MF's for the time input and 7 MF's for the Low and High temperatures inputs. Fig.(5.4) represents the inputs MFs that we have been used in building a July MISO model.



(a): Time Input MFs



(b): High Temperature Input MFs



(c): Low Temperature Input MFs

Fig. (5.4): MISO Model Inputs (Time, High Temperature, Low Temperature) MFs

Table (5.4) shows the results obtained from a two July MISO Sugeno model with hybrid optimization the first one developed using all the training datasets available. The second one developed using the training datasets after removing the outliers. The results obtained show the effect of the outliers to the accuracy of the models. The model that has been developed before removing the outliers shows a correlation value for the training datasets of 0.9414 while the correlation value for the model that has been developed after removing the outliers is 0.9815.

Table (5.4): Results for a July MISO Model Before and After Removing the Outliers.

Datasets Used	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
All the Datasets	0.9414	0.0510	0.1336	0.9255	0.0568	0.2585
Outliers Removed	0.9815	0.0257	0.0769	0.9719	0.0324	0.1715

Using the parameters that give us the best results and the outliers have been removed from the datasets, Table (5.5) shows the results obtained for the training and testing datasets from monthly (May, July) and seasonal (Spring, Summer) SISO Sugeno models with hybrid optimization. Table (5.6) shows the results obtained from the monthly and seasonal MISO Sugeno models with hybrid optimization.

Table (5.5): Results Obtained from the SISO Models with Hybrid Optimization.

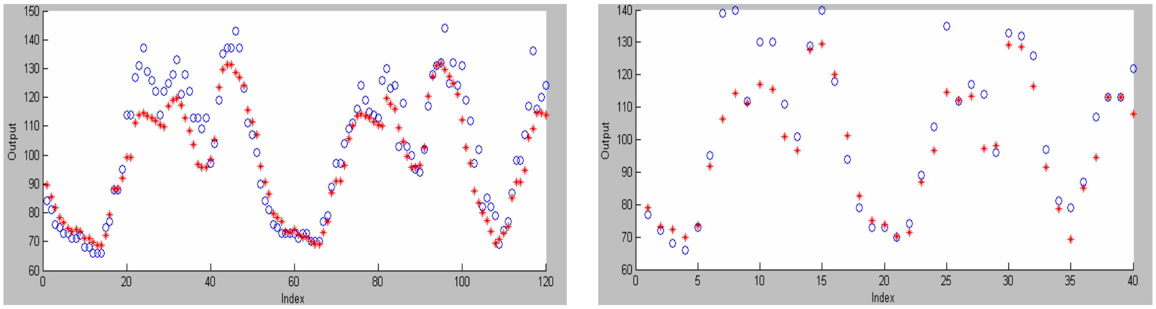
Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.9115	0.0787	0.1737	0.9095	0.0798	0.3005
July	0.9106	0.0667	0.1662	0.9085	0.0656	0.3045
Spring	0.7878	0.1297	0.2442	0.7793	0.1333	0.4294
Summer	0.9009	0.0723	0.1191	0.8956	0.0730	0.2168
Average	***0.8777***	***0.0869***	***0.1758***	***0.8732***	***0.0879***	***0.3128***

Table (5.6): Results Obtained from the MISO Models with Hybrid Optimization.

Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.9679	0.0448	0.1062	0.9595	0.0501	0.2035
July	0.9815	0.0257	0.0769	0.9719	0.0324	0.1715
Spring	0.9376	0.0761	0.1379	0.9367	0.0771	0.2399
Summer	0.9323	0.0561	0.0992	0.9290	0.0568	0.1804
Average	***0.9548***	***0.0507***	***0.1051***	***0.9493***	***0.0541***	***0.1988***

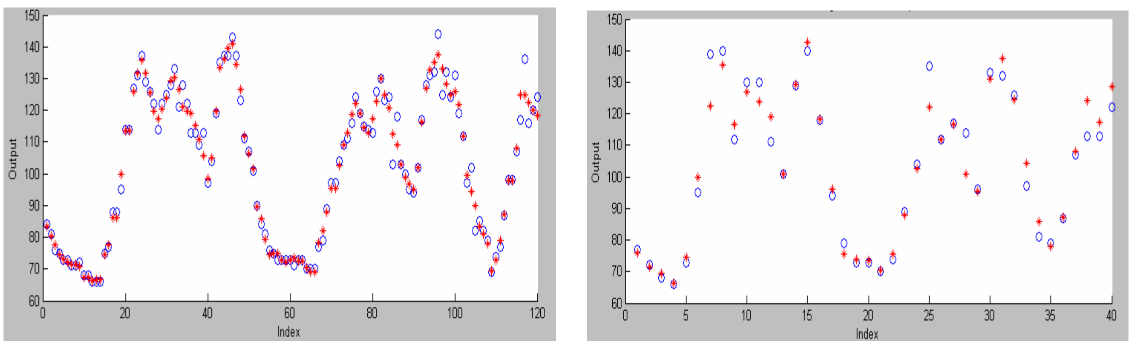
As Shown in Table (5.6), very good results obtained specially from the MISO models for a specific month such as May and July MISO models. An average CC of 0.9548 describes the agreement between the actual and predicted loads. In addition, small values for the two error measures (MAPE and RMSE) show the error using two different formulas. The results obtained from the MISO models (0.9548 average CC) are much better that the results obtained from the SISO models (0.8777 average CC). These results for SISO and MISO show the effect of the weather parameters in predicting the power loads as mentioned before.

Fig. (5.5) and (5.6), show a sample from the training and testing datasets which have been used to develop and test the adequacy of the developed July SISO and MISO Sugeno models with hybrid optimization. Part (a) represents a sample of 120 training data points. The circles represents the actual loads and the "*" sign represents the predicted loads. While part (b) shows the results obtained from a sample of 40 testing data points



(a): Training Dataset (b): Testing Datasets

Fig. (5.5): The Predicted Load Against the Actual Load for a Sample Training and Testing Datasets (120 Data Points Training and 40 Testing) From SISO July Model with Hybrid Optimization.



(a): Training Dataset (b): Testing Datasets

Fig. (5.6): The Predicted Load Against the Actual Load for a Sample Training and Testing Datasets (120 Data Points Training and 40 Testing) From MISO July Model with Hybrid Optimization.

5.3 SISO and MISO Sugeno Models with Back-Propagation Optimization Technique

As mentioned in chapter three, a neural network consists of a number of simple highly interconnected processing elements or nodes and is a computational algorithm that processes information by a dynamic response of its processing elements and their connections to external inputs. A back-propagation neural network consists of three or more layers, including an input, one or more hidden layers, and an output layer.

To develop a back-propagation SISO and MISO model, the collected training and testing datasets need to be implemented. The back-propagation learning algorithm employs a gradient- or steepest-descent heuristic that enables a network to self -organize in ways that improve its performance over time. The network first uses the input data set to produce its own output. This forward pass through the back-propagation network begins as the input layer receives the input data pattern and passes it to the hidden layer. Each processing element calculates an activation value by first summing the weighted inputs. This sum is then used by an activation function in each node to determine the activity level of that processing node.

The output generated by the network is compared with the known target value. If there is no difference, no learning takes place. If a difference exists, the resulting error term is propagated back through the network, using a gradient- or steepest-descent heuristic to minimize the error term by adjusting the connection weights.

In order to develop a SISO and MISO Sugeno models with back-propagation optimization, the same procedure that applied to the models with hybrid optimization has been followed. The same training and testing datasets have been used to develop the models, and so the same MFs types, numbers and parameters have been used.

Table (5.7) shows the results obtained for the training and testing datasets from monthly (May, July) and seasonal (Spring, Summer) SISO Sugeno models with back-propagation optimization. Table (5.8) shows the results obtained from the monthly and seasonal MISO Sugeno models with back propagation optimization.

Table (5.7): Results Obtained from the SISO Models with Back-Propagation Optimization.

Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.9115	0.0787	0.1737	0.9095	0.0798	0.3005
July	0.8340	0.0903	0.2219	0.8246	0.0893	0.4123
Spring	0.7462	0.1524	0.2639	0.7374	0.1549	0.4629
Summer	0.7995	0.0967	0.1648	0.7997	0.0964	0.2927
Average	***0.8228***	***0.1045***	***0.2061***	***0.8178***	***0.1051***	***0.3671***

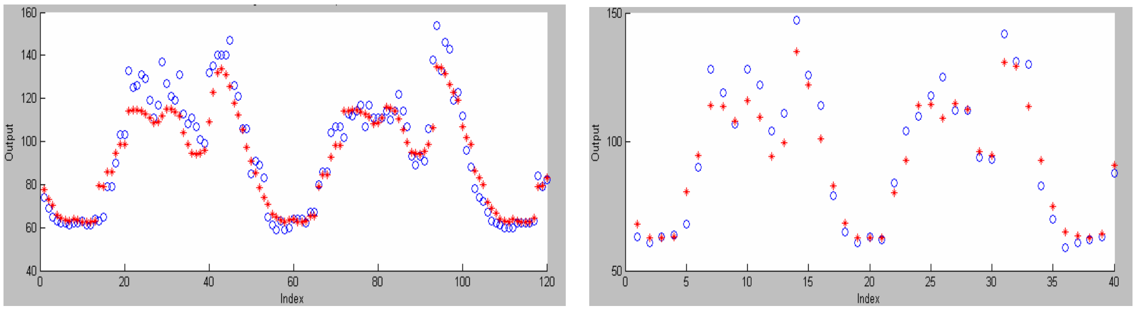
Table (5.8): Results Obtained from the MISO Models with Back-Propagation Optimization.

Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.8377	0.0991	0.2308	0.8121	0.1025	0.4217
July	0.4697	0.1166	0.3550	0.4919	0.1126	0.6345
Spring	0.2895	0.1818	0.3794	0.2930	0.1809	0.6552
Summer	0.5511	0.1256	0.2290	0.5504	0.1257	0.4069
Average	***0.5370***	***0.1308***	***0.2986***	***0.5369***	***0.1304***	***0.5296***

From the results that appear in Tables (5.7) and (5.8), the results obtained from the SISO models with back-propagation optimization are much better than the results obtained from the MISO models.

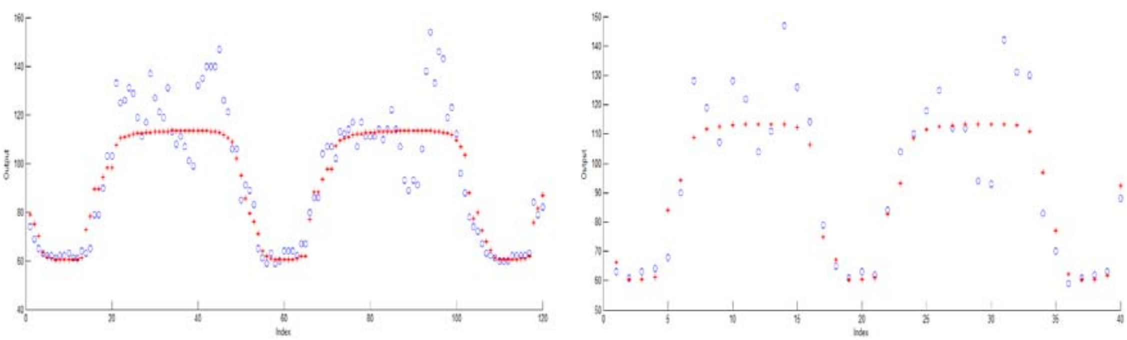
Fig. (5.7) and (5.8), represent a sample from the training and testing datasets which have been used to develop and test the adequacy of the developed May SISO and MISO Sugeno models with back-propagation optimization. Part (a) represents a sample of 120 training data points.

The circles represents the actual loads and the "*" sign represents the predicted loads. While part (b) shows the results obtained from a sample of 40 testing data points



(a): Training Dataset (b): Testing Datasets

Fig. (5.7): The Predicted Load Against the Actual Load for a Sample Training and Testing Datasets (120 Data Points Training and 40 Testing) From SISO May Model with Back-Propagation Optimization.



(a): Training Dataset (b): Testing Datasets

Fig. (5.8): The Predicted Load Against the Actual Load for a Sample Training and Testing Datasets (120 Data Points Training and 40 Testing) From MISO May Model with Back-Propagation Optimization.

As mentioned in the previous chapters, neural networks has some limitations as well which have prevented it from providing efficient solutions for a large class of nonlinear time variant problems. It is difficult to understand how the neural net actually learns the input-output relationships and maps that to its weights.

For example in (Arafeh et al, 1999) the back propagation-based neuro fuzzy model gave unexpected predictions. Furthermore, Chen in (Chen et al, 2004) reported that the results produced by neural networks are not satisfactory at all. However, these results indicate more about the incapability of the neural networks due to either the difficulty of choosing parameters, or the need for a large historical number of samples, or both.

Chen in (Chen et al, 2001) and Al-Shreef in (Al-Shareef et al., 2008) suggested a scaling scheme to all the inputs and the output to [-c,c] range, where c is a positive number. This scheme has been used to avoid the convergence problems and to enhance the output of the models with back-propagation, since the inputs and output variables have very different ranges. Furthermore, Satish in (Satish et al., 2004) mentiond that normalization of time and temperature data to the same range is an important stage for training the neural networks. The normalization of data not only facilitates the training process but also helps in shaping the activation function.

The input and output variables have been scaled as follows to enhance the models with back-propagation optimization (Al-Shareef et al., 2008)):

$$y_s = \frac{(y_{\max} - y_{\min})(x - x_{\min})}{x_{\max} - x_{\min}} + y_{\min} \quad , \tag{5.5}$$

Where y_s is the scaled data element, x is the original data element for each input and output vectors, x_{max} and x_{min} are the maximum and minimum corresponding data element respectively. The variables y_{max} and y_{min} represents the maximum and minimum scaling range.

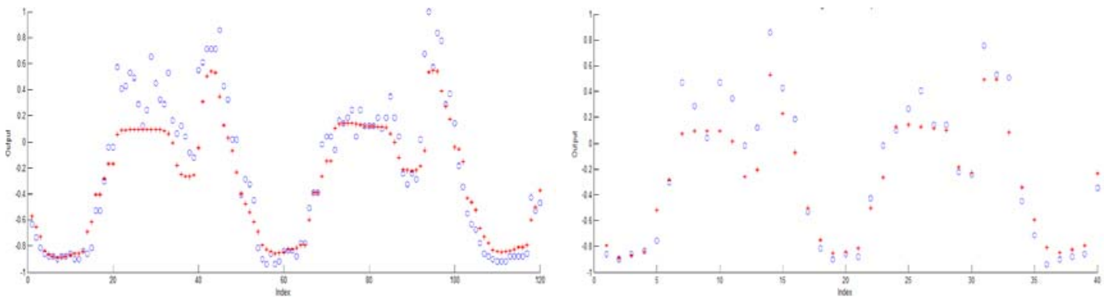
In our models that have been developed using back-propagation techniques a training stage has been repeated using the scaled inputs and output parameters to [-1,1] range. A rescaling stage required to convert the output to the original value.

Table (5.9) shows the results obtained for the training and testing datasets from monthly (May, July) and seasonal (Spring, Summer) MISO Sugeno models with back-propagation optimization after applying the scaled inputs to [-1,1] range in building the models.

Table (5.9): Results Obtained from the MISO Models with Back-Propagation Optimization Using Scaled Inputs/Output Variables.

Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.9188	0.0763	0.1668	0.9053	0.0775	0.3071
July	0.9129	0.0648	0.1642	0.9109	0.0633	0.3007
Spring	0.9073	0.0975	0.1667	0.9084	0.0953	0.2880
Summer	0.8986	0.0725	0.1204	0.8933	0.0733	0.2191
Average	***0.9094***	***0.0778***	***0.1545***	***0.9045***	***0.0774***	***0.2787***

As listed in Table (5.9), a noticeable enhancement in the average CC, MAPE, and RMSE measures shown from the obtain results. Scaling the inputs to the same range [-1,1] not only gave us better results, but it also speed up the convergence of the training process. Fig. (5.9) represents a sample from the scaled training and testing datasets which have been used to develop and test the adequacy of the developed May and MISO Sugeno models with back-propagation optimization and scaled training datasets. Part (a) represents a sample of 120 scaled training data points. The circles represents the actual loads and the "*" sign represents the predicted loads. While part (b), shows the results obtained from a sample of 40 scaled testing data points.



(a): Training Dataset (b): Testing Datasets

Fig. (5.9): The Predicted Load Against the Actual Load for a Sample Scaled Training and Testing Datasets (120 Data Points Training and 40 Testing) From MISO May Model with Back-Propagation Optimization.

Fig. (5.10) represents a comparison between the error convergence for developing a SISO Sugeno model with back-propagation optimization using normal and scaled datasets. The upper plot (a) represents the convergence when training the model using normal inputs values, and the plot in the bottom (b) represents the error convergence when using scaled inputs. The model with normal inputs and output values takes longer time to converge and it is completed the training after more than 35000 epochs. In the case of using scaled inputs the training of the model completed after 400 epochs only. Furthermore, the starting error for the model with scaled inputs is equal to (0.5), where it is equal to (100) for the model with normal inputs values.

(a)

(b)

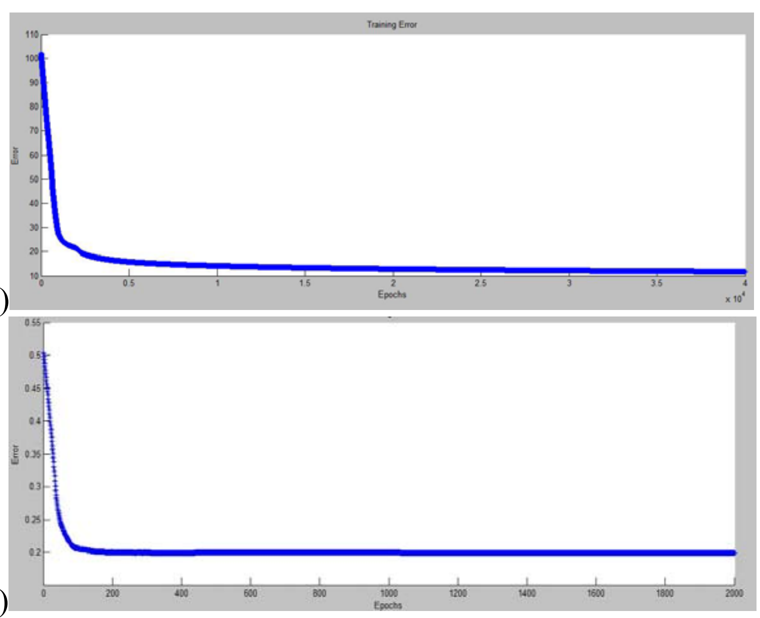


Fig. (5.10): The Error Convergence for Training a SISO July Models Using Normal and Scaled Inputs..

5.4 SISO and MISO Sugeno Models with Subtractive Clustering

As mentioned in chapter three, data clustering is concerned with the partitioning of a data set into several groups such that the similarity within a group is larger than that among groups. Clustering algorithms are used extensively not only to organize and categorize data, but are also useful for data compression and model construction. By finding similarities in data, one can represent similar data with fewer symbols for example. Also if we can find groups of data, we can build a model of the problem based on those groupings.

The purpose of clustering is to identify natural groupings of data from a large data set, such that a concise representation of system's behavior is produced. Chiu in (Chiu ,1994 and 1997) proposes a subtractive clustering method with improved computational effort in which data points themselves are considered as the candidates for cluster centers. By using this method, the computation is simply proportional to the number of data points and is independent of the dimension of the problem. According to Chiu in (Chiu, 1994) Subtractive clustering is a fast, one-pass algorithm for estimating the number of clusters and the cluster centers in a set of data.

The same steps that have been followed in developing the models in the previous sections applied here. The difference is in updating the radius of the cluster that gives us the best results.

