2013

Load Estimation For Electric Power Distribution Networks

Chiebuka Eyisi
University of Central Florida

LOAD ESTIMATION FOR ELECTRIC POWER DISTRIBUTION NETWORKS

by

EYISI CHIEBUKA V.P
B.S. Kwame Nkrumah University of Science and Technology, 2010

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Electrical Engineering
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2013

Major Professor: Saeed Lotfifard

ABSTRACT

In electric power distribution systems, the major determinant in electricity supply strategy is the quantity of demand. Customers need to be accurately represented using updated nodal load information as a requirement for efficient control and operation of the distribution network. In Distribution Load Estimation (DLE), two major categories of data are utilized: historical data and direct real-time measured data. In this thesis, a comprehensive survey on the state-of-the-art methods for estimating loads in distribution networks is presented. Then, a novel method for representing historical data in the form of Representative Load Curves (RLCs) for use in real-time DLE is also described. Adaptive Neuro-Fuzzy Inference Systems (ANFIS) is used in this regard to determine RLCs. An RLC is a curve that represents the behavior of the load during a specified time span; typically daily, weekly or monthly based on historical data. Although RLCs provide insight about the variation of load, it is not accurate enough for estimating real-time load. This therefore, should be used along with real-time measurements to estimate the load more accurately. It is notable that more accurate RLCs lead to better real-time load estimation in distribution networks.

This thesis addresses the need to obtain accurate RLCs to assist in the decision-making process pertaining to Radial Distribution Networks (RDNs).This thesis proposes a method based on Adaptive Neuro-Fuzzy Inference Systems (ANFIS) architecture to estimate the RLCs for Distribution Networks. The performance of the method is demonstrated and simulated, on a test 11kV Radial Distribution Network using the MATLAB software. The Mean Absolute Percent Error (MAPE) criterion is used to justify the accuracy of the RLCs.

I would like to dedicate this thesis to my family. In full gratitude to my amazing parents, Philip and Ifeyinwa for always encouraging me to believe in myself and keep striving for the best. Also to my siblings, Ogochukwu, Uzochukwu and Akunna for all the love and care we shared while growing up together. I also give special thanks to my late paternal grand-father for his advice and words of wisdom. He always said: Success is 99% perspiration and 1% inspiration. And special thanks to all my friends and extended family for their care and support.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Saeed Lotfifard for his support, direction and encouragement for this work. Without his tutelage and constant motivation, this work would not have been possible. Most especially, I'd like to thank him for his patience throughout the entire process. I've really learned a lot through him to help me in my pursuit of being an academic. I would also like to thank Dr. Jiann.-S. Yuan and Dr. Thomas Wu for serving on my committee.

Special thanks also to my school division for allowing me conduct this research. And also to my friends and fellow students in the College of Engineering and Computer Science for different ways we've all being able to mutually assist each other.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS (or) ABBREVIATIONS

AMR	–	Automated Meter Reading
ANFIS	–	Adaptive Neuro-Fuzzy Inference Systems
ANN	–	Artificial Neural Networks
CBR	–	Case Based Reasoning
DA	–	Distribution Automation
DLE	–	Distribution Load Estimation
DMS	–	Distribution Management System
DNMF	–	Daily Nodal Model Factor
DSE	–	Distribution State Estimation
DSM	–	Demand Side Management
DSMF	–	Daily System Model Factor
DSO	–	Distribution System Operator
FKM	–	Fuzzy K-Means
FL	–	Fuzzy Logic
FS/FIS	–	Fuzzy Inference Systems
FTU	–	Feeder Terminal Unit
GWh	–	Giga-Watt-hour
HC	–	Hierarchical Clustering
kVA	–	kilo-Volt-Ampere
kW	–	kilo-Watt
kWh	–	kilo-Watt-hour

LF	–	Load Forecasting
MAPE	–	Mean Absolute Percent Error
MDA	–	Multiple Discriminant Analysis
MF	–	Membership Function
MSE	–	Mean Squared Error
MW	–	Mega-Watt
PF	–	Power Flow
RDNs	–	Radial Distribution Networks
RLCs	–	Representative Load Curves
RMSE	–	Root Mean Squared Error
RTD	–	Real Time Demand
RTO	–	Regional Transmission Organization
SCADA	–	Supervisory Control and Data Acquisition
WLAV	–	Weighted Least Absolute Value
WLS	–	Weighted Least-Squares
WNMF	–	Weekly Nodal Model Factor
WSMF	–	Weekly System Model Factor