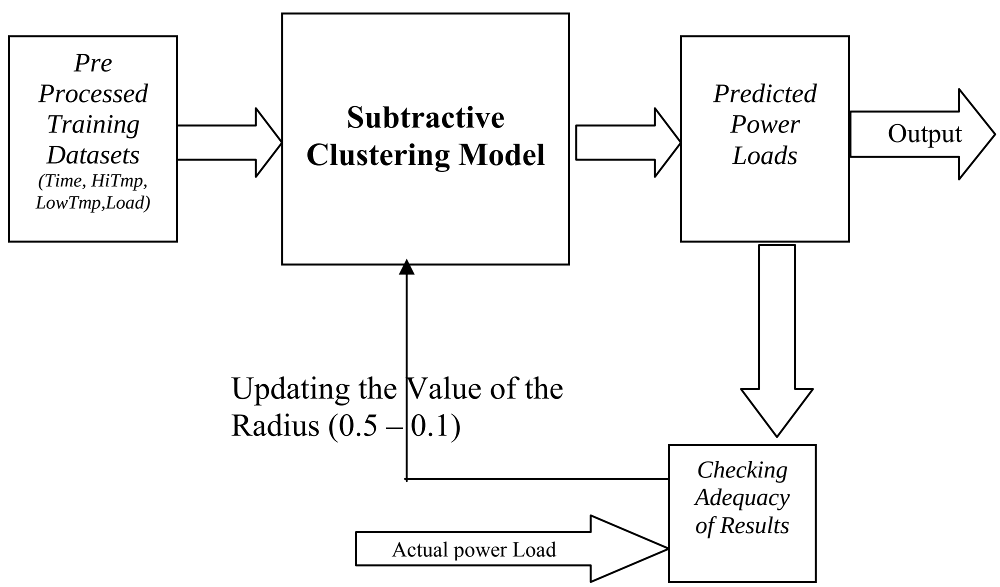


Fig. (5.11): A General Block Diagram for Developing/Training Subtractive Clustering Model

Fig. (5.11) illustrates a general developing "training" block diagram of a Subtractive model. In order to develop a suitable a model using the Subtractive clustering technique, we should determine the radius of the clustering that produces the best results instead of finding the best number and type of MFs as in a Sugeno model with Hybrid and Back-propagation optimization. An experiment with different radius ranging from 0.1 to 0.5 which is the default value used by Matlab fuzzy toolbox has been applied. It was noticed that increasing the value of the radius leads to decreasing the number of rules and to decrease the accuracy of the developed model.

Table (5.10) summarizes the results of the experiment which applied to choose the value of the clustering radius that give us the best results for a May MISO Sugeno model. It is noticed that the value of 0.1 for the radius produces the best results. In addition to the CC and the error measures, the table represents the number of rules that have been produced when changing the value of the cluster radius. It is noticed that the number of rules increased when the value of the radius decreased.

Table (5.10): Results Obtained from an Experiment to Choose the Best Value of the Clustering Radius for a May MISO Model.

Value of Radius	No. of Rules	Training Dataset Errors			Testing Dataset Errors		
		CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
0.10	313	0.9845	0.0293	0.0741	0.9541	0.0483	0.2165
0.20	61	0.9291	0.0689	0.1563	0.9215	0.0716	0.2807
0.30	24	0.8970	0.0838	0.1867	0.8953	0.0852	0.3220
0.50	7	0.8355	0.1069	0.2322	0.8395	0.1051	0.3927

Using the value of 0.1 for the radius, several SISO and MISO have been developed. Table (5.11) shows the results obtained for the training and testing datasets from monthly (May, July) and seasonal (Spring, Summer) SISO Sugeno models with Subtractive clustering technique. Table (5.12) shows the results obtained from the monthly and seasonal MISO Sugeno models with Subtractive clustering technique.

Table (5.11): Results Obtained from the SISO Models with Subtractive Clustering.

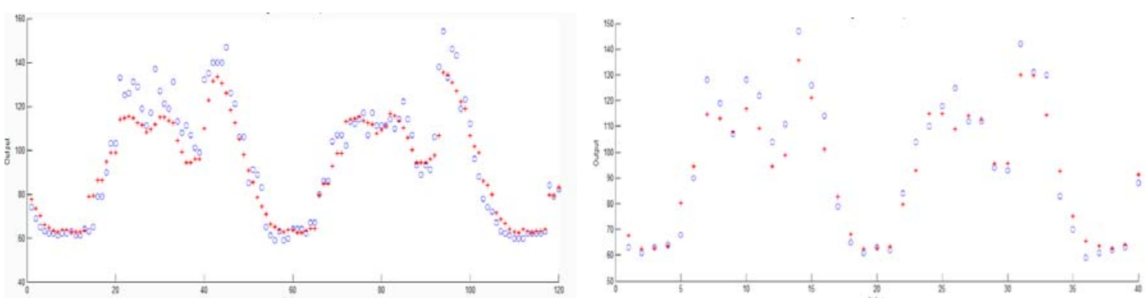
Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.9170	0.0756	0.1685	0.9140	0.0777	0.2933
July	0.9104	0.0669	0.1663	0.9083	0.0656	0.3049
Spring	0.7877	0.1297	0.2442	0.7789	0.1333	0.4298
Summer	0.9001	0.0725	0.1195	0.8959	0.0729	0.2165
Average	***0.8788***	***0.0862***	***0.1746***	***0.8743***	***0.0874***	***0.3111***

Table (5.12): Results Obtained from the MISO Models with Subtractive Clustering.

Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.9845	0.0293	0.0741	0.9541	0.0483	0.2165
July	0.9661	0.0368	0.1039	0.9563	0.0410	0.2131
Spring	0.9536	0.0639	0.1194	0.9486	0.0693	0.2169
Summer	0.9328	0.0549	0.0989	0.9335	0.0548	0.0986
Average	***0.9593***	***0.0462***	***0.0991***	***0.9481***	***0.0534***	***0.1863***

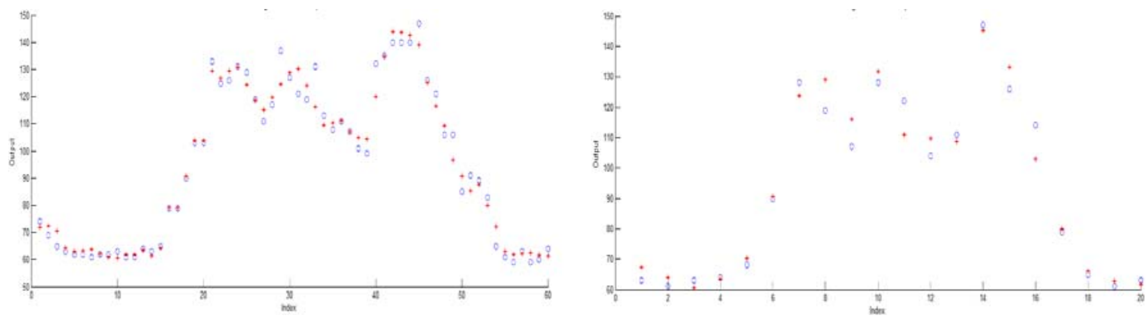
A clear difference between the results obtained from the SISO and MISO models. The average CC for the SISO models is equal to (0.8788) while it is equal to (0.9593) for the MISO models. The same thing noticed for the MAPE and RMSE measures. The average RMSE measures for the SISO models equal to (0.1746), while it is equal to (0.0991) for the MISO models. These results reflect the effect of the temperature parameters on the models outputs.

Fig. (5.12) and (5.13), represent the results obtained from a sample from the training and testing datasets which have been used to develop and test the adequacy of the developed May SISO and MISO Sugeno models with back-propagation optimization. Part (a) represents a sample of 120 training data points. The circles represents the actual loads and the "*" sign represents the predicted loads. While part (b) shows the results obtained from a sample of 40 testing data points



(a): Training Dataset (b): Testing Datasets

Fig. (5.12): The Predicted Load Against the Actual Load for a Sample Training and Testing Datasets (120 Data Points Training and 40 Testing) From SISO May Model with Subtractive Clustering.



(a): Training Dataset (b): Testing Datasets

Fig. (5.13): The Predicted Load Against the Actual Load for a Sample Training and Testing Datasets (120 Data Points Training and 40 Testing) From MISO May Model with Subtractive Clustering.

5.5 SISO and MISO Sugeno Cascaded Models with Subtractive Clustering and Hybrid Optimization

The purpose of the cascaded model is to achieve a more accurate model. According to Gross (Gross et al., 1987), it is estimated that in the British power system every 1% increase in the forecasting error is associated with an increase in operating costs of 10 million pounds per year. Two of the techniques that used before (Subtractive Clustering and Sugeno with Hybrid optimization) have been cascaded. The same steps that mentioned in the previous sections should be followed in building the cascaded model. At first a Sugeno model using Subtractive Clustering should be developed in the same way as mentioned in the previous section. After that the overall constructed model should be enhanced using the Hybrid optimization technique as shown in section (5.2).

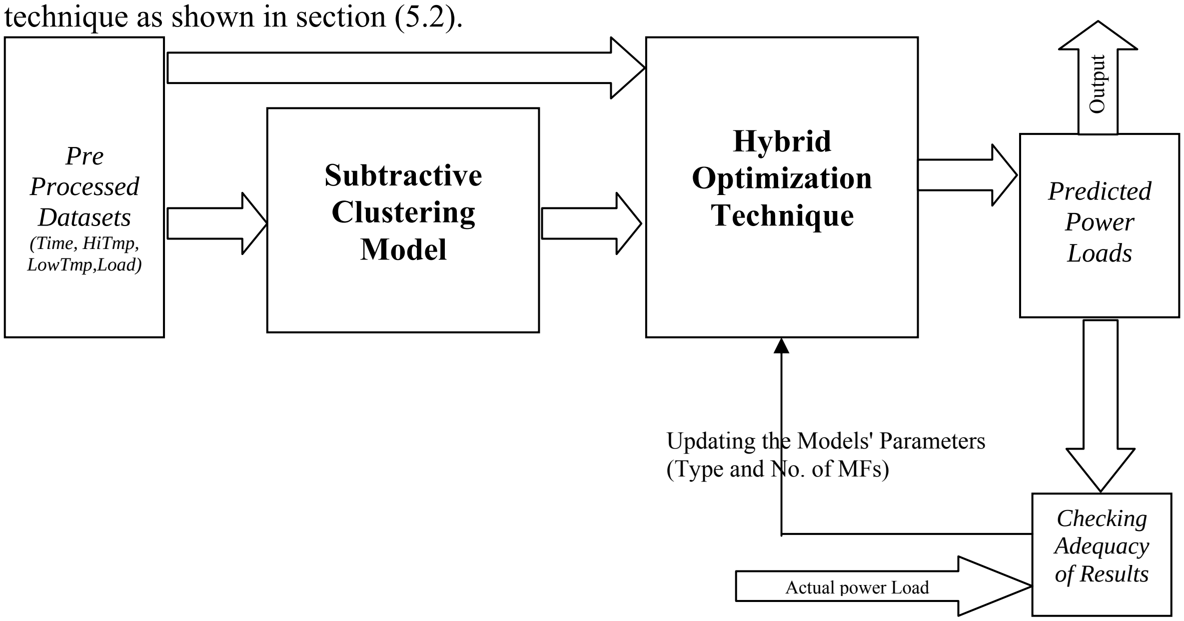


Fig. (5.14): A General Block Diagram for Developing/Training Cascaded (Subtractive Clustering with Hybrid Optimization) Model

Fig. (5.14) illustrates a general developing "training" block diagram of a cascaded (Subtractive Clustering with Hybrid optimization) model. As shown above in Fig. (5.14), developing a cascaded model consists of two main stages. At first, the same training datasets have been used to construct a Sugeno model using the Subtractive Clustering. Then, Hybrid optimization technique has been applied to fine tuning the constructed model parameters to achieve a more accurate model.

Table (5.13) shows the results obtained for the training and testing datasets from monthly (May, July) and seasonal (Spring, Summer) SISO cascaded models. Table (5.14) shows the results obtained from the monthly and seasonal MISO cascaded models.

Table (5.13): Results Obtained from the SISO Models with Subtractive Clustering and Hybrid Optimization.

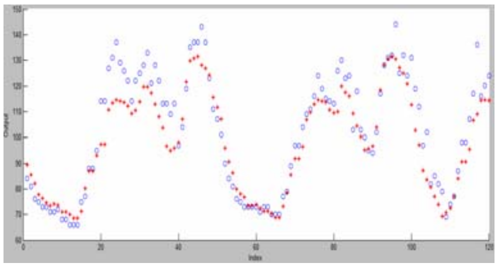
Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.9172	0.0755	0.1683	0.9135	0.0778	0.2940
July	0.9114	0.0666	0.1655	0.9057	0.0664	0.3090
Spring	0.7880	0.1296	0.2440	0.7790	0.1333	0.4297
Summer	0.9008	0.0724	0.1191	0.8956	0.0730	0.2169
Average	***0.8794***	***0.0860***	***0.1742***	***0.8735***	***0.0876***	***0.3124***

Table (5.14): Results Obtained from the MISO Models with Subtractive Clustering and Hybrid Optimization Technique.

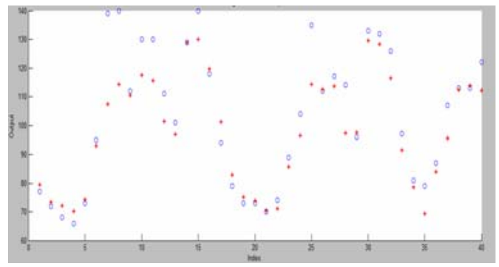
Month / Season	Training Dataset Errors			Testing Dataset Errors		
	CC	*MAPE*	*RMSE*	*CC*	*MAPE*	*RMSE*
May	0.9862	0.0275	0.0701	0.9609	0.0459	0.2002
July	0.9825	0.0259	0.0750	0.9701	0.0334	0.1768
Spring	0.9592	0.0603	0.1120	0.9529	0.0662	0.2077
Summer	0.9616	0.0414	0.0753	0.9624	0.0412	0.0747
Average	***0.9724***	***0.0388***	***0.0831***	***0.9616***	***0.0467***	***0.1649***

As we can see in the previous table, a noticeable enhancement to the models output accrued. The average CC for the MISO cascaded model equal to (0.9724), while it is equal to (0.9593) in the MISO Subtractive model. The same thing happened to the error measures MAPE and RMSE. This enhancement occurs because we have been firstly construct the model using the subtractive clustering and then the Hybrid optimization technique has been used to optimize the model parameters (two techniques have been used to build the model).

Fig. (5.15) and (5.16), represent the results obtained from the same training and testing datasets that have been used in the previous figures (Fig. (5.12) and Fig. (5.13)) a sample from the training and testing datasets which have been used to develop and test the adequacy of the developed May SISO and MISO cascaded models (Subtractive with Hybrid optimization). Part (a) represents a sample of 120 training data points. The circles represents the actual loads and the "*" sign represents the predicted loads. While part (b) shows the results obtained from a sample of 40 testing data points

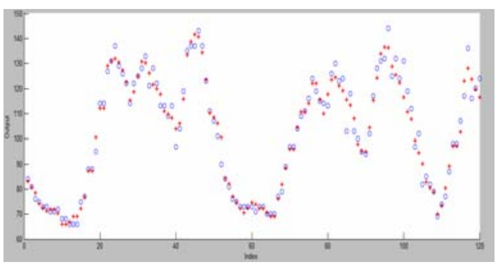


(a): Training Dataset

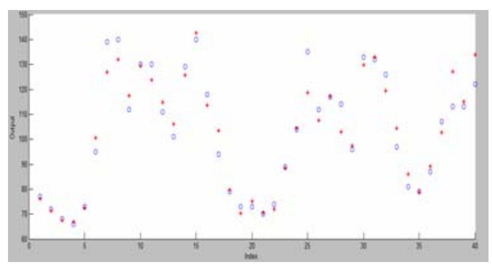


(b): Testing Datasets

Fig. (5.15): The Predicted Load Against the Actual Load for a Sample Training and Testing Datasets (120 Data Points Training and 40 Testing) From May SISO Cascaded Model.



(a): Training Dataset



(b): Testing Datasets

Fig. (5.16): The Predicted Load Against the Actual Load for a Sample Training and Testing Datasets (120 Data Points Training and 40 Testing) From May MISO Cascaded Model.

5.5 The Stand Alone Graphical User Interface (GUI): Electric Power Load Forecasting System (EPLFS)

As mentioned in the previous sections, different modeling techniques for the short term load forecasting problem have been explored. Different measures were used to check the adequacy of the developed models. These models were integrated within a stand alone application with GUI. The developed Electric Power Load Forecasting System "EPLFS" is as shown in Fig.(5.17); the figure demonstrates the forecasted power loads for a testing datasets. The three lists in the system present the times, forecasted loads, and the actual loads. Three different measures appear in the bottom of the right corner, the CC, MAPE, and RMSE.

Using EPLFS we can load the datasets, evaluate the predicted output using the developed models, plot the actual and predicted load, and calculate several measures including the CC, MAPE and RMSE.

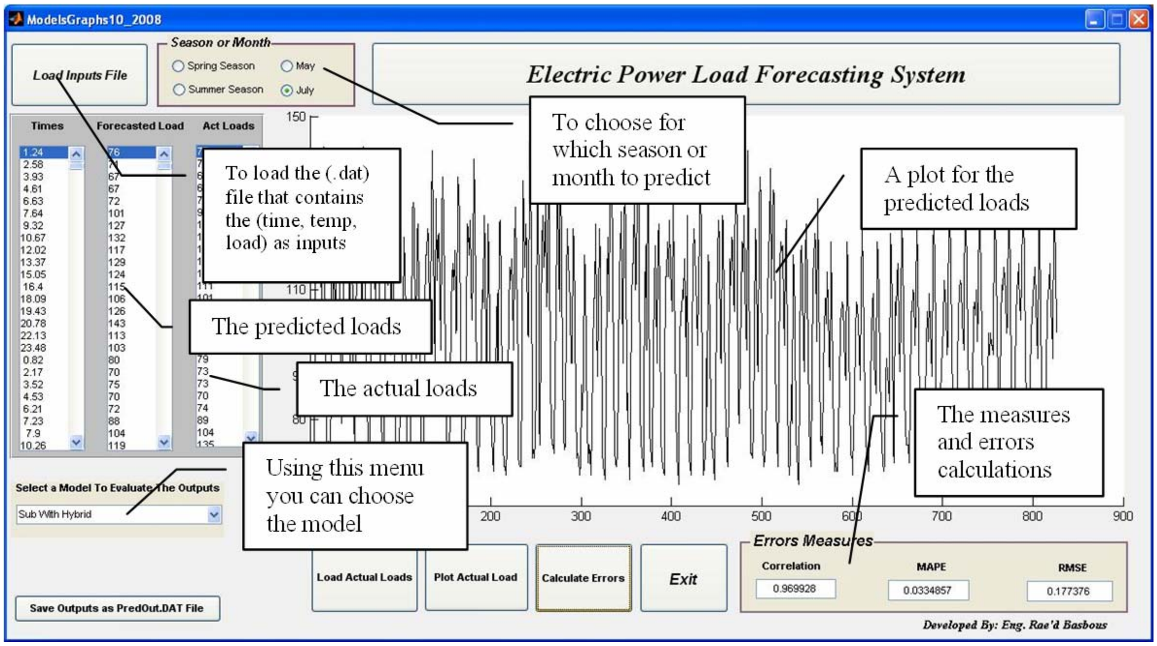


Fig. (5.17): The EPLF System: Showing a Plot for the Predicted Loads in a Certain Hours.

Fig. (5.18) represents a snapshot for the EPLFS when used to predict the power loads for certain times with temperatures parameters. A plot for the actual and predicted power loads are seen in the snapshot. The blue dotted line represents the actual loads and the black continuous one represents the predicted power loads.