CHAPTER ONE: INTRODUCTION

In any modern household, the primary source of energy consumption is undoubtedly electrical energy. In the electricity industry, the major determinant in electricity supply strategy is the quantity of demand. Another key element is the quality of obtained load data from measurements. As mentioned earlier, for Distribution Load Estimation (DLE), two major categories of data are utilized: historical data and direct real-time measured data. Due to its cost effectiveness, real-time measured data are usually more difficult to obtain compared to the availability of historical data. Distribution Load Estimation (DLE) differs somewhat from Load Forecasting (LF) in the sense that LF is usually done on a time series analysis with the goal of predicting loads, days and/or weeks in advance; whereas DLE involves studying the network topology and its current parameters (load data and line data) to obtain nodal kW consumption and/or overall system kW magnitudes using sometimes limited measurement data.

An RLC is a curve that represents the behavior of the load during a specified time span; typically daily, weekly or monthly, based on historical data. Although RLCs provide insight about the variation of load, it is not accurate enough for estimating real-time load. This therefore, should be used along with real-time measurements to estimate the load more accurately. It is notable that more accurate RLCs lead to better real-time load estimation in distribution networks. An RLC basically represents a group of load curves exhibiting similar demand patterns. Generating RLCs is what we hope to achieve in this thesis.

There has been substantial interest in identifying RLCs in applications where daily fluctuation of customer demand is an important characteristic. A brief review of literature showed how RLCs were determined using both clustering and statistical methods [1], [2]. In essence, RLCs can somewhat be used in conjunction as pseudo-measurements to estimate loads and also identify factors influencing variations in demand. Hence, if accurate enough, it's of good advantage for the DLE process. They can be built in the set of similar load curves. These RLCs can be used in distribution network calculation; Distribution Load Estimation (DLE) and Distribution State Estimation (DSE) for example.

This thesis addresses the need to estimate Representative Load Curves (RLCs) to assist in the decision-making process of Radial Distribution Networks (RDNs). A method is proposed based on Adaptive Neuro-Fuzzy Inference Systems (ANFIS) architecture to estimate the RLCs for Distribution Networks. Proposed by J.-S.R Jang in 1993 [3], ANFIS integrates the best features of Fuzzy Systems (FS) and Artificial Neural Networks (ANN) and can be used to learn information about a set of data. Further details will be explained going forward. The performance of the method is demonstrated and simulated, on a test 11kV Radial Distribution Network using the MATLAB software. The Mean Absolute Percent Error (MAPE) criterion is used to justify the accuracy of the RLCs.

1.1 Organization of Thesis

Objectively, in this thesis we aim to estimate RLCs. Available information possibly currently present describing the network include: historical data (load survey/research and billing data), customer information, load profiles and direct measurements amongst others. Transformers, distribution substations and some important metered loads are prominent sources of direct measurements.

The thesis would therefore be organized as follows: In this Chapter, we'd describe the power distribution networks briefly before shedding some light on its operation and connectivity of loads to the network.

Chapter 2 presents a comprehensive literature review on Distribution Load Estimation (DLE) including some proposed methods on estimating the aforementioned Representative Load Curves (RLCs) for distribution networks.

In Chapter 3, we'd present the test system chosen for this thesis. It's a sample 11kV, 15-node, radial distribution network. We'd further explain the source of the data used and how it was modified to suit the desired purpose of this thesis. We'd present a few computed statistical reports on the historical load data used in relation to the system. Its network parameters can be found in the appendix to aid in distribution state estimation and power flow approaches for anyone interested in performing these calculations.

Chapters 4 and 5 are where attempts are made to estimate the RLCs using Adaptive Neuro-Fuzzy Inference Systems (ANFIS). A brief overview of ANNs, FS and ANFIS is presented first, before proceeding to explain the method used. Using the MATLAB software a simple approach to estimating RLCs for the sample 15-node distribution network is presented.