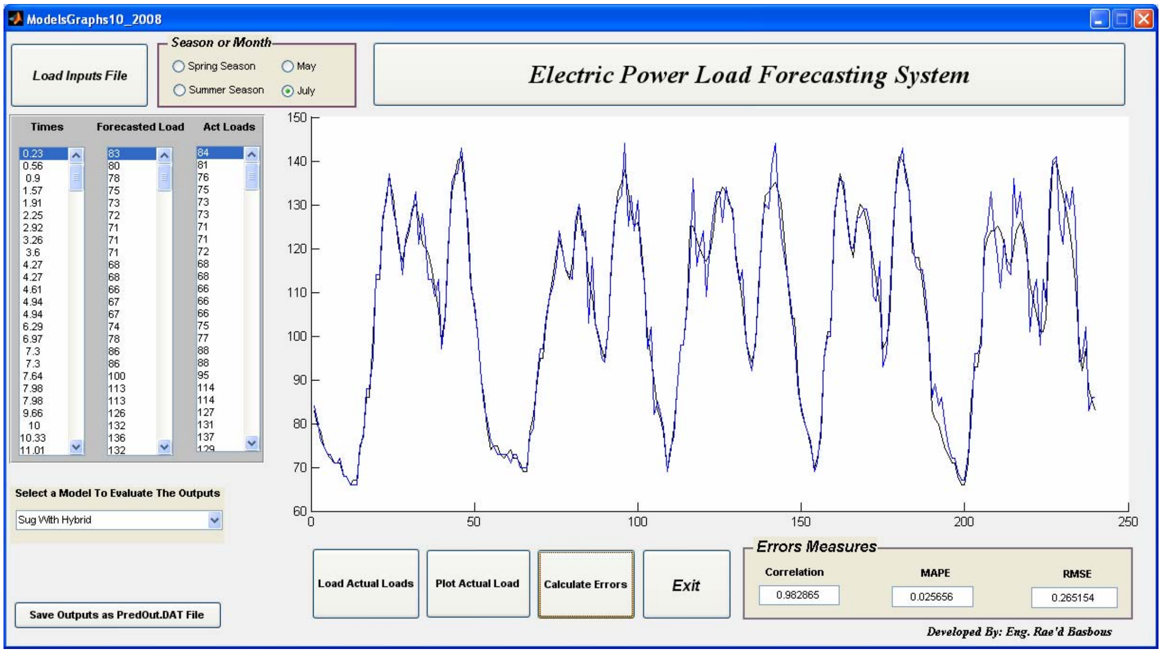
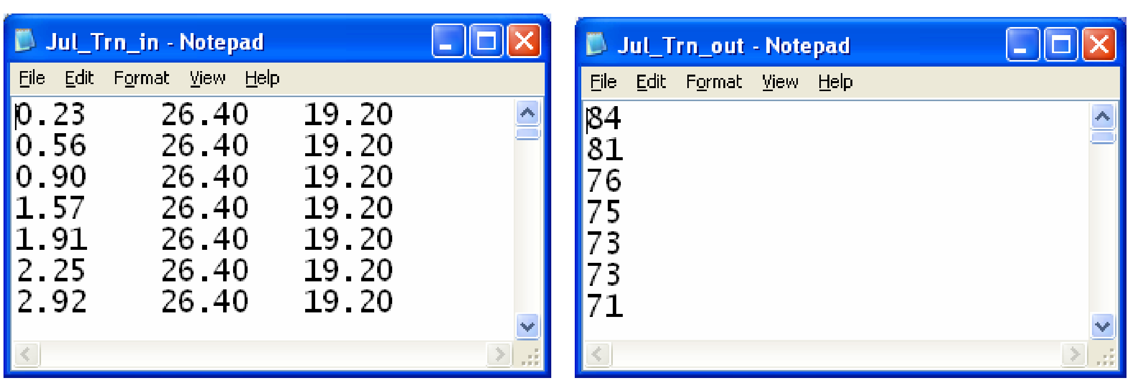


Fig. (5.18): The EPLF System: Showing a Plot for the Predicted Loads vs. Actual Ones.

To load an input and actual load files to the EPLFS, it should be in (.DAT) format. For the input file it should be with three columns. The first column representing the times formatted as mentioned in the previous chapter. The second and third columns representing the temperature parameters (High and Low) respectively. For the file that containing the actual loads to be compared with the predicted ones, the loads should be in one column.

Fig. (5.19), represents the format of the input parameters and actual loads for a sample of the available datasets.



(a) (b)

Fig. (5.19): (a) The Input Parameters File, (b)The Corresponding Actual Loads File for the Input / Output Parameters File.

5.6 Testing the EPLFS Using Unseen Datasets from the Year 2008

As mentioned in chapter four, new historical profiles have been obtained from JDECO and PMO for the year 2008. The selected datasets (one day and one week from July and May) have been used to test the EPLFS for new unseen datasets. These datasets have not been considered in the cross validation process that has been applied in developing our models. The first day of May and July (01/05, 01/07) has been chosen in order to use the system to predict the load for one day. To test the system for predicting the load for a period of one week the first week of May and July (01-07/05, 01-07/07) has been considered.

The average CC obtained for the developed models in case of one day prediction is equal to 0.9412, while it is equal to 0.9327 in the case of one week prediction. These results indicate how accurate the models in predicting the power loads for new unseen datasets are.

Table (5.15) and Table (5.16) show the average correlation and error measures for one day and one week power load prediction using the developed models.

Table (5.15): The Average CC and Error Measures for One Day Prediction

Model	*CC*	*MAPE*	*RMSE*
July	0.9680	0.0407	0.2242
Summer	0.9435	0.0506	0.3187
May	0.9794	0.0350	0.1902
Spring	0.8738	0.1078	0.4874
Average	***0.9412***	***0.0585***	***0.3051***

Table (5.16): The Average CC and Error Measures for One Week Prediction

Model	*CC*	*MAPE*	*RMSE*
July	0.9428	0.0533	0.2872
Summer	0.9255	0.0576	0.3513
May	0.9578	0.0464	0.2850
Spring	0.9047	0.0809	0.4308
Average	***0.9327***	***0.0595***	***0.3385***

Fig. (5.20) and Fig. (5.21) below show the results obtained from the EPLFS for one day and one week prediction using the developed models.

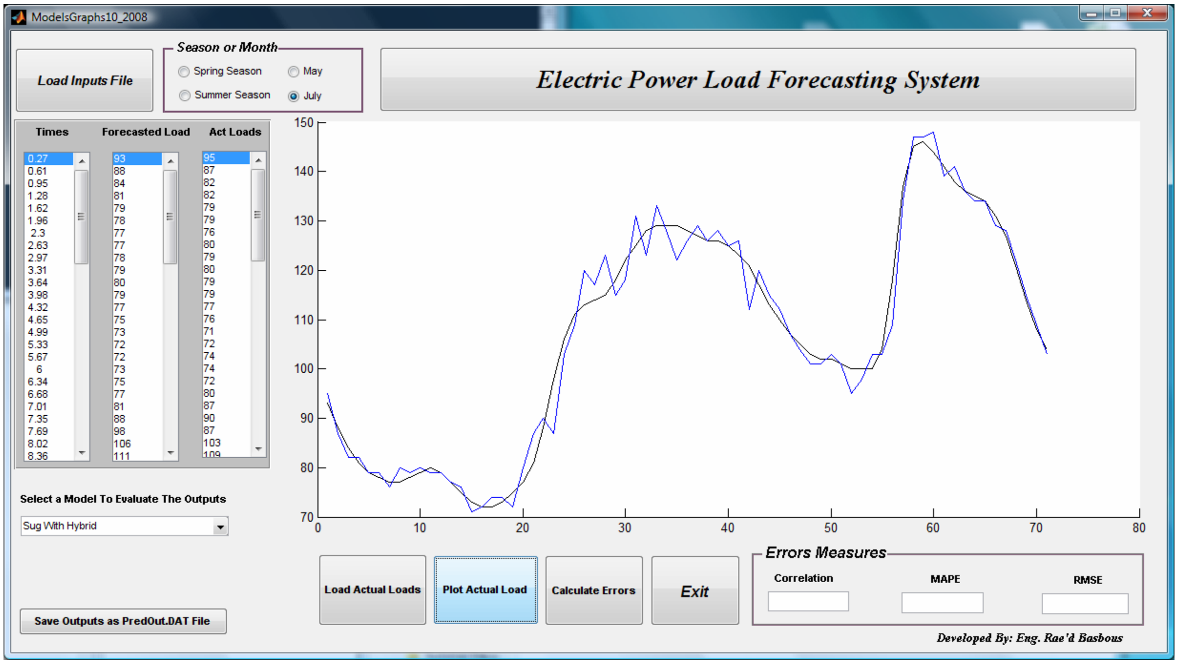


Fig. (5.20): One Day Prediction using the EPLFS for New Unseen Datasets.

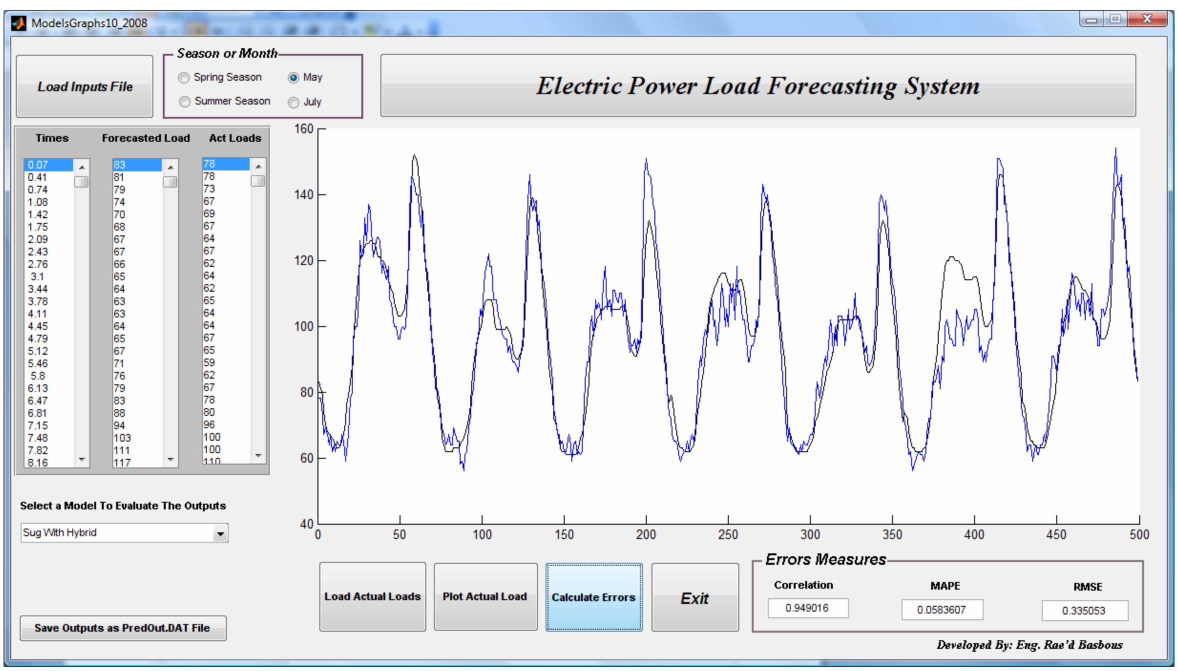


Fig. (5.21): One Week Prediction using the EPLFS for New Unseen Datasets.

Chapter Six:

Discussions, Results and Comparisons

6.1 Introduction

As mentioned in the previous chapter, several models have been developed based on FIS using several optimization techniques such as hybrid and back propagation to solve the short term load forecasting for a chosen power line in Beir Nabala village at Jerusalem district. Furthermore, Sugeno models using Subtractive clustering have been developed, and finally a cascaded-type Sugeno models using Subtractive clustering with Hybrid optimization have been developed to enhance the results obtained from the models constructed using Subtractive clustering.

Different measures have been used to check the adequacy of the developed models. These measures including: the Correlation Coefficient (CC), MAPE and RMSE.

Two kinds of models have been developed, SISO models, and MISO models. For the SISO models the time has been used as the input for the models and the power load at that time has been used as the output. In the MISO models, three variables (time, High Temperature, and the Low Temperature for that day) have been used as an input for the developed models, and the power load at that time was considered as the model output.

The results that have been obtained from the MISO models are better than those obtained for SISO models using the same trained data (time input), similar number of Memberships Functions (MF), and similar number of rules. Further more, the predicted loads for the months of May and July are nearer the actual loads than those of other months.

A detailed comparisons and discussions will be illustrated in the next sections.

6.2 Results and Comparisons between the Developed Models

In this thesis and using the available historical datasets profiles presented in Table (4.1) in chapter four, we have started by developing eight Sugeno models with hybrid optimization technique. Four of the models are SISO models and the other four are MISO models. These models have been used to predict the power load in specific months (May, July) or general model to be used in seasons (Spring, Summer). Another eight models have been developed using the same datasets but with the back-propagation optimization technique. After that the Subtractive clustering has been used to construct a Sugeno models for the same months and seasons. Then, the Subtractive clustering with the Hybrid optimization technique have been used to construct a cascaded model to improve and enhance the results obtained from the previous models.

As mentioned in chapter five, the training data sets and the models parameters (number and type of MF, number of rules, and cluster radius) have been fixed and used for the proposed models. For example 12 MF for the Time input and 7 MF for the Low and High temperatures of the type G-Bell have been fixed for all the models with Hybrid and Back-propagation optimization techniques. Furthermore, for the models that have been constructed using the Subtractive Clustering, a cluster radius of the value 0.1 has been fixed and used to construct these models.

Fig.(6.1) show the CC measures for the results obtained from the developed SISO models. The figure shows the CC for the training and testing datasets. It is clear from the figure that the best results obtained from the models that have been developed for specific month (May, July) using the Subtractive clustering with Hybrid optimization technique. These high CC values that have been obtained refer to the smooth variations in the consumed power loads in May and July compared to the variations of the consumed power loads in the seasons. The average CC for the developed May and July SISO models equal to 0.90 and 0.88 respectively, while the average CC for the developed Spring and Summer models equal to 0.76 and 0.86 respectively.

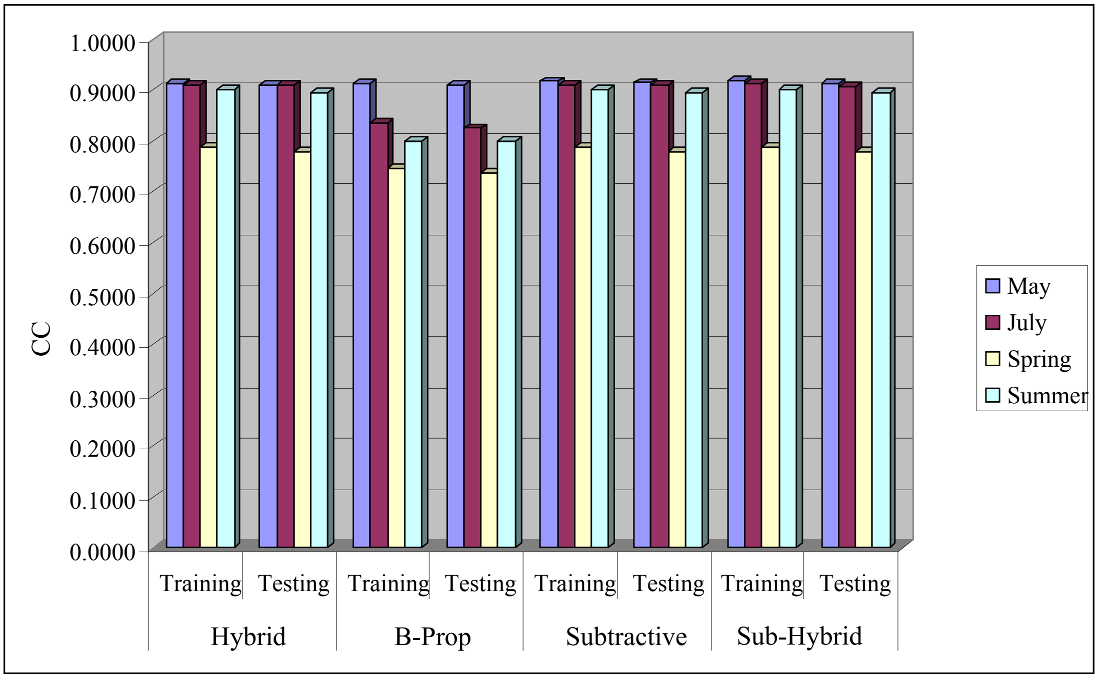


Fig. (6.1): The Correlations Measures for the Results Obtained from SISO Models

Table (6.1) below lists the correlation measures for the results obtained from the developed SISO models. The results from the above figure can be noticed in the table.

Table (6.1): The Correlations Measures for the Developed SISO Models

	Hybrid		Back-Prop		Subtractive		Sub-Hybrid	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
May	0.9115	0.9095	0.9115	0.9095	0.9170	0.9140	0.9172	0.9135
July	0.9106	0.9085	0.8340	0.8246	0.9104	0.9083	0.9114	0.9057
Spring	0.7878	0.7793	0.7462	0.7374	0.7877	0.7789	0.7880	0.7790
Summer	0.9009	0.8956	0.7995	0.7997	0.9001	0.8959	0.9008	0.8956

Fig.(6.2) shows the CC measures for the results obtained from the developed MISO models for the training and the testing datasets. From Fig. (6.2) and Table (6.2) below, it is clear that the results obtained from the SISO models have been improved which reflect the effect of the temperatures parameters on the power load. We can notice also that the Subtractive-based models have scored the highest average CC values which range between (0.93 and 0.95). These CC values between actual (testing data) and the predicted ones have been furtherly improved by cascading the Subtractive Clustering and Hybrid optimization technique. The new average CC values range between (0.95 and 0.97) with an average value of 0.96.

To have a solid conclusion, the other two measures have been used. We can notice that when the CC value increased the value of MAPE and RMSE decreased. As shown in Table (6.6) and

Table (6.7) below the average MAPE value of 0.05 for the developed Subtractive-based models has been furtherlly reduced to 0.04 by cascading the Subtractive clustering and Hybrid optimization technique, while the average RMSE value has been reduced from 0.18 to 0.16

The same as mentioned for the results obtained from the developed SISO models, the highest CC results have been obtained from the models that have been developed for the month of May and July using the Subtractive clustering and Subtractive Clustering with Hybrid optimization technique (cascaded model). These high CC values that have been obtained refer to the smooth variations in the temperatures parameters (Low and High temperature) which reflect a smooth variation in the consumed power loads. The average CC for the developed May and July MISO models equal to 0.94 and 0.95 respectively, while the average CC for the developed Spring and Summer models equal to 0.93 and 0.92 respectively. This suggests that the Subtractive Clustering-based models showing a promising solution with high performance to solve the STLF problem.

Furthermore, it is noticed that the cascaded model obtained the highest CC results with an average value of 0.96. The reason of obtaining such high CC from the cascaded models is from applying two optimization techniques (Subtractive Clustering and Hybrid optimization) in building the models. The Subtractive Clustering build the model by finding similarities in the datasets with the minimum number of rules, while the Hybrid optimization technique combines the least-squares estimator and the gradient descent method to optimize the developed model. Also, the models that have been developed for the month of July using Hybrid optimization technique performs very good with correlation equal to (0.9815). This is because of the smoothness in the variations to the weather parameters in July compared with the other months as listed in Table (4.2) in chapter four.

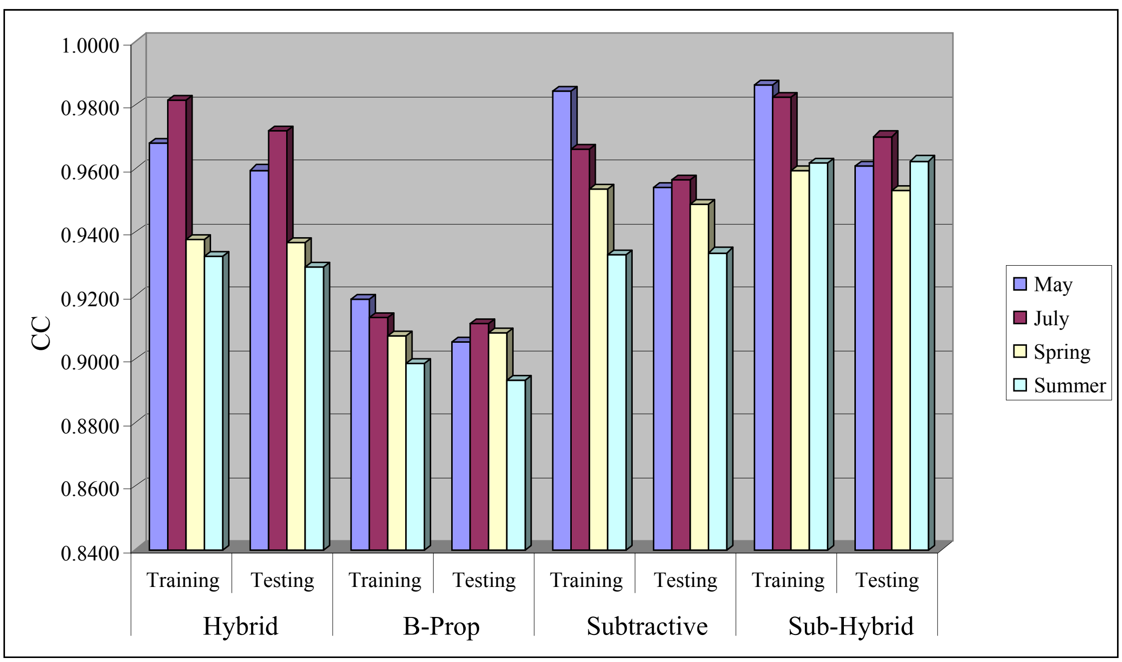


Fig. (6.2): The Correlations Measures for the Results Obtained from MISO Models

Table (6.2) below lists the CC measures for the results obtained from the developed MISO models. The results are for the training and testing datasets.

Table (6.2): The Correlations Measures for the Developed MISO Models

	Hybrid		Back-Prop		Subtractive		Sub-Hybrid	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
May	0.9679	0.9595	0.9188	0.9053	0.9845	0.9541	0.9862	0.9609
July	0.9815	0.9719	0.9129	0.9109	0.9661	0.9563	0.9825	0.9701
Spring	0.9376	0.9367	0.9073	0.9084	0.9536	0.9486	0.9592	0.9529
Summer	0.9323	0.9290	0.8986	0.8933	0.9328	0.9335	0.9616	0.9624

In order to get a general look for the correlation measures for the developed SISO models versus the developed MISO models, Fig. (6.3) shows the SISO models CC measures against the MISO models correlation measures for the results that have been obtained from these models. From Fig. (6.3), it is clear that using the temperature parameters leads to an improvement in the models outcome.

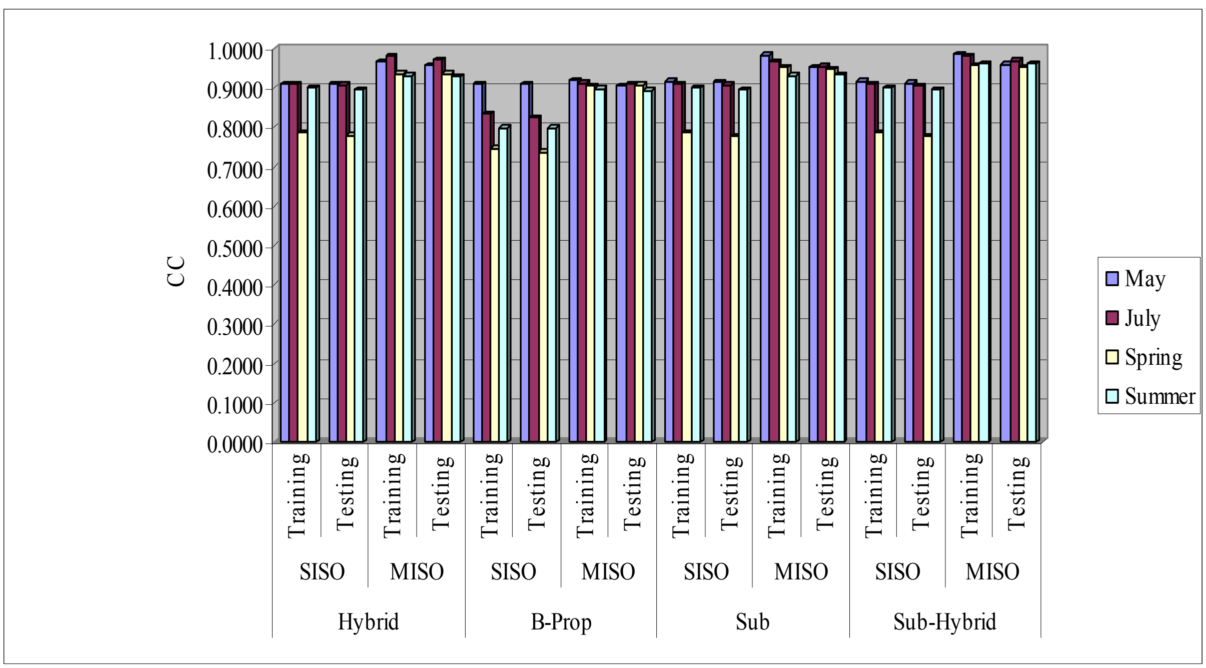


Fig. (6.3): The Correlations Measures for the Results Obtained from SISO vs. MISO Models

Table (6.3) below summarizes the CC measures for the results obtained from the SISO and MISO models that have been developed using the two optimization techniques the Hybrid and Back-propagation. The Hybrid optimization technique obtained results are better than the results that have been obtained using the back-propagation with average CC equal to (0.9548) for the MISO models with hybrid optimization compared to average CC equal to (0.9094) for the MISO models with back-propagation optimization. As mentioned before this is because of that the Hybrid optimization technique combines the least-squares estimator and the gradient descent method to build the model.

The same thing is noticed for the SISO models that have been developed using the same optimization techniques. Fig. (6.4) represents the results listed in Table (6.3) using a columns chart. As mentioned in the previous chapter, neural networks has some limitations as well which have prevented it from providing efficient solutions for a large class of nonlinear time variant problems. For example Arafeh in (Arafeh et al., 1999) and Chen in (Chen et al., 2004) reported that the back propagation-based neuro fuzzy model gave unexpected predictions and that the results produced by neural networks are not satisfactory at all.

Table (6.3): The Correlations Measures for the Models Developed using Hybrid and Back-propagation Optimization Techniques

	Sugeno with Hybrid Optimization				Sugeno with Back-Prop Optimization			
	SISO		MISO		SISO		MISO	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
May	0.9115	0.9095	0.9679	0.9595	0.9115	0.9095	0.9188	0.9053
July	0.9106	0.9085	0.9815	0.9719	0.8340	0.8246	0.9129	0.9109
Spring	0.7878	0.7793	0.9376	0.9367	0.7462	0.7374	0.9073	0.9084
Summer	0.9009	0.8956	0.9323	0.9290	0.7995	0.7997	0.8986	0.8933
Average	***0.8777***	***0.8732***	***0.9548***	***0.9493***	***0.8228***	***0.8178***	***0.9094***	***0.9045***

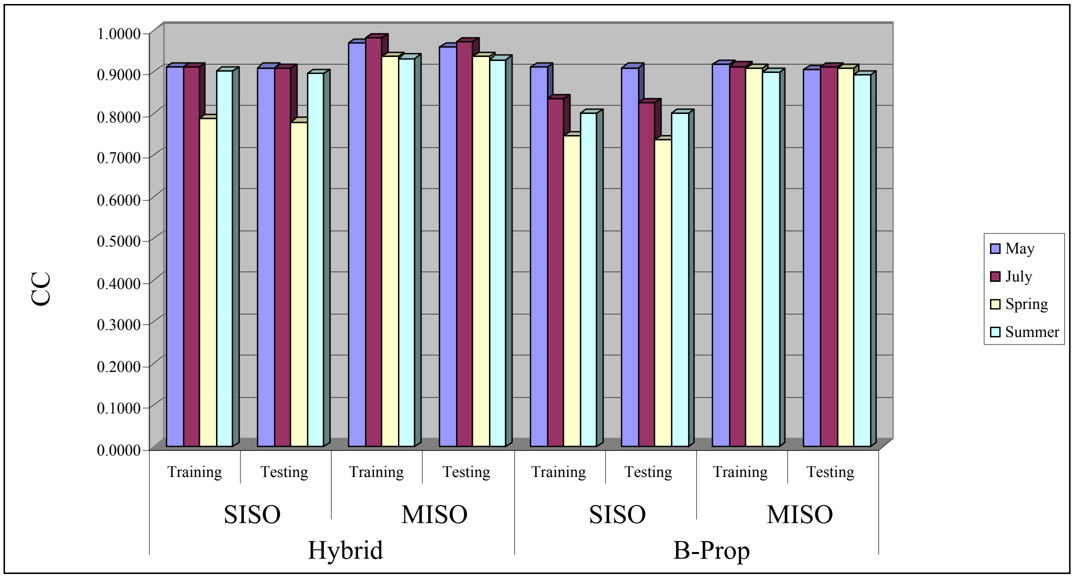


Fig. (6.4): The Correlations Measures Chart for SISO and MISO models with Hybrid vs. Back-propagation Optimization Techniques

Another comparison in Table (6.4) below summarizes the CC measures for the results obtained from the SISO and MISO models that have been developed using the Subtractive clustering and the cascaded method (Subtractive clustering with Hybrid optimization). As mentioned before the cascaded models show an enhancement to the predicted load because of using two optimization techniques in building this kind of models. The average CC for the models that have been developed using the Subtractive clustering only is equal to (0.9593) and for the cascaded models is equal to (0.9724) which is better than the average CC obtained from the models constructed using the Subtractive clustering. Fig. (6.5) represents the results listed in Table (6.4) using a columns chart.

Table (6.4): The Correlations Measures for the Models Developed using Subtractive clustering and Subtractive clustering with Hybrid optimization

	Sugeno Using Subtractive Clustering				Sugeno Using Sub-Clust with Hybrid			
	SISO		MISO		SISO		MISO	
	Training	*Testing*	*Training*	*Testing*	*Training*	*Testing*	*Training*	*Testing*
May	0.9170	0.9140	0.9845	0.9541	0.9172	0.9135	0.9862	0.9609
July	0.9104	0.9083	0.9661	0.9563	0.9114	0.9057	0.9825	0.9701
Spring	0.7877	0.7789	0.9536	0.9486	0.7880	0.7790	0.9592	0.9529
Summer	0.9001	0.8959	0.9328	0.9335	0.9008	0.8956	0.9616	0.9624
Average	***0.8788***	***0.8743***	***0.9593***	***0.9481***	***0.8794***	***0.8735***	***0.9724***	***0.9616***

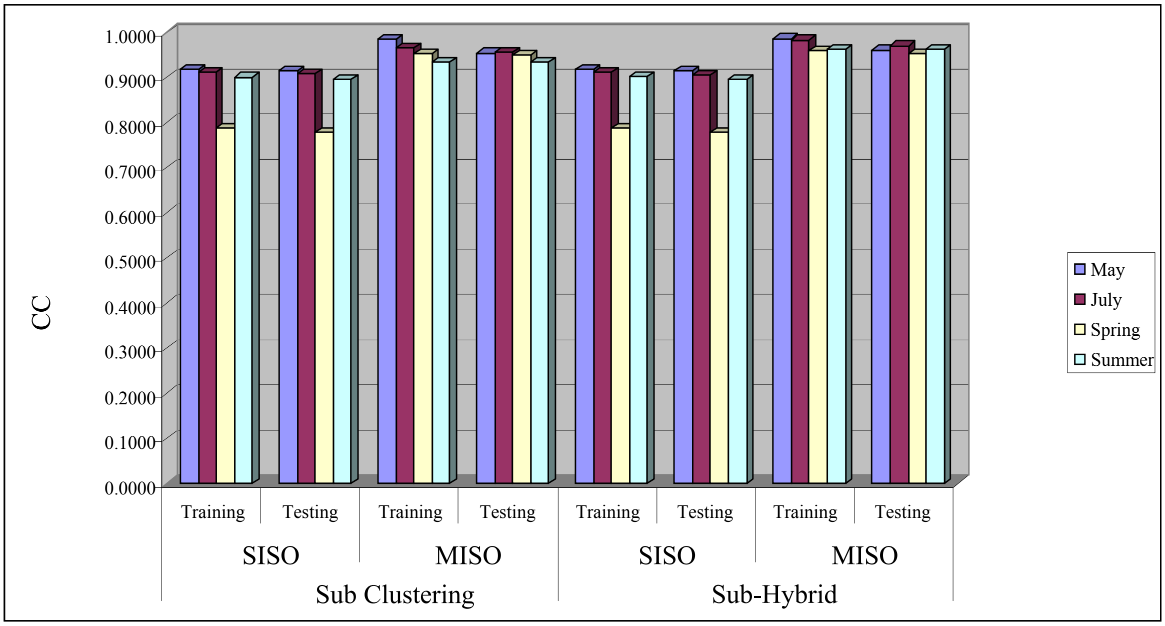


Fig. (6.5): The Correlations Measures Chart for SISO and MISO models Developed using Subtractive clustering vs. Cascaded (Subtractive with Hybrid)

Table (6.5) below lists the average CC measures for all the developed models using Hybrid, Back-propagation, Subtractive Clustering and Subtractive clustering with Hybrid optimization (cascaded model). In addition, Fig. (6.6) represent these averages using a chart graph. It is clear from the table and the figure that the cascaded model has produced the best results with an average equal to (0.97238) for the MISO models followed by the models that have been developed using the subtractive clustering with an average correlation equal to (0.95925). The results of the cascaded models have been obtained using less number of rules compared with the models that have been developed using the Hybrid and Back-propagation optimization techniques.

From Table (5.10) in chapter five, we can see that he cascaded model with cluster radius equal to 0.1 produced 313 rules, while with Hybrid and Back-propagation optimization techniques the number of rules that have been produced equal to the multiplication of inputs membership functions (12 MFs for the time * 7 MFs for the Hi-Temp * 7 MFs for the Low-Temp) which is equal to 588 rules.

Table (6.5): The Correlations Measures Average for all the Developed Models

Optimization Method	SISO		MISO	
	Training	*Testing*	*Training*	*Testing*
Sugeno With Hybrid Optimization	0.87770	0.87323	0.95483	0.94928
Sugeno With B-Prop Optimization	0.82280	0.81780	0.90940	0.90448
Sugeno Using Subtractive	0.87880	0.87428	0.95925	0.94813
Sugeno Using Subtractive with Hybrid	0.87935	0.87345	0.97238	0.96158

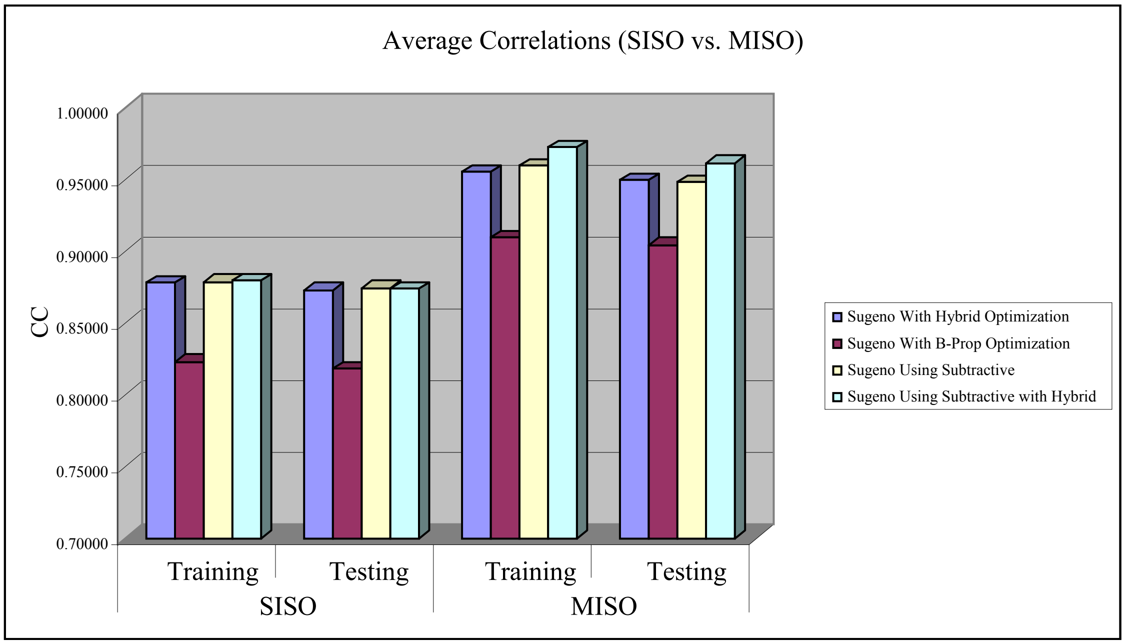


Fig. (6.6): The Average Correlations Measures Chart for all the SISO and MISO models

As mentioned in chapter five two different error measures, the MAPE and the RMSE have been used to examine and show the adequacy of the developed models and its outcome. The CC measures the agreements between the actual and predicted power loads, while the error measures RMSE and MAPE give an indication how the performance of the developed models are.

Fig. (6.7) below represents a summery chart graph for the MAPE values calculated for the results obtained from all the developed models (SISO and MISO) with different optimization techniques. As shown in the figure below the MAPE values for the cascaded model is the lowest ones and this reflect the highest CC that achieved from these models as shown in the tables listed above. You can notice that the MISO models have the lowest MAPE values over the SISO models because of the temperature parameters effect on the power load. The same thing has been noticed in the CC measures since the MISO models produced the best results and have the highest CC values.

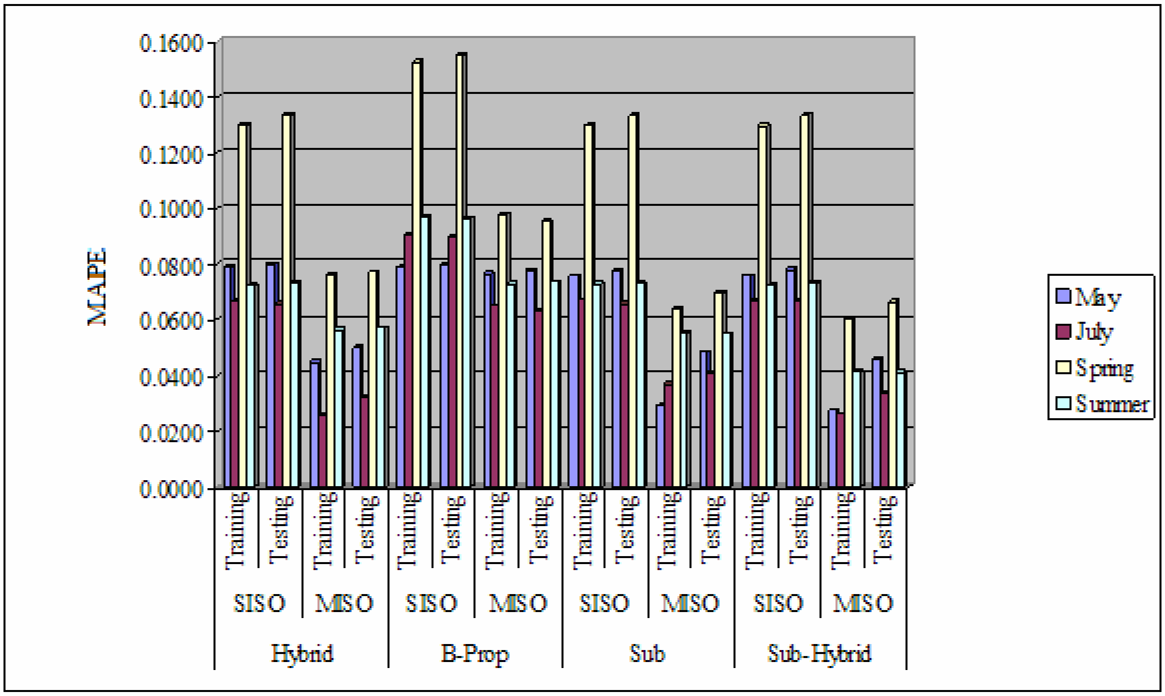


Fig. (6.7): The MAPE Measures Chart for all the SISO and MISO models

Table (6.6) shows the average results of the error measure MAPE for all the models. These results are shown graphically in Fig. (6.8). It is clear from the table and the figure that the lowest values are for the MISO models developed using the cascaded models with an average of the MAPE equal to (0.03).