Chapter 6 is where we display all obtained results. Hopefully, these RLCs present a quick snapshot or quick snapshots of how loads vary over time over a select period; and are also able to provide further information upon embarking a load estimation. The RLCs can also be used as pseudo-measurements in state estimation as well. Pseudo-measurements are generally used to augment the available real-time measurements. Various tables and graphical representations are obtained and shown as well as simulated results.

We also then conclude with chapter 7, discussing also any possible future research in determining/estimating RLCs.

1.2 Distribution Networks

1.2.1 Overview and Operation

Electric power systems are real-time energy delivery systems, implying that power is generated, transmitted and distributed to loads/consumers instantaneously. They are also one of the largest and most important life support systems in engineering. Distribution networks transport this energy from substations at distribution centers to service-entrance equipment found at residential, commercial and industrial consumer facilities. Distribution feeders are normally radially connected and are fed from one or more sub-transmission lines. The major components found in all substations include: low-side and high-side switching, voltage transformation, voltage regulation, equipment protection and metering. The major components of radial feeders include: voltage regulators, transformers, loads, voltage laterals, the primary feeder and shunt capacitor banks amongst others [4].

For efficient operation and control of power distribution networks, updated load information at each node is required to represent customers accurately. It's important that electric power distribution networks meet customer load demands at all times in a safe and efficient way. The quality of load data obtained plays a huge role in DLE.

It's true that, power systems in all three phases rarely have balanced loads, impedances, voltages and currents. Balanced three-phase power systems have all three-phase voltages and currents having the same amplitude and are phase-shifted by 120° with respect to each other. By using the techniques of symmetrical components, the analysis of unbalanced cases is greatly simplified. The neutral current in a balanced system is zero; even the removal of the neutral core would have no effect on the circuit. The superposition of 3 balanced systems, each with positive, negative or zero sequence of balanced voltages is used to analyze an unbalanced system.

For simplicity and ease of comprehension in the study of power systems, the assumption of balanced three-phase systems is what's mostly used in the analysis of most power networks. This assumption of balance simplifies the network so that a single-phase equivalent model of the network can be investigated. This assumption is propagated through this study and has also been found to be more sufficient in interconnected systems compared to distribution system analysis and modeling. This is due to a considerable number of served single-phase loads and non-equilateral conductor spacing of overhead and underground line segments amongst others [4]. An interconnected system can be but not limited to a distribution system with multiple sources of available power that loops through the network, so that service is still maintained even though one power source goes down. This implies improved reliability, stability and reduction in the overall cost of providing reserves.

Distribution systems designs include: Radial, Loop or Network. Widely used in sparsely populated areas and cheapest to build is the radial system. Probably is the least secure network too, coupled with the added advantage of fast fault localization. As the name implies, the loop system loops through the service area and returns to the original point. It's usually tied to an alternate power source and is more secure and expensive than the radial system. As complicated as they are and located in congested areas, network systems are interlocking loop systems. Provides added reliability, but it's the most expensive.

Effective operation of distribution networks are required to meet the increasing daily demand of various consumers on the network. Usually, one substation could supply many customers with power. It is widely accepted that placing meters or load monitoring devices on every feeder in the network is somewhat not economically justified why fulfilling its daily operation. Substantial research has been done in both state estimation and load estimation of distribution networks. Often times referred to as DSE and DLE respectively. Deregulated electricity markets where both generating companies and customers are active participants in ensuring maximum total social welfare and a fair market is where electric power systems are moving towards. Prominent features of distribution systems include [4], [5]:

- Distributed generation
- Radial or near radial structure
- Unbalanced distribution of loads
- Large number of branches/nodes
- Multiphase, unbalanced operation

Nodal load information and steady state analysis are amongst the basic requirements for efficient operation.

Distribution system analysis differs somewhat from transmission systems in many ways and that's why careful attention is given in their study. Distribution networks differ characteristically from transmission systems as outlined below [5]:

- Presence of distributed generators (DGs).
- Weakly meshed/radial structures.
- High resistance/reactance (R/X) ratio of the lines.

- Low voltage levels compared with those of transmission systems.
- Unbalanced networks/loads.
- Shunt capacitor banks and distribution transformers.

The power distribution system can also be described as a group of buses which are interconnected through distribution lines, switches and transformers [4]. Each bus may connect with loads, shunt capacitor banks, Distributed-Generators/Cogenerators etc. The 3 phases could be connected in delta or wye.