Table (6.6): The Average MAPE Measures for all the Developed Models

Optimization Method	SISO		MISO	
	Training	*Testing*	*Training*	*Testing*
Sugeno With Hybrid Optimization	0.08685	0.08793	0.05068	0.05410
Sugeno With B-Prop Optimization	0.10453	0.10510	0.07778	0.07735
Sugeno Using Subtractive	0.08618	0.08738	0.04623	0.05335
Sugeno Using Subtractive with Hybrid	0.08603	0.08763	0.03878	0.04668

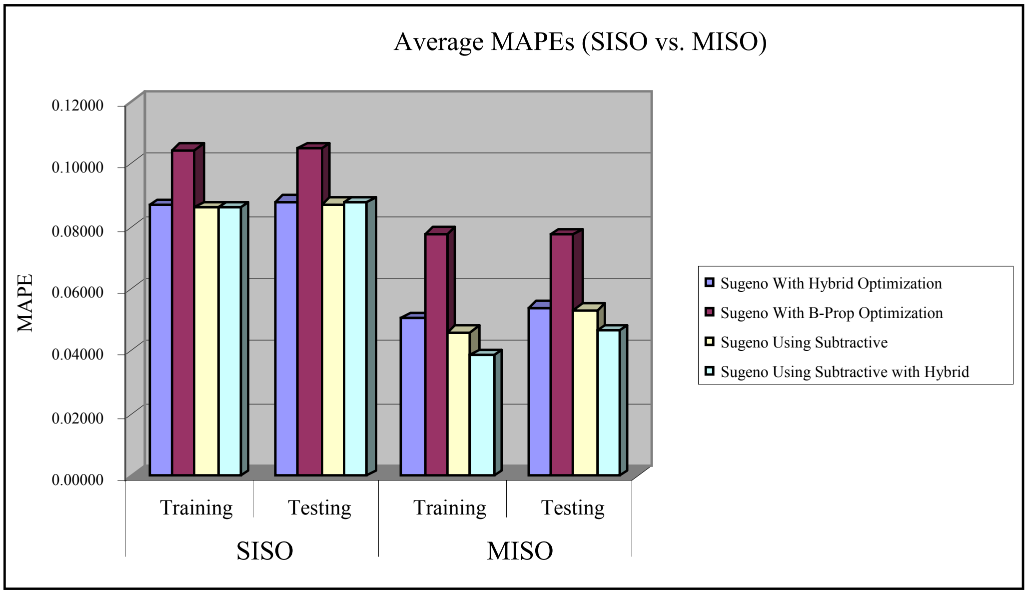


Fig. (6.8): The Average MAPE Measures Chart for all the SISO and MISO models

The other error measure values (RMSE) shown graphically in Fig. (6.9). This measure show the adequacy of the developed models too in addition to the MAPE measures. The same thing for the RMSE results as in the MAPE results achieved where the cascaded model have the best results (the lowest RMSE values). The developed MISO models with the temperature parameters produce the lowest RMSE values over the SISO models.

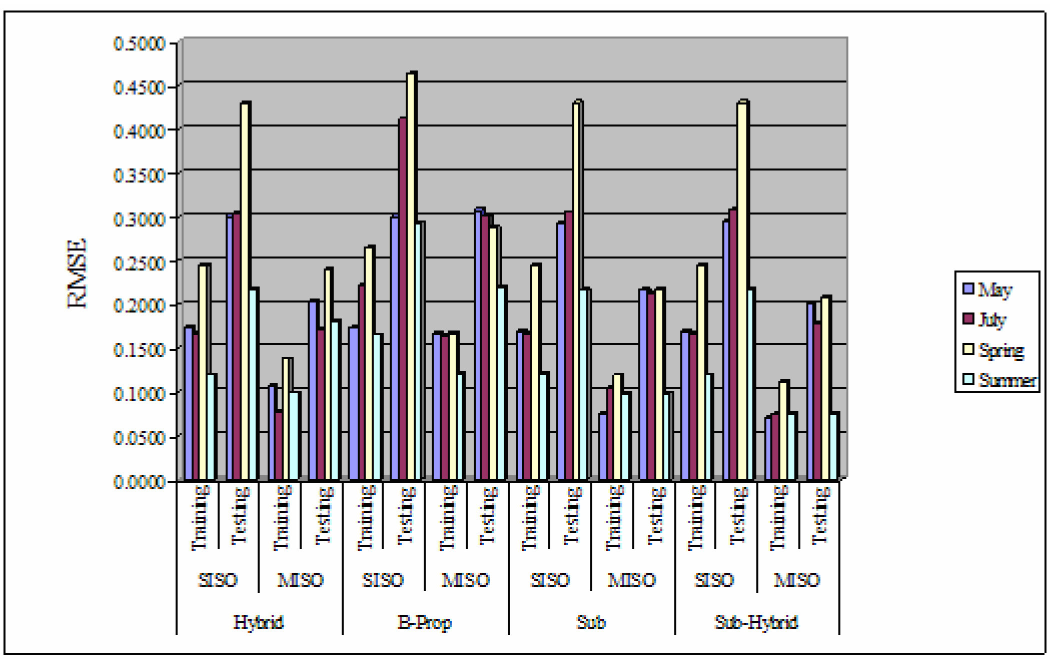


Fig. (6.9): The RMSE Measures Chart for all the SISO and MISO models

Table (6.7) shows the average results of the error measure RMSE for all the models. These results are shown graphically in Fig. (6.10). It is clear from the table and the figure that the lowest values are for the MISO models developed using the cascaded models with an average of the RMSE equal to (0.08).

Table (6.7): The Average RMSE Measures for all the Developed Models

Optimization Method	SISO		MISO	
	Training	*Testing*	*Training*	*Testing*
Sugeno With Hybrid Optimization	0.17580	0.31280	0.10505	0.19883
Sugeno With B-Prop Optimization	0.20608	0.36710	0.15453	0.27873
Sugeno Using Subtractive	0.17463	0.31113	0.09908	0.18628
Sugeno Using Subtractive with Hybrid	0.17423	0.31240	0.08310	0.16485

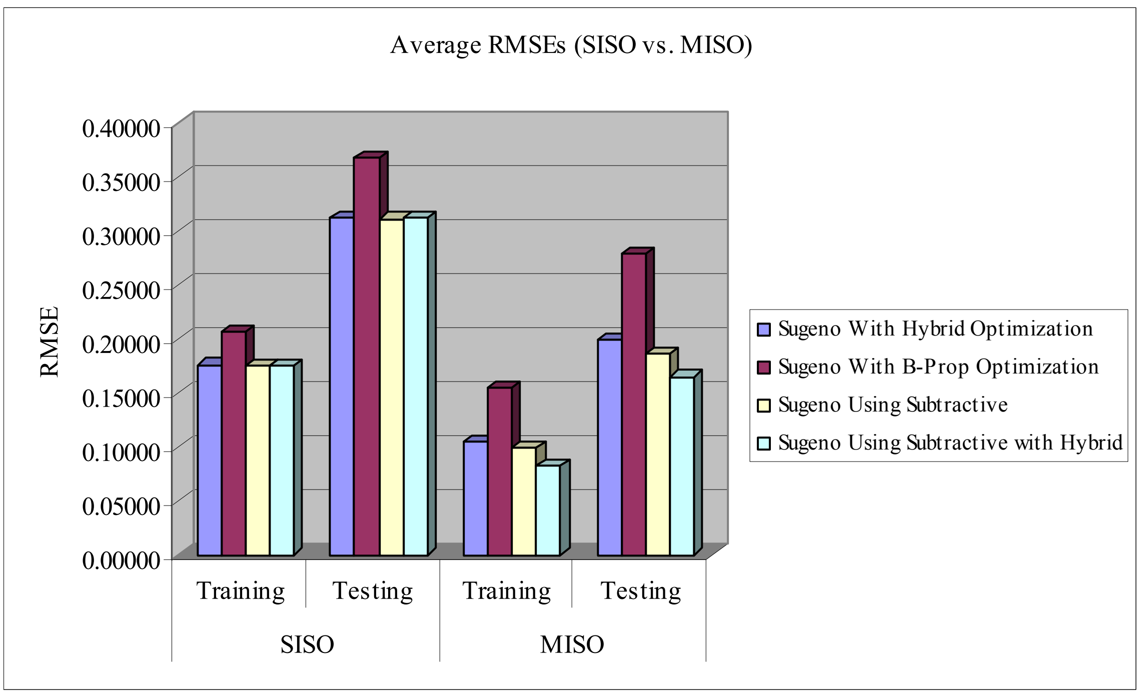


Fig. (6.10): The Average RMSE Measures Chart for all the SISO and MISO models

A graphical representation for the average CC and error measures (RMSE and MAPE) are shown in Fig. (6.11). The three measures are combined together in the same graph to take a clear look to the behavior of these measures. A relation can be concluded from the graph which is: an increasing in the CC leads to decrease in the error measures (RMSE and MAPE). As an example for this, from the above tables when the average CC for the MISO cascaded models are equal to (0.97) the corresponding average MAPE and RMSE values for the cascaded model are equal to (0.03 and 0.08) respectively. The second case is when the average CC measure for the SISO cascaded models is equal to (0.87) the corresponding average error measures MAPE and RMSE for the cascaded model are equal to (0.08 and 0.17) respectively.

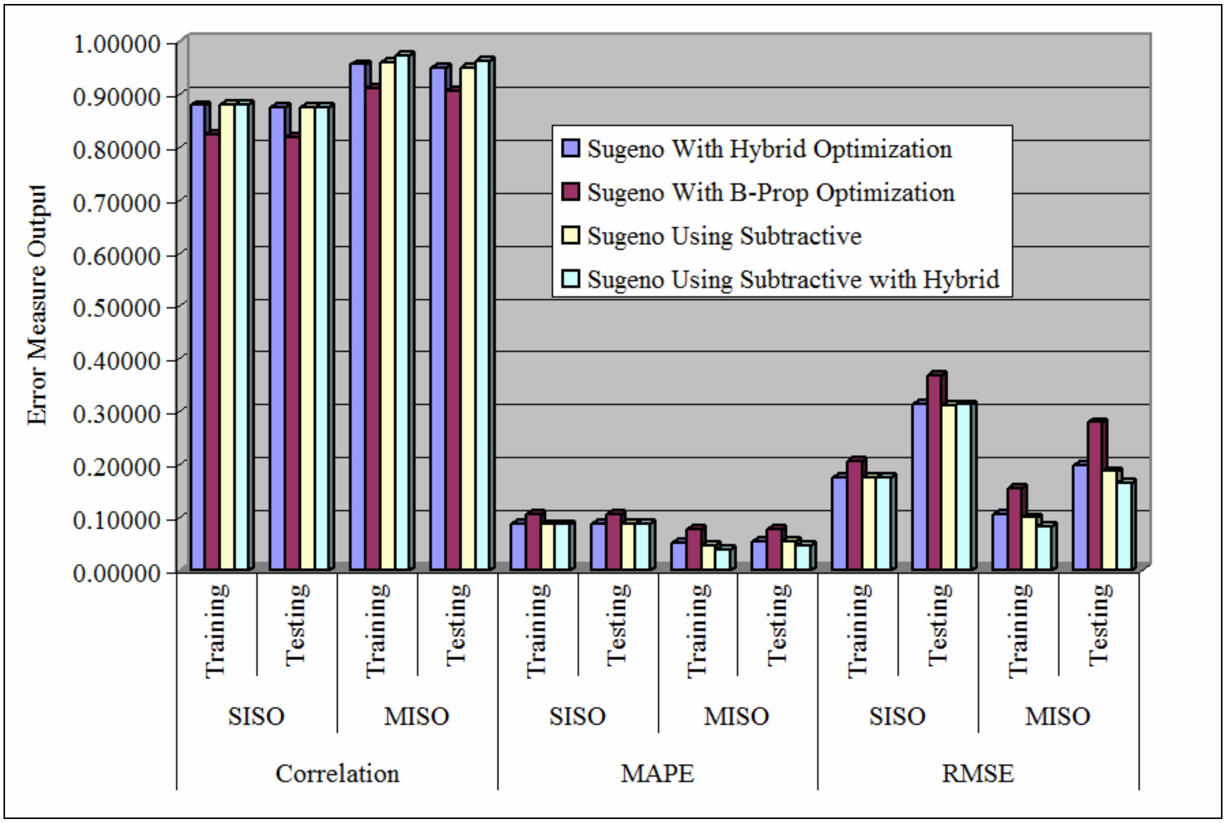


Fig. (6.11): The Average Correlation Measures Against the Error Measures (MAPE and RMSE) for all the developed SISO and MISO models

Fig. (6.11) can be summarized by the following points:

1. The developed cascaded models (Subtractive Clustering with Hybrid optimization) produced the highest results because of applying two optimization techniques in developing these models. The average CC between the actual and predicted loads ranging between 0.95 and 0.98 with average equal to 0.97.

2. The developed models with Back-propagation optimization techniques have the lowest CC compared to the other models and similarly for the two error measures. The correlation coefficient ranging between 0.89 and 0.91 with average equal to 0.90.

3. The developed models using the Subtractive Clustering can be enhanced when subject to the Hybrid optimization technique. The average correlation for the developed

models using the Subtractive Clustering is equal to 0.95 and after applying the Hybrid optimization technique it is enhanced to an average equal to 0.97.

4. Finally, we can notice that the highest the CC the lowest the MAPE and RMSE. For example, the average CC of the cascaded model is equal to 0.97, the average MAPE is equal to 0.03, and the average RMSE is equal to 0.08. While, the average CC of the developed models using the Back-propagation optimization technique is equal to 0.90, the average MAPE is equal to 0.07, and the average RMSE is equal to 0.15.

6.3 One Day and One Week Prediction using the Unseen Datasets from the Year 2008

As mentioned in the previous chapter, the historical profiles which have been obtained for the year 2008, used to test our models to predict the power load for a period of one day and one week. These datasets have been not used in the cross validation for developing the models. The average CC that has been obtained for one day prediction is equal to 0.94 and the average MAPE is equal to 0.0585. The average correlation in case of one week prediction is equal to 0.93 while the average MAPE is equal to 0.0595. Fig. (6.12) and Fig. (6.13) represent the average CC and error measures obtained for one day and one week prediction using the developed models.

From the figures below it is clear that the best results have been obtained from the general May model in case of one day and one week prediction. The lowest average CC has been obtained from the general Spring model because of the wide range of the model input parameters (the wide variations of low and high temperatures) as it has been mentioned in chapter four (Table (4.2)).

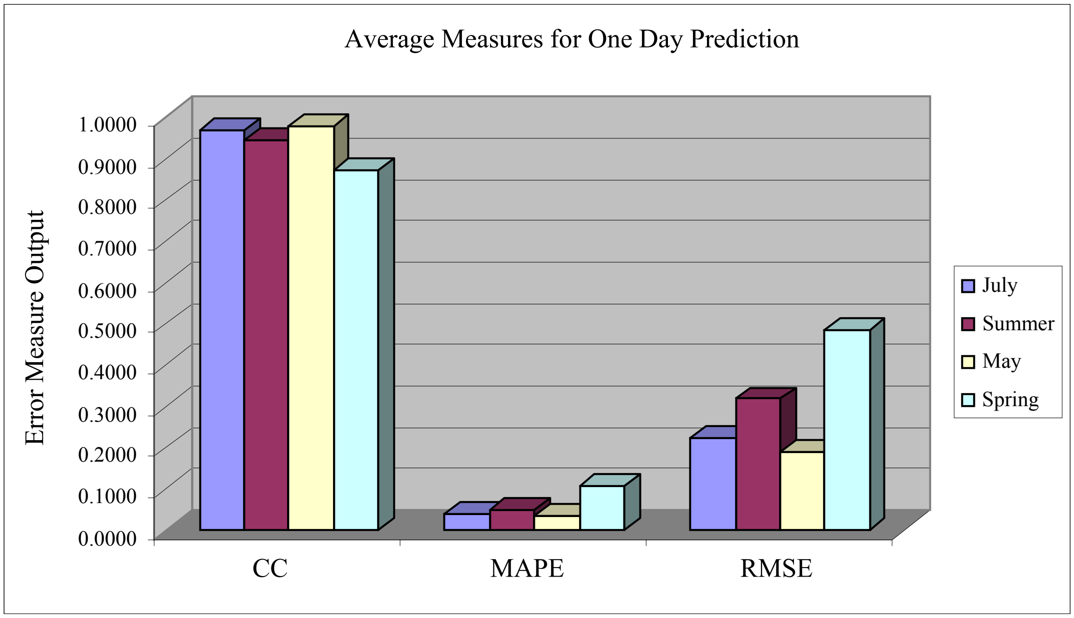


Fig. (6.12): The Average Correlation and Error Measures for One Day Prediction.

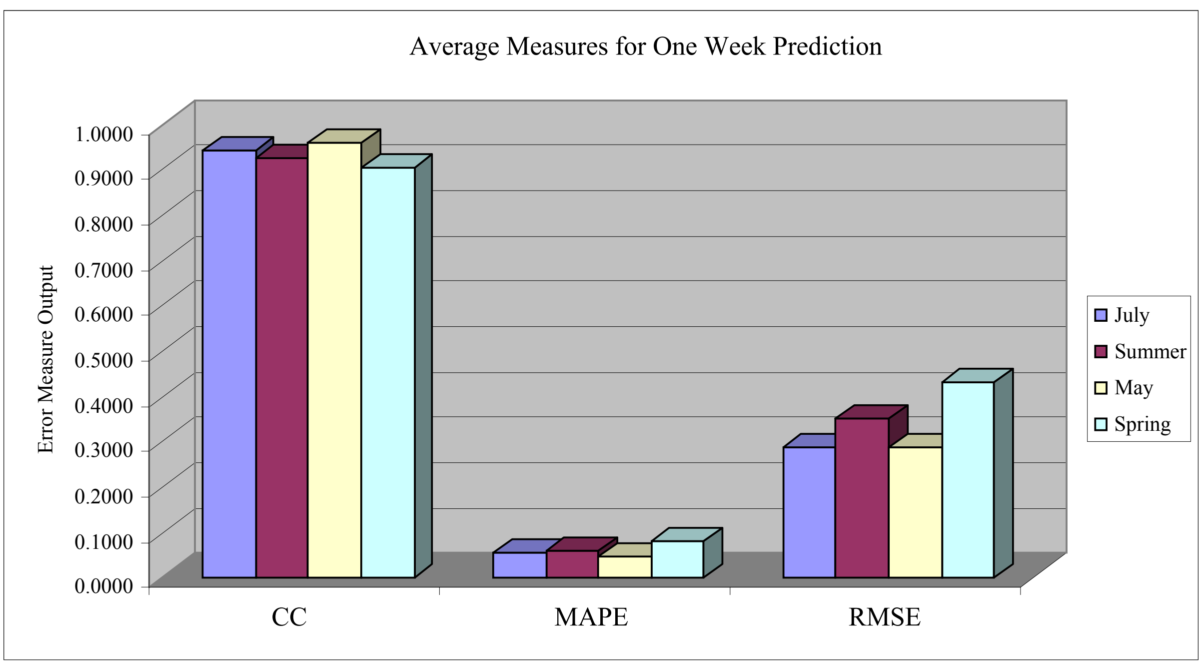


Fig. (6.13): The Average Correlation and Error Measures for One Week Prediction.

The results in Fig. (6.12) and Fig.(6.13) show the accuracy of the developed models to predict the power loads for the new unseen datasets one day and one week ahead.

6.4 Comparison with Other Studies

As mentioned in chapter two a plenty of works can be found in the STLF field. Some of these works are mentioned here briefly. Many papers that have been published recently in the refereed journals are considered and the ones whose main interests are STLF by soft computing methods are taken into account.

It is important to mention that different datasets and different approaches have been used in developing these models. However, we are trying to compare the obtained MAPE as a measure of errors. Furthermore, the same equations of the CC and MAPE that have been presented in chapter five used in these papers, while in some papers the Average Percentage Error (APE) is used to check the adequacy of the developed models. All the CC, APE and MAPE results listed as a real number with fractions instead of using the percentage sign.

The APE can be calculated by the equation (Tamimi et al., 2000):

$$\text{APE}=\left|\frac{y_i - x_i}{y_i}\right| * 100\% \quad , \tag{6.1}$$

And then we can represent the MAPE in equation (5.6) using the APE by the following one (Tamimi et al., 2000):

$$\text{MAPE}=\frac{1}{N}\sum_{i=1}^{N} APE \quad , \tag{6.2}$$

where y_i : is the i^th^ actual data,

x_i : is the i^{th} predicted data.

N : is the number of data points under consideration.

Equation (6.2) above indicates that the used MAPE measure represents the average APE.

Hwang (Hwang et al., 2001) described a new practical knowledge-based expert system (called LoFY) for short-term load forecasting equipped with graphical user interfaces. Also, various forecasting models like trending, multiple regression, artificial neural networks, fuzzy rule-based model, and relative coefficient model have been included to increase the forecasting accuracy. The simulation based on historical sample data shows that the forecasting accuracy is improved when compared to the results from the conventional methods. Through the fuzzy rule-based approach, the forecasting accuracy at special days has been improved remarkably. The average MAPE results found for this system is 0.020.

Erkmen and Topallı reported four methods for STLF in their recent work (Erkmen et al, 2003). These methods are generalized learning vector quantization for data clustering, genetic algorithms for optimum topology, neural networks and fuzzy logic for forecasting. The one giving the most successful forecasts is a hybrid neural network model which combines off-line and on-line learning and performs real-time forecasts 24-hour in advance. Loads from all day types are predicted with 0.017 average APE for working days, 0.017 for Saturdays and 0.020 for Sundays.

Khan (Khan et al., 2001) presented a comparative study of six soft computing models namely multilayer perceptron networks, Elman recurrent neural network, radial basis function network, Hopfield model, fuzzy inference system and hybrid fuzzy neural network for the hourly electricity demand forecast of Czech Republic. The soft computing models were trained and tested using the actual hourly load data obtained from the Czech Electric Power Utility for seven years (January 1994 – December 2000). A comparison of the proposed techniques is presented for predicting 48 hourly demands for electricity. Simulation results indicate that hybrid fuzzy neural network and radial basis function networks are the best candidates for the analysis and forecasting of electricity demand for the experimented data, with the following MAPEs: For weekday forecast, 0.010 by radial basis function networks, 0.009 by fuzzy neural network; and for weekend forecast, 0.013 by radial basis function networks, 0.020 by fuzzy neural network.

Kim (Kim et al., 1995), presented a model for STLF that integrates neural networks and Fuzzy Expert System (FES) is presented. The load forecast is obtained by passing through two steps. In the first procedure, the neural networks are trained with the load patterns corresponding to the forecasting hour, and the provisional load is forecast by the trained neural network. In the second phase, the fuzzy expert systems modify it considering the possibility of load variations due to the changes in temperature and the day type, regular or holiday. Proposed model is tested with the data from Korea Electric Power Corporation. Results show that specific rules are required to deal with the consecutive holidays that have inconsistent periods with the previous year. Moreover, several rules are added for special days that have elections, rainy season, typhoon or special television programs. The proposed model predicts the load of holidays with a similar forecasting accuracy of non-holidays where the conventional methods or neural networks provide poor forecasts. Non-holiday average APE results are given for four months: 0.0125 for January, 0.0116 for February, 0.0130 for March, and 0.0107 for April. The average APE for holidays is 0.0219.

A modeling technique based on the fuzzy curve notion is proposed by Papadakis (Papadakis, 1998) to generate fuzzy models for STLF. Different forecast models are developed for each day type in every season. The model is considered as a fuzzy neural network described in terms of a parameter vector and is trained using a genetic algorithm with enhanced learning and accuracy attributes. The performances of the developed fuzzy models are tested using load data of the Greek interconnected power system. They achieve a MAPE of 0.0167 with the data from year 1995.

Saini and Soni's work (Saini et al., 2002), the daily electrical peak load forecasting is done using the feed forward neural network based upon the conjugate gradient backpropagation methods, by incorporating the effect of 11 weather parameters, the previous day peak load information, and the type of day. To avoid the trapping of the network into a state of local minima, the learning rate and error goal optimizations are performed. For redundancy removal in the input variables, reduction of the number of input variables is done by the principal component analysis method. The resultant data set is used for the training of a three-layered neural network. To increase the learning speed, the weights and biases are initialized according to the Nguyen and Widrow method. To avoid over fitting, an early stopping of training is done at the minimum validation error. The daily weather and electrical peak load data of four years of Haryana Vidyut Prasaran Nigam Ltd., India is taken for this study. Data from 1997 to 1999 are used for neural network training. Data of the year 2000 are used to test the trained neural network. Among the experimented conjugate gradient algorithms, the Powell-Beale method has given the best performance with 0.0231 MAPE.

A feed-forward neural network with a back-propagation algorithm is presented by Bhattacharyya (Bhattacharyya et al., 2004) for three types of short-term electric load forecasting: daily peak (valley) load, hourly load and the total load. The forecast has been made for the northern areas of Vietnam using a large set of data on peak load, valley load, hourly load and temperature. The data were used to train and calibrate the artificial neural network, and the calibrated network was used for load forecasting. The results obtained from the model show that the application of neural network to short-term electric load forecasting problem is very useful with quite accurate results. The method has given the best performance with 0.9427 CC and 0.108 MAPE.

A Fuzzy Logic (FL) expert system is integrated with Artificial Neural Networks (ANN) for a more accurate short-term load forecast is presented by Tamimi (Tamimi et al, 2000). The 24 hour ahead forecasted load is obtained through two steps. First, a FL module maps the highly nonlinear relationship between the weather parameters and their impact on the daily electric load peak. Second, 12 ANN modules are trained using historical hourly load and weather data combined with the FL output data, to perform the final forecast. Comparisons made between this model, an ANN model, and an Autoregressive Moving Average (ARMA) model were show the efficiency and accuracy of this new approach. The average MAPE for these methods is found equal to 0.029.

Table (6.8) below summarizes the approaches mentioned above with the average APE and MAPE that have been obtained.

Table (6.8): Comparison between the Developed STLF

Researcher	**Approach**	**Average APE/ MAPE**
Hwang et al., 2001	MR, ANN, FL, and KBES	0.020
Erkmen et al., 2003	Data Clustering, GA, ANN, and FL	0.017 for work days, and 0.020 for weekends
Khan et al., 2001	MPN, ERNN, RBFN, Hopfield, FIS, and HFNN	0.010 by RBFN, 0.009 by HFNN for weekdays, and 0.013 by RBFN, 0.020 by HFNN for weekends
Kim et al., 1995	ANN integrated with FES	0.0125 for January, 0.0116 for February, 0.0130 for March, 0.0107 for April, and 0.0219 for holidays
Papadakis, 1998	FNN integrated with GA	0.0167
Saini et al., 2002	ANN with BackProagation	0.0231
Bhattacharyya et al., 2004	ANN with BackProagation	0.1080
Tamimi et al., 2000	FL integrated with ANN	0.0290
Basbous, 2009	Sugeno FIS with different optimization techniques	Average for the cascading model 0.0387
MR: Multiple Regression***,*** ***KBES***: Knowledge Based Expert System, ***GA***: Genetic Algorithms, ***FL***: Fuzzy Logic, ***MPN***: Multilayer Perceptron Networks, ***ERNN***: Elman Recurrent Network, ***RBFN***: Radial Basis Function Network, ***FIS***: Fuzzy Inference System, ***HFNN***: Hybrid Fuzzy Neural Network***,*** ***ANN***: Artificial Neural Networks, ***FES***: Fuzzy Expert System, ***FNN***: Fuzzy Neural Networks,		

The summery of the results that have been mentioned above are graphically presented by the author and the year when the work published in Fig. (6.12).

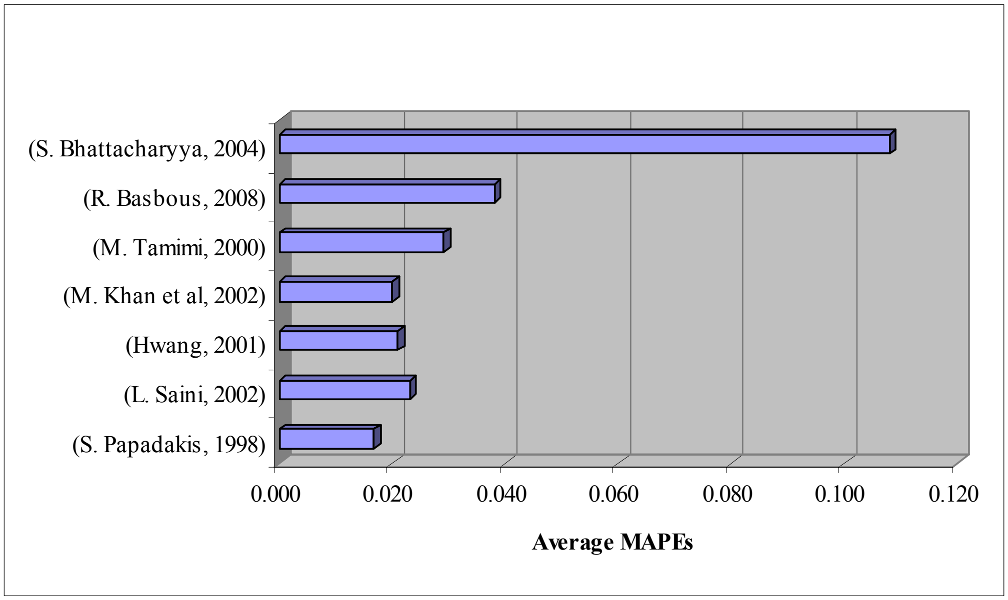


Fig. (6.14): MAPEs Results Comparison with Other Studies

Average Correlation comparison is presented in Fig. (6.13) between our work results and the results obtained by Bhattacharyya in (S. Bhattacharyya, 2004).

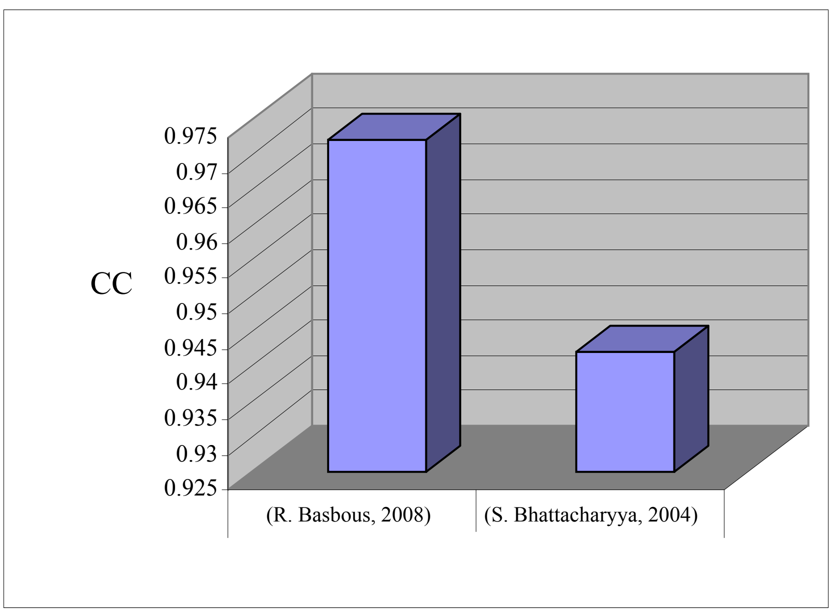


Fig. (6.15): Average Correlation Results Comparison

From the results mentioned above, it is clearly noticed that the soft computing methods provide a promising solution to the STLF problem. In addition, combining or integrating more than one method together leads to an enhancement to the proposed models. For example

Tamimi (Tamimi et al, 2000) combined the NN with FL, Kim (Kim et al., 1995) integrated NN with FES, and Hwang (Hwang et al., 2001) developed a forecasting system and include it with different forecasting models to increase the forecasting accuracy. Furthermore, the lowest results over the proposed models that haven mentioned above found in the models with Back-propagation optimization proposed by Bhattacharyya (Bhattacharyya et al, 2004). This agree with the results obtained from our developed models with Back-propagation optimization.

Comparing our results with these systems we can see that our developed models produced satisfactory results using the temperature parameters only to predict the electric load without taking in account the other weather parameters or the type of the day or any other conditions. In our developed models, the CC for one day ahead prediction ranges between (0.87 and 0.97) with an average value 0.94; the corresponding MAPE ranges between (0.03 and 0.10) with an average 0.05. Whereas; the obtained CC for one week ahead prediction ranges between (0.90 and 0.95) with an average value 0.93; the corresponding MAPE ranges between (0.04 and 0.08) with an average 0.05.

An improvement to the results that have been obtained from our developed models can be achieved when a hourly temperature and weather data profiles are available. Also, an average high and low temperature of the day which has been considered in the MISO models leads to reduce the error measures between the actual and predicted power loads compared to SISO models.

Chapter Seven:

Conclusions and Suggestions for Future Works

7.1 Conclusions

The general objective of this work is to explore the use of soft computing and artificial intelligence approaches to develop Short- Term Load Forecasting (STLF) system that predict the power load for one day up to one week ahead in specific month or specific season.

In conclusion, it is mentioned that Soft Computing is an emerging approach to computing, which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. While, Artificial Intelligence approaches imitate human beings' way of thinking and reasoning to get knowledge from the past experience and predict the future load. In the modern electricity market, the energy trade is based on a precise load forecasting result. The significance of STLF inspires us to go through this work.

A state of arts about STLF has been presented in this thesis. The research approaches of STLF can be mainly divided into two categories: statistical approaches and artificial intelligence approaches. In the statistical approaches, equations can be obtained showing the relationship between load and its relative factors after training the historical datasets, while artificial intelligence approaches try to imitate human beings' way of thinking and reasoning to get knowledge from the past experience and forecast the future load. The statistical category includes multiple linear regression, general exponential smoothing, state space, etc. Expert system, ANN and Fuzzy Inference approaches belong to the artificial intelligence category.

In our review of literature survey of STLF approaches we have been found that Various weather variables could be considered for load forecasting such as, temperature, humidity, wind speed, cloud cover, light intensity and so on. Temperature and humidity are the most commonly used load predictors.

On the whole, this thesis is composed of three parts: historical data treatment, individual approaches proposed for load forecasting, and the design of an integrative and convenient stand alone GUI system combining different approaches.

Real JDECO power line in Bier Nabala village and PMO historical data profiles for two years (2006 and 2007) have been collected and used to develop and test the various models. The JDECO data includes the time (a reading at every 20 minutes approximately) and the corresponding power load at that time. Furthermore, the weather history data, provided by PMO, including humidity, highest temperature, lowest temperature, and wind speed for each day.

The existence of bad data in the historical load curve affects the precision of the load forecasting result. There are three kinds of bad data in the daily load curve: False channel bad data, abnormal event bad data and zero load bad data. An existing algorithm to detect the outliers has been used and integrated with a designed algorithm to remove these outliers from the available training datasets. A CC measures have been applied to ensure that there is no loss in signal details after removing the outliers. The CC found equal to 0.998 which shows that there is no crucial loss in the signal details.

Developing the models for load forecasting has been applied firstly using the available datasets. Two kinds of models have been developed, Single Input Single Output (SISO) models, and Multiple Inputs Single Output (MISO) models. Three main inputs (Time (T), High Temperature (HT), and Low Temperature (LT)) and one output (Power Load (PL)) have been considered in building these models. For the SISO models, only the time has been considered as the input for the models and the power load at that time has been used as the output.

The weather condition is very influential to the load. Common weather variables include temperature, humidity, sunshine duration, amount of daylight, wind velocity. In this thesis it has been found that the temperature is a major input parameter on STLF. Models, that do not utilize temperature measurements in training, produce quite larger errors than the ones exploiting them as input parameters. The correlation has been improved from 91% for the SISO model to about 98% for the MISO model as demonstrated in Table (5.1) using the same parameters (number of MFs, type of MFs, and cluster radius). These results clearly reveal the effect of the temperature parameters on predicting the power load.

Fuzzy Inference System (FIS) with different optimization techniques have been used to develop our models. Firstly we started by developing a SISO and MISO models using ANFIS with hybrid optimization technique. Then a SISO and MISO models have been developed using Back-propagation optimization technique. After that the Subtractive clustering has been used to develop such models. Finally cascaded models have been developed for STLF by constructing the models using the Subtractive clustering and then learning these models using the hybrid optimization techniques to achieve more accurate models.

The adequacy of the developed models has been checked using the Correlation Coefficient (CC) to measure the agreements between the actual and predicted power loads. In addition, two error measures were used namely, Mean Absolute Performance Error (MAPE), and the Root Mean Square Error (RMSE) to indicate the accuracy and the performance of the developed models.

While testing these models using the testing datasets that has been isolated before the training stage using the developed cross validation algorithm, the average obtained CC between the actual and predicted power loads for all the developed SISO models ranges between (0.81 and 0.87), with an average CC value 0.85; The corresponding average MAPE that ranges between (0.08 and 0.10) with an average value of 0.09, and average RMSE that ranges between (0.31 and 0.36), with an average value of 0.32. Whereas; the obtained average CC for the MISO models ranges between (0.90 and 0.96) with an average CC value 0.94; The corresponding average MAPE that ranges between (0.04 and 0.07) with an average value of 0.05, and average RMSE that ranges between (0.16 and 0.27) with an average value 0.20. This demonstrates the adequacy of adopting these types of approaches to STLF problem as well as the improvements of models' forecasting performance when taking the weather into consideration.

It was noticed that the forecasting performance has been furtherlly improved by the MISO cascaded models, while maintaining all other factors including MFs types and numbers, and cluster radius. This improvement is notices as an improvement in the obtained CC that results from the MISO cascaded models which ranges between (0.95 and 0.97) and with an average CC value of 0.96 for all the developed models as compared with the forecasting performance of CC that ranges between (0.90 and 0.94) when developing the models using the other optimization techniques; The corresponding MAPE ranges between (0.03 and 0.06) with an average value 0.04, and RMSE ranges between (0.07 and 0.20) with an average value 0.16 compared to average MAPE ranges between (0.05 and 0.07), and average RMSE ranges between (0.18 and 0.27) for the other optimization techniques.

These models have been integrated with a stand alone application with GUI. The developed Electric Power Load Forecasting System "***EPLFS***" can be accessed online through either a local area network, or using a web server. Online testing "load forecasting" can be done using this system, so one can load the input datasets saved in a text file for example (.DAT file), obtain the forecasted load using the developed models, plot the predicted power loads against the actual ones if available, and evaluate the predicted output by calculating several measures including the CC, MAPE and RMSE.

The EPLFS has been tested using the obtained power load historical profile for the year 2008 and used as the actual load. The system has been used to predict the load for one day and one week ahead using the developed models. In case of one day ahead prediction the first day of May and July (01/05 and 01/07) has been considered while in case of one week ahead prediction the first week of May and July has been considered (01-07/5 and 01-07/07). The CC for one day ahead prediction ranges between (0.87 and 0.97) with an average value 0.94; The corresponding MAPE ranges between (0.03 and 0.10) with an average 0.05, and RMSE ranges between (0.19 and 0.48) with an average 0.30. Whereas; the obtained CC for one week ahead prediction ranges between (0.90 and 0.95) with an average value 0.93; The corresponding MAPE ranges between (0.04 and 0.08) with an average 0.05, and RMSE ranges between (0.28 and 0.43) with an average 0.33.

Finally, different works in the field of STLF using different techniques accomplished by other researchers have been compared with our developed models. These works show the ability of the soft computing techniques to represent the STLF, and agree with our results that the Back-propagation optimization technique produced the lowest results. Our over all results indicates the suitability and adequacy of the developed models to solve the short term load forecasting problem using the time and weather variables.

7.2 Suggestions and Future Works

Although, we have obtained a preliminary and promising results, but still the following recommendations may help to further contributions in this area:

1. Further investigations using more historical data and varying parameters of the models need to be performed to conclude the adequacy of these approaches to solve the STLF problem.

2. The developed models are methods to find the input-output relationship. Therefore they shouldn't be limited to short-term load forecasting. Future work might employ these proposed methods to mid-term and long-term load forecasting.

3. In this work other weather parameters including humidity, wind speed and sunshine were neglected due to the lack of data. It is worthy to consider these parameters in future. Also, there can be several alternative approaches that use different temperature variables as inputs. Among them, the highest and the lowest temperature values of past days, the greatest and the lowest temperature forecasts of forecast day, the average temperature forecast of the forecast day and the temperature forecast of the forecast hour can be mentioned and considered in further development of our proposed models. Furthermore, we can take into account the change in population.

4. Daily load profile has several characteristic regions, such as working hours, startup hours, evening, night, etc. Data clustering can be performed according to this discrimination and separate models can be formed. Additional things to be considered in developing the models are the holidays, weekends, and special days.

5. After a year is passed, off-line training should be repeated with the last year's data to enhance the outcome of the developed models since the electric power consumption profile can be changed due to many factors (weather conditions, change in population, price, etc.).

6. To compare the developed models with other techniques using the same historical datasets profiles.

7. Additional suggestion to be a future work is an enhancement to the stand alone application to be able to automatically pre-process the available datasets (detecting and removing the outliers, bad data, time formatting), and to online train the models with new available datasets, and accordingly predict the power load.

8. Finally, it is worthy to explore the use of other different soft computing modeling techniques such as Genetic Algorithms, and Wavelets for the Short Term Load Forecasting (STLF).

References

1. Al-Shareef, A., Mohamed, E., Al-Judaibi, E. (2008), One Hour Ahead Load Forecasting Using Artificial Neural Network for the Western Area of Saudi Arabia, Proceedings of World Academy of Sceince, Engineering and Technology, Vol. [27], pp. 219-221, 2008.

2. Amjady, N. (2001), 'Short-term hourly load forecasting using time-series modeling with peak load estimation capability', IEEE Transactions on Power Systems, Vol. [16], No.(3), pp. 498 – 505.

3. Arafeh, L., Singh, S., Putatunda, S. (1999), 'A neuro fuzzy logic approach to material processing', IEEE Transactions on Systems, Man, and Cybernetics, Part C, Vol. [29], No.(3), pp. 362-370.