As we all understand that most distribution networks are typically unbalanced, this study assumes a balanced system for simplicity of computations.

1.2.2 Loads

Electrical loads vary with time and the generation and distribution of power must quickly respond to the customers' load demand at any time. Demand is load averaged over a specific period of time. The sampling time interval chosen for this study is 60-minutes (1-hour). For example, a typical 1-hour kW demand could be 250kW. Usually, the demand curve is broken into equal time intervals to define the load. The average value of the demand in each interval is what constitutes the load profile of a consumer [4]. The shorter the time interval, the more accurate the load value will be. How electric energy is used at various times (daily, weekly, monthly, and seasonally) and aggregated customers' share of the utility's total load is of major emphasis to Distribution System Operators (DSOs).

The demand for electricity constantly varies and somewhat increases with population. The modern DSO needs accurate load data for the following purposes:

- Tariff planning and pricing
- Proper operation and network planning
- Efficient management of loads and power production planning
- Customer service
- Billing
- Availability of information to the general public

To estimate some sort of RLCs from load research measurements/historical data to pictorially represent how loads vary at both nodes and substations at select periods (daily, weekly, monthly, and seasonally); is one objective we hope to achieve at the end of this study. In electric power distribution systems, the need to improve the knowledge of loads by developing improved load models from aggregated load information is endless. Important specifications of load data include its classification (residential, commercial, industrial etc.), time, and magnitude and system location.

A load profile is a graph of the variation in the end user electrical load over time. The most important factors influencing electric loads include: customer behavior, weather conditions, electrical appliances and installations, time dependencies and previous load values amongst others. Billing meters (or transformer capacity) also provide the measurements from customer loads. The annual energy data can be used, but more preferably is the annual hourly load data for determining RLCs. Sometimes, typical daily load curves in different load classes are also available. Monthly energy consumption and hired power contracts are also good data sources. All these information can be mined for the abstraction of load information. Data collection can be quite costly taking into account the time and volume of data involved, and the fact that this is done continuously.

Recently, there's also been an increase in Automated Metered Reading (AMR) and Feeder Terminal Units (FTUs) in few electric utilities to obtain enough information about the system [5]. This is also sometimes uneconomical as suggested by most utilities. AMRs provide utilities with accurate and up-to-date electricity consumption and status data. FTUs are used for the supervision, control, measurement and possibly, protection of medium voltage networks. Most utility companies also embark on load research and/or measurement campaigns to collect and analyze load data from various locations on the distribution network. This helps to implicitly characterize consumers' behavior. Rather expensive it is these days due to costs of human work and state-of-the-art metering instruments; but results of improved accuracy and more efficient energy production have been proven. The expenses incurred in gathering load data from the network are significant. Now that AMR systems are becoming a common feature, RLCs can also be estimated using actual consumption data. But that's another topic for the future.

We assume loads to be normally distributed where the parameters (*mean*, *standard deviation* and *variance*) are used to describe a random variable. For more details on distribution networks, see [4]. Composed of thousands of individual components, the system load is a random non-stationary process.

CHAPTER TWO: LITERATURE REVIEW

2.1 Distribution Load Estimation (DLE)

Accurate load estimates not only brings utilities big economic benefits by lowering the cost of operation through a high automation degree, but also improves the satisfaction of customers by increased quality of power. Load estimates are needed in advanced functions of Distribution Automation (DA). A DA as adopted from the IEEE definition, “is a combination of automation systems that enables an electric utility to monitor, coordinate and operate some or all the distribution network components in real time”. However, DLE has been found to be somewhat challenging because of the limited availability of real time measurements and the sheer number of loads [6], [7]. Traditionally, load estimates are obtained through occasional real-time measurements, monthly billing data and monthly peak load readings.

Many methods have been proposed for DLE. Some methods used customer kWh consumption, transformer kVA ratings, or monthly peak load reading to estimate loads. A few others have combined historical data with available real-time measurements to improve load estimates. Our objective; which is to estimate RLCs in this light is what we hope to achieve, replacing historical data as pseudo-measurements in DLE calculations. A short review of various methodologies developed in various literatures is presented here briefly. Methods ranging from traditional to intelligent methods are briefly discussed. These methods try to exploit various power system properties and/or data to estimate the load. In general, the only information available regarding loads, other than data from major distribution substations and equipment installations, is the billing cycle customer kWh consumption, all of which can be mined for the abstraction of load information [5].