4. Armstrong, J. (2001), Principles of Forecasting: A Handbook for Researchers and Practitioners, Kluwer Academic Publishers, Dordrecht.

5. Azuaje, F., Dubitzky, W., Black, N., Adamson, K. (2000), "Discovering Relevance Knowledge in Data: A Growing Cell Structures Approach," IEEE Transactions on Systems and Machine Cybernetics-Part B: Cybernetics, Vol. [30], No.(3), pp. 448.

6. Bakirtzis, A., et al. (1996), 'A neural network short-term load forecasting model for the Greek power system', IEEE Transactions on Power Systems, Vol. [11], pp. 858 – 863.

7. Barnett, V., Lewis, T. (1994), Outliers in Statistical Data, Wiley Series in Probability & Statistics, Chichester, 3rd Edition.

8. Bhatnagar, R., Rahman, S. (1986), Application of knowledge based algorithms in electric utility load forecasting, Proceedings of the IEEE South east con., Richmond, Virginia, pp. 60-64.

9. Bhattacharyya, S., Thanh, L. (2004), Short-Term Electric Load Forecasting Using an Artificial Neural Network: Case of Northen Vietnam, International Journal of Energy Research, Vol. [28], pp. 463-472.

10. Bougadis, J., Adamowski, K., Diduch, R. (2005), Short-Term Municipal Water Demand Forecasting, Wiley InterScience, Hydrol. Process., Vol. [19], pp. 137-148.

11. Bryson, A., Ho, Y. (1969), Applied Optimal Control: Optimization, Estimation and Control, Blaisdell, New Yourk.

12. Carpenter, G., Grossberg, S. (1987), ART2 Self-Organisation of Stable Category Recognition Codes for Analog Input Patterns, Applied Optics, Vol. [26], pp. 4919-4930.

13. Chen, B., Chang, M., Lin, C. (2004), 'Load forecasting using support vector machines: A study on EUNITE competition 2001', Transactions on Power Systems, Vol. [19], No. (4).

14. Chen, H., Canizares, C., and Singh, A. (2001), ANN-Based Short-Term Load Forecasting in Electricity Markets. Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference, Vol. [2], pp. 411–415.

15. Chiu, S. (1994), Fuzzy model identi6cation based on cluster estimation, Journal of Intelligent Fuzzy Systems, Vol. [2], pp. 267–278.

16. Chiu, S. (1997), Extracting fuzzy rules from data for function approximation and pattern classification, in: D. Dubois, H. Prade, R. Yager (Eds.), Fuzzy Information Engineering: a Guide Tour of Applications, John Wiley & Sons, New York, Chapter 9, pp. 149–162.

17. Chow, T., Leung, C. (1996), 'Nonlinear autoregressive integrated neural network model for short-term load forecasting', IEE Proceedings, Generation, Transmission and Distribution, Vol. [143], pp. 500 – 506.

18. Chow, T., Leung, C. (1996), Neural network based short-term load forecasting using weather compensation, IEEE Transactions on Power Systems, Vol. [11], No. (4), pp. 1736-1742.

19. Christianse, W. (1971), 'Short Term Load Forecasting Using General Exponential Smoothing', IEEE Transactions on PAS, pp. 900 – 910.

20. Chuang, C., Jen, J. (2004), A Soft Computing Technique for Noise Data with Outliers, Proceedings of the 2004 IEEE, International Conference on Networking, Sensing & Control, Taipei, Taiwan, March 21-23.

21. Dennis, J. (1977), Nonlinear least-squares, State of the art in numerical analysis, Academic Press, Orlando, FL.

22. Desouky, A., Elkateb, M. (2000), 'Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA', IEE Proceedings of Generation, Transmission and Distribution, Vol. [147], No.(4), pp. 213 – 217.

23. Encarta (2008), Meteorology: Weather Prediction, MSN Encarta Encyclopedia, http://encarta.msn.com/encyclopedia_761571037_5/Meteorology.html.

24. Erkmen, I., Oezdogan, A. (1997), 'Short Term Load Forecasting Using Genetically Optimized Neural Network Cascaded With a Modified Kohonen Clustering Process', Proceedings of the 12th IEEE International Symposium on Intelligent Control, Vol. [1], pp. 107 – 112.

25. Erkmen, I., Topallı, A. (2003), Four methods for short term load forecasting using the benefits of artificial intelligence, Electrical Engineering, Vol. [85], pp. 229–233, Springer-Verlag.

26. Feinberg, E., Genethliou, D. (2005), Applied Mathematics for Power Systems, Load Forecasting, Chapter 12, pp. 269-285.

27. Felix, R. (2002), ‘Electric power load forecasting using periodic piece-wise linear models’, Examine thesis in control theory, Institution of mathematical statistics, Technical University of Lund, pp. 11-12.

28. Gellings, C. (1996), Demand Forecasting for Electric Utilities, The Fairmont Press, Lilburn, GA.

29. Gross, G., Galiana, F. (1987), ‘Short-term load forecasting’, Proceedings of the IEEE, Vol. [75], No. (12), pp. 1558 – 1571.

30. Guldemır, H., Sengur, A. (2006), Comparison of clustering algorithms for analog modulation classification, Expert Systems with Applications, Vol.[30] , pp. 642–649.

31. Guler, I., Ubeyli, E. (2004), Application of adaptive neuro-fuzzy inference system for detection of electrocardiographic changes in patients with partial epilepsy using feature extraction, Expert Systems with Applications, Vol. [27], pp. 323–330.

32. Hartigan, J., Wong, M. (1979), “A k-means clustering algorithm,” Applied Statistics, Vol. [28], pp. 100-108.

33. Hebb, D. (1949), The Organisation of Behaviours, John Wiley & Sons, New York.

34. Hippert, H., Pedreira, C., and Souza, R. (2001), Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. IEEE Transactions on Power Systems, Vol. [16], pp: 44–55.

35. Ho, K. et al (1990), ‘Short-term load forecasting of Taiwan power system using a knowledge based expert system’, IEEE Transactions on Power Systems, Vol. [5], pp. 1214 – 1221.

36. Hopfield, J. (1982), Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proc. Nat. Acad. Sci, USA, Vol. [79], pp. 2554-2558.

37. Huang, C., Yang, H. (2001), Evolving Wavelet-Based Networks for Short Term Load Forecasting, IEE Proceedings - Generation, Transmission and Distribution, Vol. [148], No.(3), pp. 222-228.

38. Hwang, K., Kim, G. (2001), ‘A short-term load forecasting expert system’, Proceedings of The Fifth Russian-Korean International Symposium on Science and Technology, pp. 112 – 116.

39. Jang, J. (1993), ANFIS: Adaptive-network-based fuzzy inference system, IEEE Transactions on Systems and Machine Cybernetics, Vol. [23], No.(3) pp. 665–685.

40. Jang, J., Sun, C. and Mizutani, E. (1997), Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice-Hall, Englewood Cliffs, NJ, Chapter One, Chapter Eight, and Chapter Twelve, pp. 7-9 and pp. 219-336.

41. Jin, Y. (2003), Advanced Fuzzy Systems Design and Applications, Springer, pp. 26-27.

42. Jonung, J., Larch, M. (2006), Improving the Fiscal Policy in the EU: the Case for Independent Forecasts, Economic Policy, pp. 491-543.

43. Karayiannis, N. (1996), Hybrid learning schemes for fast training of feed-forward neural networks, Department of Electrical and Computer Engineering, University of Houston, USA, Mathematics and Computers in Simulation, Vol. [41], pp. 13–28.

44. Khan, M., Abraham, A., Ondrusek, C. (2001), Soft computing for developing short term load forecasting models in Czech Republic, Hybrid Information Systems, pp. 207-222.

45. Kholer, U. (1995), Traffic and Transport Planning in German Cities, Transportation Research Part A, Vol. [29A], pp. 253-261.

46. Khotanzad, A., Davis, M., Abaye, A., and Martukulam, D. (1996), An Artificial Neural Network Hourly Temperature Forecaster with Applications in Load Forecasting, IEEE Transactions om Power Systems, Vol. [l1], No.(2), pp. 870-876.

47. Khotanzad, A., et al. (1997), 'ANNSTLF - A neural-network-based electric load forecasting system', IEEE Transactions on Neural Networks, Vol. [8], pp. 835 – 846.

48. Khotanzad, A., Rohani, R., Maratukulam, D. (1998) 'ANNSTLF – Artificial neural network short-term load forecaster - Generation three', IEEE Transactions on Neural Networks, Vol. [13], pp. 1413 – 1422.

49. Kiartzis, S., Bakirtzis, A. (2000), 'A Fuzzy expert system for peak load forecasting. Application to the Greek power system', 10th Mediterranean Electrotechnical Conference, Vol. [3], pp. 1097 – 1100.

50. Kim, K. et al. (1995), Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems, IEEE Transactions on Power Systems, Vol. [10], No. (3), pp. 1534-1539.

51. Kim, K., Youn, H., Kang, Y. (2000), 'Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference method', IEEE Transactions on Power Systems, Vol. [15], No.(2), pp. 559 – 565.

52. Kohonen, T. (1989), Self-Organisation and Associative Memory, Third Edition, Springer Verlag, Heidelberg.

53. Kotsiantis, S., Kanellopoulos, D., Pintelas, P. (2006), Data Preprocessing for Supervised Leaning, International Journal of Computer Science, Vol. [1], No.(2), pp 111-117.

54. Levenberg, K. (1944), A method for the solution of certain nonlinear problems in least squares, Quart Appl Math , Vol. [2], pp. 164–168.

55. Li, Y., Fang, T. (2003), 'Wavelet and support vector machines for short – term electrical load forecasting', Proceedings of the International Conference on Wavelet Analysis and its Applications, Vol.[1], pp. 399 – 404.

56. Li, Y., Fang, T., Yu, E. (2002), 'Short-term electrical load forecasting using least squares support vector machines', International Conference on Power System Technology, pp. 230 – 233.

57. Lin, C., Lee, C. (1996), "Neural Fuzzy Systems: a neuro-fuzzy synergism to intelligent systems," Prentice Hall, NJ.

58. Lin, L., Chang, H. (2007), An Adaptive Neuro-Fuzzy Inference System for Sea Level Prediction Considering Tide-Generating Forces and Oceanic Thermal Expansion, Department of Civil Engineering, National Chiao Tung University, Hsinchu, Taiwan.

59. Llata, J., Sarabia, E., Oria, J. (2001), Fuzzy expert system with double knowledge base for ultrasonic classification, Expert Systems with Applications, Vol. [20], pp. 347-355.

60. Mamdani, E., Assilian, S. (1975), An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man-Machine Studies, Vol. [7], No.(1), pp.1–13.

61. Marquardt, D. (1963), An algorithm for least-squares estimation of nonlinear parameters, Journal of the Society for Industrial and Applied Mathematics., Vol. [11], pp. 431–441.

62. MathWorks (2008), Fuzzy Logic Toolbox™ User's Guide, The Mathworks Inc., 3 Apple Hill Drive, Natick, Ma 01760-2098, http://www.mathworks.com

63. McCullock, W., Pitts, W. (1943), A Logical Calculus of the Ideas Imminent in Nervous Activity, Bulletin of Mathematical Biophysics, Vol. [5], pp. 115-133.

64. McSharry, P. (2006), 'Evaluation of short-term forecasting methods for electricity demand in France', RTE-VT workshop, Paris, May 29-30.

65. Miranda, V., Monteiro, C. (2000), 'Fuzzy inference in spatial load forecasting', Power Engineering Winter Meeting, IEEE, Vol. [2], pp. 1063 – 1068.

66. Nauck, D., Kruse, R., Klawonn, F. (1997), "Foundations of Neuro-Fuzzy Systems," John Wiley & Sons Ltd., NY.

67. Oriqat, Y. (2007), 'Modeling Techniques Applied to Short Essay Auto-Grading Problem', M.Sc. Thesis, Al-Quds University, pp. 46-47.

68. Ouyang, C., Lee, S. (2000), A hybrid algorithm for structure identification of neuro-fuzzy modeling, Proceedings of the IEEE International Conference on Systems and Machine Cybernetics, Nashville, Tennessee, pp. 3611–3616, 8–11 October, 2000.

69. Papadakis, S. (1998), A novel approach to short-term load forecasting using fuzzy neural network, IEEE Transactions on Power Systems, Vol. [13], No. (2), pp. 480–492.

70. Papalexopoulos, A., Hao, S., Peng, T. (1994) ‘An implementation of a neural network based load forecasting model for the EMS’, IEEE Transactions on Power Systems, Vol. [9], pp. 1956 – 1962.

71. Papalexopoulos, A., Hesterberg, T. (1990), ‘A Regression Based Approach to Short Term Load Forecasting’, IEEE Transactions on Power Systems, Vol. [5], No.(1), pp. 40 – 45.

72. Parker, D. (1982), Learning-logic, Invention Report S81-64, Office of Technology Licensing, Standford University.

73. Park, D., El-Sharkawi, M., Marks, R. , Atlas, A., and Damborg, M. (1990), Electrical Load Forecasting Using Artificial Neural Network to Power System Operation, Proceedings of the 1990 IEEE International Symposium on Circuits and Systems, New Orleans, Louisiana, May 1-3, pp. 2933-2937.

74. Peng, T., Hubele, N., Karady, G. (1990), Conceptual Approach to the Application of Neural Network for Short-Term Load Forecasting, Proceedings of the 1990 IEEE International Symposium on Circuits and Systems, New Orleans, Louisiana, May 1-3, pp. 2942-2945.

75. Peng, M., Hubele, N., Karady, G. (1992), ‘Advancement in the application of neural networks for short-term load forecasting’, IEEE Transactions on Power Systems, Vol.[7], pp. 250 - 257.

76. Pyle, D. (1999), Data Preparation for Data Mining, Morgan Kaufmann Publishers, Los Altos, CA.

77. Rahman, S. (1990), Formulation and Analysis of a Rule-Based Short-Term Load Forecasting Algorithm. Proceedings of the IEEE, Vol.[78], pp. 805–816.

78. Rahman, S., Bhatnagar, R. (1988), An expert system based algorithm for short term load forecast, IEEE Transactions on Power Systems, Vol. [3], pp. 392-399.

79. Rahman, S., Hazim, O. (1996), ‘Load forecasting for multiple sites: development of an expert system-based technique’, Electric Power Systems Research, Vol. [39], pp. 161 – 169.

80. Rojas, I., Pomares, H., Ortega, J., Prieto, A. (2000), A self-organized fuzzy system generation from training examples, IEEE Transactions on Fuzzy Systems, Vol. [8], 23–36.

81. Rong, L., Wang, Z. (1996), An algorithm of extracting fuzzy rules directly from numerical examples by using FNN, Proceedings of the IEEE International Conference on Systems and Machine Cybernetics, Beijing, China, pp. 1067–1072, 14–17 October, 1996.

82. Rumelhart, D., Hinton, G., Williams, R. (1986), Learning Internal Representations by Error Propagation, Parallel Distributed Processing - Explorations in the Microstructure of Cognition, Vol. [1], Chapter 8, pp. 318-364, The MIT Press: Cambridge.

83. Saini, L., Soni, M. (2002), Artificial neural network-based peak load forecasting using conjugate gradient methods, IEEE Transactions on Power Systems, Vol. [17], No. (3), pp. 907-912.

84. Satish, B., Swarup, K., Srinivas, S., Hanumantha, A. (2004), Effect of Temperature on Short Term Load Forecasting using an Integrated ANN, Electric Power Systems Research, Vol. [72], pp. 95-101.

85. Sugeno, M. (1985), Industrial applications of fuzzy control, Elsevier Science Inc. Pub. Co., NY, USA.

86. Sugeno, M., Takagi, T. (1985), Fuzzy identification of systems and its application to modeling and control, IEEE Transactions on Systems and Machine Cybernetics, Vol. [15], pp.116–132.

87. Tamimi, M., Egbert, R. (2000), Short term electric load forecasting via fuzzy neural collaboration, Electric Power Systems Research, Vol. [56], pp. 243–248.

88. Thomas, R. (1993), New Product Development: Managing and Forecasting for Strategic Success, Chapter One, pp. 3-26, John Wiley & Sons Ltd.

89. Tsoukalas, L., Uhrig, R. (1997), "Fuzzy and Neural Approaches in Engineering," John Wiley & Sons, Inc., NY.

90. Villalba, S., Bel, C. (2000), 'Hybrid demand model for load estimation and short-term load forecasting in distribution electrical systems', IEEE Transactions on Power Delivery, Vol. [15], No.(2), pp. 764 – 769.

91. Werbos, P. (1974), Beyond regression: New tools for prediction and analysis in the behavioral sciences, PhD Thesis, Harvard University.

92. White, P. (1979), Planning of Urban Transport Systems in the Soviet Union: A Policy Analysis, Transportation Research Part A, Vol. [13A], pp. 231-240.

93. Yang, H., Huang, C. (1998), 'A new short-term load forecasting approach using self-organizing fuzzy ARMAX models', IEEE Transactions on Power Systems, Vol. [13], pp. 217 – 225.

94. Yang, J. (2006), Power System Short-term Load Forecasting, PhD Thesis, pp. 8-9.

95. Yang, J., Cheng, H. (2004), 'Application of SVM to power system short-term load forecast', Power System Automation Equipment China, Vol. [24], No.(4), pp. 30 – 32.

96. Yu, I., Kim, C., Song, Y. (2000), A Novel Short-Term Load Forecasting Technique Using Wavelet Transform Analysis, Electric Machines and Power Systems, Vol. [28], pp. 537–549.

97. Yu, R., The, H., Jaques, P., and Froines, J. (2004), Quality control of semi-continuous mobility size-fractionated particle number concentration data, Atmospheric Environment, Vol. [38], pp. 3341-3348, http://www.mathworks.com/matlabcentral/fileexchange/11106.

98. Zadeh, L. (1973), Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems and Machine Cybernetics, Vol.[3], pp. 28–44.

99. Zadeh, L. (1994), Fuzzy logic, neural network and soft computing, Communication of the ACM, Vol.[37], pp. 77-84.

100. Zadeh, L. (1997), Toward a theory of information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Set and systems, Vol. [90], pp.111-127.

Appendix A: Matlab Codes and Functions

The Developed Matlab File for Evaluating the Developed Models

```
clc                          %To Clear the Command Window

load Trn_out_Jul_in.dat;
load Trn_out_Jul_out.dat;    %To load the training datasets
load Ts_out_Jul_in.dat;
load Ts_out_Jul_out.dat;     %To load the Testing Datasets

InputTrn=Trn_out_Jul_in;
ActTrn=Trn_out_Jul_out;

InputTest=Trn_out_Jul_in;
ActTs=Trn_out_Jul_out;



%To read the Sugeno FIS model with hybrid optimization
fismat=readfis('Jul_sughy_out_1277_new');

PredTrn=evalfis(InputTrn,fismat);
%To genarate the Predicted Loads for the training datasets

PredTs=evalfis(InputTest,fismat);
%To genarate the Predicted Loads for the testing datasets

[corrTr,mapeTr,rmseTr]= eval_CorrMape(ActTrn,PredTrn);
%To Measure the errors for the training data

[corrTs,mapeTs,rmseTs]= eval_CorrMape(ActTs,PredTs);
%To Measure the errors for the testing data

err1=[1.0 corrTr mapeTr rmseTr corrTs mapeTs rmseTs];
%To store the measures


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


fismat=readfis('Jul_sugbp_out_1277_new');
%To read the Sugeno FIS model with Back-Prop optimization

PredTrn=evalfis(InputTrn,fismat);
PredTs=evalfis(InputTest,fismat);

[corrTr,mapeTr,rmseTr]= eval_CorrMape(ActTrn,PredTrn);
[corrTs,mapeTs,rmseTs]= eval_CorrMape(ActTs,PredTs);

err2=[2.0 corrTr mapeTr rmseTr corrTs mapeTs rmseTs];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fismat=readfis('Jul_sub_out_10_new');
%To read the Sugeno FIS model with Subtractive Clustering

PredTrn=evalfis(InputTrn,fismat);
PredTs=evalfis(InputTest,fismat);

[corrTr,mapeTr,rmseTr]= eval_CorrMape(ActTrn,PredTrn);
[corrTs,mapeTs,rmseTs]= eval_CorrMape(ActTs,PredTs);

err3=[3.0 corrTr mapeTr rmseTr corrTs mapeTs rmseTs];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fismat=readfis('Jul_subsug_out_10_new');
%To read the Sugeno FIS model with Sub-Clust and Hybrid Optimization

PredTrn=evalfis(InputTrn,fismat);
PredTs=evalfis(InputTest,fismat);

[corrTr,mapeTr,rmseTr]= eval_CorrMape(ActTrn,PredTrn);
[corrTs,mapeTs,rmseTs]= eval_CorrMape(ActTs,PredTs);

err4=[4.0 corrTr mapeTr rmseTr corrTs mapeTs rmseTs];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

errors=[err1;err2;err3;err4];

%To Output the correlation and the error measures
CorrTr=errors(:,2)
MapeTr=errors(:,3)
RmseTr=errors(:,4)
CorrTs=errors(:,5)
MapeTs=errors(:,6)
RmseTs=errors(:,7)
```

The Matlab Function of Calculating the Correlation and the Error Measures

```
%Calculate  and returns the Correlation, MAPE and the RMSE measures between
the Actual (ActD) and Predicted (PredD) Power Loads
function[corrxy,mapexy,rmseyx]=eval_CorrMape(ActD,PredD);

yi=ActD;    %To Load the actual power loads
xi=PredD;   %To Load the predicted power loads
[m n]=size(xi);
[o p]=size(yi);
if(m~=o)|(n~=p)   %To check if the predicted and actual loads have the
                  same size
    error('Different Size of Actual and Predicted Loads');
    end;
MAPE=0;
Nim=0;
Din=0;
y=mean(yi);       %To calculate the mean of the actual loads
for i=1:m
    ni=(yi(i)-xi(i))^2;
    di=(yi(i)-y)^2;
    Nim=Nim+ni;
    Din=Din+di;
    merr1=(yi(i)-xi(i));
    if yi(i)==0         %To avoid division by zero
        merr=merr1;
    else
        merr=merr1/yi(i);
    end;
    MAPE=MAPE+abs(merr);
end
corrxy=sqrt(1-(Nim/Din));
mapexy=((MAPE)/m);
rmseyx=sqrt(Nim)/(m-1);
end
```

Removing the Outliers Matlab Program (Remove_Outliers)

```
% Remove_Outliers Matlab code has been used to detect and remove the
outliers from the datasets.

clc;
alpha=3;
outs_num=25;
% The maximum number of outliers to be removed

load 'l0001.dat';
% To load the datasets

y1=l0001;

index=outlier1(y1(:,6),alpha,outs_num);
% To detect the position of the outliers

[n m]=size(y1(:,6));

y1(index,:)=[];
% To delete the detected outliers

[n m]=size(y1(:,6));
```

Outliers Detecting Algorithm

```
%
% Detection and Removal of Outliers in Data Sets
% ( Rosner's many-outlier test)
%
% index = outlier( y, crit, k )
%
% where index = indices of outliers in the data
% y = data set (should be stationary)
% crit = detection criterion (default 2)
% k = number of outliers to be detected
%
% Originally written by Bob Newell, February 1996
% Modified by Jaco de Groot, May 2006
% Bob Newell used a fixed value for lambda. This script calculates the
% critical values for lambda based on the equations in
% "Quality control of semi-continuous mobility size-fractionated particle
% number concentration data", Atmospheric Environment 38 (2004) 3341?3348,
% Rong Chun Yu,*, Hee Wen Teh, Peter A. Jaques, Constantinos Sioutas,
% John R. Froines)
%-----------------------------------------------------
%
function index = outlier( y, alpha, k)
%
y = y(:);
n = length( y );
if nargin < 2
alpha = 0.05;
else
if nargin < 3, k = 1; end
end
R = zeros( k+1, 1 );
% sort deviations from the mean
ybar = mean( y );
[ ys, is ] = sort( abs( y - ybar ));
% calculate statistics for up to k outliers
for i = 0:k,
yy = ys(1:n-i);
R(i+1) = abs( yy(n-i) - mean(yy) ) / std(yy);
end;
% statistical test to find outliers
index1 = [];
imax=0;
for i = 1:k
%
pcrit=1-(alpha/((2*(n-i+1))));
t=tinv(pcrit, n-i-1);
lambda(i)=(n-i)*t./sqrt(((n-i-1+t^2)*(n-i+1)));
%
if R(i) > lambda
index=is(n-i+1:end);
index1 = [ index1 is(n-i+1) ];
end
end
% report results
```

```
if exist('index', 'var')
disp(' '), disp( [ 'Outliers detected = ' num2str( length( index ) ) ] )
disp(' '), disp( 'Outlier indices are:' )
disp( index )
else
disp(' '), disp( 'No outlier is detected!' ), disp(' ')
end
%----------------------------------------------------
% the end
```

Cross Validation Algorithm

```
clc;
load alldataset.dat; % To load the available datasets
allset=alldataset;

s=size(allset);
n=s(1);
col=s(2);
trainlength=round(n-(n/4)+3);
testlength=round((n/4)+3);

% To initiate two matrices the first one for the training datasets and the
second one for the testing datasets

trainset=zeros(trainlength,col);
testset=zeros(testlength,col);
tsindex=1;
trindex=1;

for i=1:n   % For statement to devide the available datasets
    if (mod(i,4)==0)
       testset(tsindex,:)=allset(i,:);
       tsindex=tsindex+1;
    else
       trainset(trindex,:)=allset(i,:);
       trindex=trindex+1;
    end

end


trdates=trainset(:,1);
trtimes=trainset(:,2);
trhis=trainset(:,3);
trlows=trainset(:,4);
travgs=trainset(:,5);
trloads=trainset(:,6);

tsdates=testset(:,1);
tstimes=testset(:,2);
tshis=testset(:,3);
tslows=testset(:,4);
tsavgs=testset(:,5);
tsloads=testset(:,6);
```

GUI: The Function of Loading the Input Datasets File

```
function LoadFile_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;

col=3;
[filename,pathname]=uigetfile('*.dat','please select the input file');
cd(pathname);
fid=fopen(filename,'r');
a = fscanf(fid,'%g %g',[col inf]); .
a = a';

Nbr2=num2str(a(:,1));
set(handles.listbox1,'string',Nbr2);
EmptyList=[];

% To clear the models Predicted and Actual Loads lists for the new loaded
% datasets
set(handles.listbox2,'String',EmptyList);
set(handles.actlist,'String',EmptyList);

%To load the Time, High Temperature and Low Temperature
Inputs=[a(:,1) a(:,2) a(:,3)];

% Initiating the Models List Menu
set(handles.ModelMenu,'String',{'Sug With Hybrid ','Sug With BPro','Sub
Clustering','Sub With Hybrid '});

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

handles.Input=Inputs;

guidata(hObject,handles);

status=fclose('all');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

GUI: The Function of Choosing Which Model to Use in the Prediction

```
function ModelMenu_Callback(hObject, eventdata, handles)
% hObject    handle to ModelMenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns ModelMenu contents as
% cell array contents{get(hObject,'Value')} returns selected item from
ModelMenu

Input=handles.Input;
choice=handles.modeltype; % To import the optimization technique type

val = get(hObject,'Value'); % To import your month/season choice

%****************** Spring Prediction Models *********************

if (choice==1)
switch val
case 1
    fismat=readfis('Spring_sughy_out_1277_NoMar');
    PredTrn1=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn1);
    handles.pred=PredTrn1;

case 2
     fismat=readfis('Spring_sugbp_out_1277_NoMar');
     PredTrn2=round(evalfis(Input,fismat));
     set(handles.listbox2,'String',PredTrn2);
     handles.pred=PredTrn2;
case 3
    fismat=readfis('Spring_sub_out_10_NoMar');
    PredTrn3=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn3);
    handles.pred=PredTrn3;

case 4
    fismat=readfis('Spring_subsug_out_10_NoMar');
    PredTrn4=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn4);
    handles.pred=PredTrn4;

end

elseif (choice==2)
%****************** Summer Prediction Models *********************
switch val
case 1
    fismat=readfis('Summer_sughy_out_1277_new');
    PredTrn1=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn1);
    handles.pred=PredTrn1;

case 2
```

```
     fismat=readfis('Summer_sugbp_out_1277_new');
     PredTrn2=round(evalfis(Input,fismat));
     set(handles.listbox2,'String',PredTrn2);
     handles.pred=PredTrn2;
case 3
    fismat=readfis('Summer_sub_out_10_new');
    PredTrn3=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn3);
    handles.pred=PredTrn3;

case 4
    fismat=readfis('Summer_subsug_out_10_new');
    PredTrn4=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn4);
    handles.pred=PredTrn4;

end

%****************** May Prediction **********************
elseif (choice==3)
switch val
case 1
    fismat=readfis('May_sughy_out_1277');
    PredTrn1=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn1);
    handles.pred=PredTrn1;
    %handles.pred1=PredTrn1;

case 2
     fismat=readfis('May_sugbp_out_1277');
     PredTrn2=round(evalfis(Input,fismat));
     set(handles.listbox2,'String',PredTrn2);
     handles.pred=PredTrn2;
case 3
    fismat=readfis('May_sub_out_10');
    PredTrn3=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn3);
    handles.pred=PredTrn3;

case 4
    fismat=readfis('May_subsug_out_10');
    PredTrn4=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn4);
    handles.pred=PredTrn4;

end

%****************** July Prediction **********************
elseif (choice==4)
switch val
case 1
    fismat=readfis('jul_sughy_out_1277_new');
    PredTrn1=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn1);
    handles.pred=PredTrn1;
```

```
    %handles.pred1=PredTrn1;

case 2
     fismat=readfis('jul_sugbp_out_1277_new');
     PredTrn2=round(evalfis(Input,fismat));
     set(handles.listbox2,'String',PredTrn2);
     handles.pred=PredTrn2;
case 3
    fismat=readfis('jul_sub_out_10_new');
    PredTrn3=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn3);
    handles.pred=PredTrn3;

case 4
    fismat=readfis('jul_subsug_out_10_new');
    PredTrn4=round(evalfis(Input,fismat));
    set(handles.listbox2,'String',PredTrn4);
    handles.pred=PredTrn4;

end



end


PredTrn=handles.pred;
cla;
hold on
%grid

plot(handles.axes1,PredTrn,'k-'); % To plot the predicted loads
hold off
guidata(hObject,handles);
```

GUI: The Function of Loading the Actual Loads

```
function loadact_Callback(hObject, eventdata, handles)
% hObject    handle to loadact (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename,pathname]=uigetfile('*.dat','please select Dat file');
cd(pathname);
finp=fopen(filename,'r');
x=1;
while finp > 2
    Nbr{x}=fgetl(finp);
        if (Nbr{x}==-1)
            Nbr(x)=[];
        break
        end
    x=x+1;
end
Nbr2=char(Nbr);
set(handles.actlist,'String',Nbr2);
EmptyList=[];


ActTrn=str2num(Nbr2);


handles.ActTrn=ActTrn;

guidata(hObject,handles);

status=fclose('all');
```

GUI: The Function of Plotting the Actual Loads

```
function plotactual_Callback(hObject, eventdata, handles)
% hObject    handle to plotactual (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ActTrn=handles.ActTrn;
plot(handles.axes1,ActTrn,'k-');
```

GUI: The Function of Calculating the Error Measures

```
function calcerror_Callback(hObject, eventdata, handles)
% hObject    handle to calcerror (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
PredTrn1=handles.pred;
ActTrn=handles.ActTrn;
[corrTr,mapeTr,rmseTr]= eval_CorrMape(ActTrn,PredTrn1);
set(handles.Corr, 'string', corrTr);
set(handles.Mape, 'string', mapeTr);
set(handles.Rmse, 'string', rmseTr);
```

Appendix B: Glossary

ANFIS	Adaptive Neuro Fussy Inference System
ANN	Artificial neural networks
Back-Prop	Back-propagation
EPLFS	Electric power load forecasting system
FES	Fuzzy expert system
FIS	Fuzzy Inference System
FIS	Fuzzy inference system
FL	Fuzzy logic
GA	Genetic Algorithms
GD	Gradient Descent
GUI	Graphical User Interface
HL	Hybrid learning
HT	High temperature
JDECO	Jerusalem District Electricity Company
KVA	Kilo Volt Ampere
LF	Load Forecasting
LSQ	Least-square optimization algorithm
LT	Low temperature
LTLF	Long term load forecasting
MAPE	Mean absolute percentage error
MBPN	Multilayer back-propagation network
MISO	Multiple input single output
MTLF	Medium term load forecasting
NN	Neural networks
PL	Power loads
PMO	Palestinian Meteorology Office
RMSE	Root mean square error
SISO	Single input single output
STLF	Short term load forecasting
Tr	Training datasets
TS	Takagi Sugeno
Ts	Testing Datasets

التنبؤ بمقدار حمل الطاقة الكهربائية المطلوب على المدى القصير

إعداد: رائد يوسف بصبوص

إشراف: د. لبيب عرفه

ملخص

يعتبر التنبؤ بحمل الطاقة الكهربائية المطلوب على المدى القصير جزءا هاما من عملية توليد الطاقة. يتوافر ولسنوات عدة طرق تقليدية تستخدم لحل المشكلة تشمل المتسلسلات الزمنية، الإنحدار الخطي المتعدد ، التمهيد الأسي العام، ... إلخ. لكن وفي الآونة الأخيرة ظهرت أساليب وطرق جديدة تعتمد على الذكاء الاصطناعي لتنافس الأساليب القديمة. ومن أجل متابعة آخر التطورات وللحصول على أنظمة حديثة كان هدف الباحث تقديم بحث يتعلق بالتنبؤ المطلوب بمقدار الطاقة الكهربائية على المدى القصير والتي سيتم استهلاكها في فلسطين باستخدام نظام الاستدلال الضبابي (Fuzzy Inference System, FIS)، ولهذا الغرض تم اقتراح عدة أساليب للتنبؤ بأحمال منطقة بيرنبالا كعينة على مدى يوم واحد وأسبوع واحد للأمام.

هدف الباحث في هذه الدراسة للتعرض إلى أنظمة التنبؤ بحمل الطاقة الكهربائية الحالية، دراسة المؤثرات على حمل الطاقة، مقارنة نتائج الأنظمة المختلفة، إيجاد طرق ونماذج تستطيع محاكاة الإنسان من حيث التفكير والاستنباط، حيث قام ببناء نوعين من النماذج، نموذج ذو مدخل واحد ومخرج واحد (سيسو) (SISO)، ونموذج ذو عدة مداخل ومخرج واحد (ميسو) (MISO) وتمت مقارنتهما في تمثيل مشكلة التنبؤ بحمل الطاقة الكهربائية من خلال تطبيق التقنيات الضبابية (Fuzzy) مثل الأنظمة الإستنتاجية العصبية الضبابية القابلة للتكييف (ANFIS) والتي تعتمد على طرق التعلم في الشبكات العصبية الصناعية. وقد قام الباحث ببناء وفحص النماذج باستخدام بيانات حقيقية سابقة من السنوات 2006م و 2007م تم تزويده بها من قبل شركة كهرباء محافظة القدس (JDECO) ودائرة الأرصاد الجوية الفلسطينية (PMO). وتحتوي البيانات المقدمة من قبل شركة كهرباء محافظة القدس على الوقت ومقدار حمل الطاقة المستهلك في ذلك الوقت. بينما تحتوي البيانات المقدمة من قبل دائرة الأرصاد الفلسطينية على قياسات معاملات الجو لأيام السنة وتشمل هذه المعاملات، درجة الحرارة العليا، درجة الحرارة الدنيا، معدل درجة الحرارة، نسبة الرطوبة، سرعة الرياح.

قام الباحث قبل بناء النماذج المقترحة بعملية معالجة لهذه البيانات تشمل تحديد البيانات غير الدقيقة وتلك التي تقع بعيدا عن نطاق المعدل العام (Outliers) وإزالتها، إعادة تنسيق لمتغير الوقت ليصبح عبارة عن رقم حقيقي مع كسور عشرية، تقسيم البيانات الحقيقية المتوفرة إلى قسمين بنسبة 3:1 بواسطة أسلوب التدقيق المتبادل (Cross Validation). فقد استخدم الباحث ما نسبته 75% من البيانات المتوفرة في عملية بناء النماذج، والنسبة المتبقية 25% استخدمها لفحص كفاءة هذه النماذج.

في عملية بناء نماذج "سيسو" اعتمد الباحث متغير الوقت كمدخل وحيد للنموذج، أما في نماذج "ميسو" فقد تم اعتمد متغير الوقت، درجة الحرارة العليا، درجة الحرارة الدنيا كمدخلات للنموذج، واعتمد حمل الطاقة الكهربائية كمخرج وحيد لكلا النوعين من النماذج. ولقد قام الباحث ببناء نماذج مختلفة من كلا النوعين "سيسو" و "ميسو" مثل نماذج "سوجينو" (Sugeno) بتقنيات مختلفة تشمل نموذج "سوجينو" بواسطة تقنية التحسين الهجينة أو المركبة (Hybrid)، تقنية التحسين بالتغذية الراجعة (Back-Propagation)، نموذج "سوجينو" باستخدام تقنية التكتل الطرحي (Subtractive Clustering)، نموذج "سوجينو" باستخدام الطريقة المتعاقبة (تقنية التكتل الطرحي ثم تحسين النموذج باستخدام تقنية التحسين الهجينة).

للتأكد من مدى ملاءمة وصلاحية النماذج المقترحة وقياسها. استخدم الباحث معامل درجة التوافق (Correlation Coefficient) لقياس مدى تطابق أحمال الطاقة الكهربائية الحقيقية مع المتنبأة، بالإضافة لذلك اعتمدت طريقتان لحساب قيمة الخطأ وهما متوسط نسبة القيمة المطلقة "ميب" (MAPE)، ومتوسط مربع الجذر "آرمسي" (RMSE).

وقد أشارت النتائج الأولية التي حصل عليها الباحث بأن نموذج "ميسو" الذي بني باستخدام الطريقة المتعاقبة قد حصل على أعلى مستوى مقارنة مع النماذج الأخرى، حيث كانت قيم معامل درجة التوافق تتراوح ما بين (0.95-0.98) وبمعدل يساوي 0.97، وكانت قيم "ميب" تتراوح ما بين (0.02-0.06) وبمعدل 0.03، بينما كانت قيم "آرمسي" تتراوح ما بين (0.07-0.11) وبمعدل يساوي 0.08. ودمجت النماذج المقترحة التي تم بناؤها في تطبيق قائم بذاته ذي واجهة رسومية للمستخدم ليمثل نظاماً للتنبؤ بمقدار حمل الطاقة الكهربائية المطلوب على المدى القصير. يستطيع المستخدم عن طريق هذا التطبيق (Electric Power Load Forecasting System, EPLFS) أن يقوم بالتنبؤ بمقدار حمل الطاقة المطلوب ليوم واحد أو أسبوع واحد للأمام مع رسم بياني للأحمال المتنبأ بها وإمكانية مقارنتها بالأحمال الحقيقية إن توفرت لقياس مدى كفاءة النظام.

في النهاية قام الباحث بفحص كفاءة النظام (EPLFS) باستخدام بيانات حقيقية من العام 2008 تم تزويده بها حديثا من قبل شركة كهرباء محافظة القدس ودائرة الأرصاد الجوية الفلسطينية، حيث قام الباحث باستخدام هذا النظام للتنبؤ بأحمال الطاقة الكهربائية ومقارنتها بالقيم الحقيقية لفترة يوم واحد واسبوع واحد للأمام، فكانت قيم معامل درجة التوافق تتراوح بين (0.87-0.97) وبمعدل مقداره 0.94 لجميع الأنظمة في حالة التنبؤ لفترة يوم واحد للأمام. بينما عند التنبؤ لفترة أسبوع واحد للأمام فقد كانت قيم معامل درجة التوافق تتراوح بين (0.90-0.95) وبمعدل يساوي 0.93 لجميع النماذج المقترحة.

تدل هذه النتائج الأولية جميعها على مدى ملاءمة وصلاحية وكفاءة النماذج المقترحة التي تم بناؤها لحل مشكلة التنبؤ بمقدار حمل الطاقة الكهربائية المطلوب على المدى القصير باستخدام متغيرات الوقت والطقس. ويقترح الباحث مزيدا من التحسين على هذه النماذج عن طريق جمع بيانات حقيقية أكثر، الأخذ بعين الاعتبار متغيرات الطقس الأخرى والتي تشمل (معدل درجة الحرارة للأيام السابقة، نسبة الرطوبة، سرعة الرياح،...إلخ)، التغير في نسبة السكان، استخدام تقنيات مختلفة لتحسين النماذج مثل تقنية الخوارزميات الجينية (Genetic Algorithms).