The efficiency and effectiveness of the proposed methods depend on how well the resulted estimates match these available measurements and satisfy some relations and constraints based on power flow calculations. DLE provides static real and reactive load estimates for each system node in a power network given synchronized measurements. For accurate representation of customers in electric power distribution networks, up-to-date nodal load information is required. An overview of Distribution Load Estimation (DLE) in power distribution systems is also presented in [5]. Desirable Distribution Automation (DA) functions or Distribution Management System (DMS) applications such as service restoration, planning and Demand Side Management (DSM), depends highly on the load data. Estimates can be processed to meet the different requirements of the DMS applications if real and reactive load estimates of these networks can be provided.

Most times, DLE algorithms have two steps; the first step involving an initial guess and its adjustment of accuracy at the second step. DLE, different from Distribution State Estimation (DSE) can provide the estimates of both load and states. Often times, State Estimation (SE) tools are widely carried-out in high voltage transmission networks where a number of redundant on-line measurements and dependable communication channels are available. Dispatching the separate measures to a centralized controller, they form a fully integrated Supervisory Control and Data Acquisition (SCADA) system. Distinctive from the high voltage bulk transmission network, reliable on-line measurements and communication mediums may be at present not fully available at the distribution level.

Several approaches have been proposed to estimate loads in various literatures. In [8], [9], V.P. Borozan *et al*, and [10], Broadwater *et al*., load estimates were obtained by scaling measurements according to transformer's peak load analysis or existing actual load curves respectively. Loads were designated/allocated to individual line sections. Affecting load allocation is diversification of load groups and coincidence of peak loads. These methods are more suitable for estimating peak load. In [11], [12], the idea using DSE techniques for DLE was mentioned by Baran *et al*. as a by-product of DSE. In [13], Wang *et al*. proposed a two-step procedure combining load allocation with DSE techniques. Firstly, loads are allocated according to billing data and typical load curves. In the second step, the coarse load estimates from the first step were used as load pseudo-measurements. A Weighted Least-Squares (WLS) SE was performed with on-line measurements and load pseudo-measurements to compute real and reactive loads based on state estimates on the assumption that the network was balanced and single phase analysis was used.

Ghosh *et al*. proposed a statistical load modeling technique to express the variation of active power demand in radial networks [14]. With power flow measurements taken into consideration, their procedure divides the network into sub-trees to handle multiple measurements and provides a measure of uncertainty/ambiguity in load estimates for different classes of loads. Class-specific daily load curves with their means and deviations were obtained using statistical approaches. The mean of the load estimate at specific times were computed based on the mean of the corresponding load model factor and the average daily customer demand in the billing cycle. Most essentially, it's used for probabilistic distribution state estimation (DSE) in radial networks.

In [15], Nazarko *et al*. applied ideas of fuzzy regression to express the correlation between substation peak active loads and supplied customer active loads in radial networks.

Unreliable and inaccurate input data having been modeled by means of fuzzy numbers, and trapezoidal and triangular forms of fuzzy numbers were used for illustrating input data. They determined a regression model, expressing the correlation aforementioned existing in the substation population. An intelligent approach used for peak load estimation.

In [16], Kuo and Hsu used expert knowledge and operator experience, where fuzzy variables were used to represent linguistic descriptions for the size of loads. Based on the ratio of the sum of the rated transformer capacities of the branching point to the sum of rated capacities of transformers supplied from the feeder, the load current at the branching point was scaled down from the available feeder current. They approximated the load current at a bus as a fuzzy variable described by the membership function. This is a time of day dependent DLE technique.

In [17], Irving *et al*. proposed a Weighted Least Absolute Value (WLAV) approach to reduce the effect of gross errors in measurements. Active power flows in each branch of the network were used to define state variables, which were in turn used to express measurement functions. Through some constant coefficients, voltage information and reactive power amongst others were included in measurement equations. In [18], Falcao *et al*. applied neural and fuzzy set techniques to obtain load curves for customers' classes based on their monthly energy consumption and a large set load curves' data extracted from measurements. They applied a kohonen network and fuzzy techniques to classify customers into clusters judiciously. The range of uncertainty of the load curve was also induced and used to obtain a rough estimate of the load. They came up with a linear programming estimator to refine the load estimates to match the actual real-time measurements at initial feeder points.

A zonal load estimation that divides radial distribution networks into several zones by profiting from how the special properties of the measurement Jacobian is expressed is proposed by J. Wan *et al*. in [5, 6, 19]. They all treated loads as variables as opposed to pseudo-measurements, considering the load driven nature of distribution networks. The same authors in [7], also proposed a Weighted Least Squares (WLS) method with multiple load parameters for DLE by treating loads as variables as well. An exterior penalty method is used to reconstruct the formulated nonlinear constrained optimization problem into an unconstrained problem. Also noted in these methods was the importance of the percent accuracy of pseudo-measurements to be used to ensure system observability.

Other proposed methods not mentioned above include; a Case-Based-Reasoning (CBR) method is presented for distribution network nodal load estimation in [20] by J. Wu *et al*. and also incorporates fuzzy neural networks. A more recent DLE method using clustering techniques is presented in [21] by Grigoras *et al*. in which k-means clustering is used as part of the estimation process to obtain coarse estimates and these estimates are further refined to obtain the estimated load. Konjic *et al*. applied Fuzzy Inference Systems (FIS) to estimate substation load in [22], by aggregating individual FIS of Takagi-Sugeno type. The model was developed from actual measurements forming a base of raw data of customer information allowing one to build large tests and training sets of simulated low voltage (LV) substations, leading to the development of the fuzzy system.

The methods listed above are only a subset of many other methods developed in literature and can be readily divided into four classes: traditional, intelligent, and statistical and those that apply SE related techniques. Most methods employ pseudo-measurements. Some of which include; average daily customer demand and/or classified typical load curves. These are good candidates for DLE, but not as good enough as the RLC. This is because the RLCs are customer/node dependent and are obtained from historical data. In essence RLCs give a better indication of the customer/nodal load and is therefore a better candidate for pseudo-measurements. The figure below presents a snapshot summary of the literature review, showing 4 classifications of Distribution Load Estimation (DLE) techniques. Some proposed methods may combine two or more of these techniques:

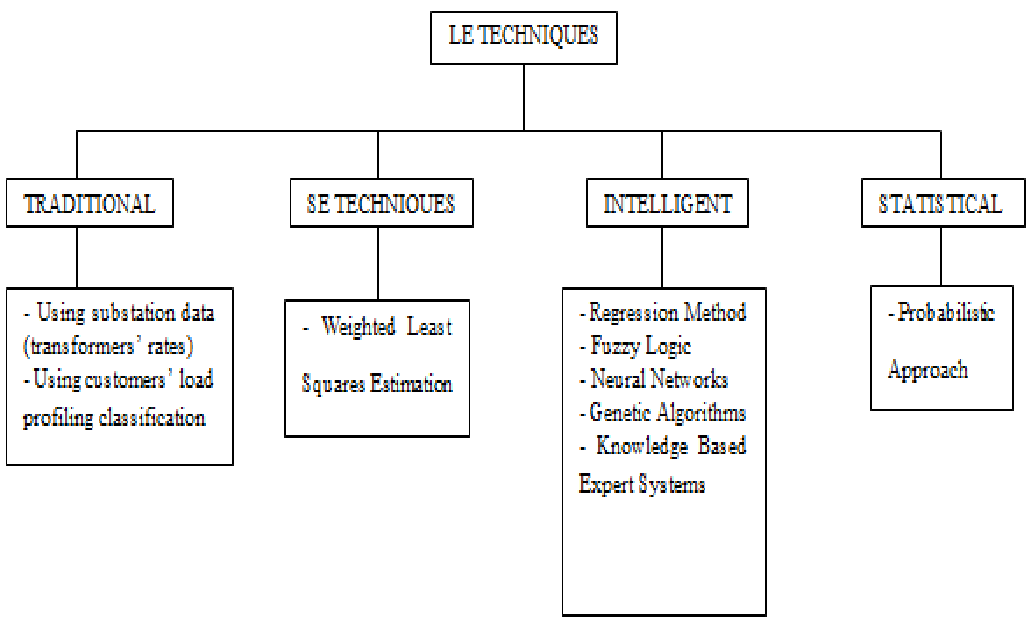

Figure 1: Classifying Load Estimation Techniques

2.1.1 Representative Load Curves

Representative Load Curves (RLCs) show typical daily, weekly or monthly load curve which represents a group of load curves exhibiting similar demand patterns. Highly accurate

Representative Load Curves (RLCs) can be used to describe electricity demand. A brief review of literature showed how RLCs were determined using both clustering and statistical methods. The demand can be expressed as a peak demand (MW), annual demand (GWh) or annual load-duration curves for electricity depending on the type of planning. Essentially, RLCs can somewhat be used to estimate loads and also identify factors influencing variations in demand.

The planning horizon for the estimated RLCs can be at least one year and encompass all variations. It is important to note that RLCs in most cases represent a means to an end, and not an end in itself. And that's why we propose this be used in conjunction with real-time measurements to achieve more accurate load estimates. The studies on building RLCs are rare, and therefore is the objective of this thesis. We proceed by reviewing the literature on this topic.

In [2], Balachandra *et al*. propose the use of Multiple Discriminant Analysis (MDA) to cluster daily load curves into a set of RLCs. MDA is a method for compressing a multivariate signal to yield a lower dimensional signal tractable to classification. They proceed to explain why it's important that RLC's capture the dynamics of demand variations if it's in any way to be used in the control and operation of distribution networks.

An order-specific clustering algorithm for the determination of representative load curves is proposed by Marton *et al*. in [1]. They proposed an algorithm that utilizes RLCs to describe the cluster it generates. Their objective is to generate clusters that represent a segment or segments of a time-ordered data set which in this case would be historical data, while preserving and accommodating daily fluctuations.

In [23], Binh *et al*. built RLCs in the set of similar demand/load curves by clustering analysis as well, on the basis of their electricity behavior. Fuzzy K-Means (FKM) is utilized in their work. Actual measurements from different feeders derived from a distribution network were used as load data in their work. They also used Bellman-Zadeh maximization principle and global criterion method to compromise the cluster validity indexes and determine the optimal cluster number. Details on determining a suitable weighting exponent was further introduced. Similarly, in [24], Hossain *et al*. also apply FKM to determine typical load profiles of consumers. Their results demonstrated how to assign typical load profiles to consumers on a test feeder efficiently.

Another clustering algorithm presented by Gerbec *et al*. for determining RLCs are presented in [25]. They apply both the hierarchic clustering algorithms and FKM to derive typical load profiles from obtained measurements with Ward distance between the clusters. They aimed at comparing the classification techniques applied for classification of measured load profiles and showing how they all generate comparable cluster results.

A review and analysis of residential electric load curve models is presented by Grandjean *et al*. in [26]. They identified two main types of load curve models; Top-Down and Bottom-Up; used in residential class-loads and proceed further in comparing the two. Applications of RLCs were also found. For example, a method using RLCs of each consumer's activity to determine expected loading in preset part of the distribution network is presented by Jardini *et al*. in [27]. RLCs being obtained from field measurements are aggregated to determine the expected loading in equipment supplying power to the network.

Fidalgo in [28], uses an innovative way, combining Kohonen clustering with the use of Artificial Neural Networks (ANN) to estimate load curves for distribution systems. The method includes three main procedures: clustering, inference of load diagrams of MV/LV public stations and finally, the estimation of error bars providing its performance measures.

It has been proven that ANFIS combines the best properties of Artificial Neural Networks (ANN) and Fuzzy Logic (FL) in estimation/curve-fitting aspects and it's therefore our desire to prove that fact remains the same in our study of estimating RLCs for distribution networks. Secondly, not much work is done with regards to ANFIS applications in estimating RLCs; another reason for this study. Finally, none of these methods were applied to data sets ranging from year to year, in order to estimate weekly/daily RLC's for each month for either the substation or the node. This is to be achieved here based on section by section, or window by window comparisons in terms of minimum absolute percent error between the estimated RLC and the historical measured data. RLCs are therefore customer/node dependent and are obtained from historical data. In essence RLCs give a better indication of the customer/nodal load and is therefore a better candidate for pseudo-measurements.

So in this thesis, we ask these questions: *How do you prioritize which subset of the historical dataset one intends to use as pseudo-measurements alongside real-time measurements for distribution system load estimation? What criterion makes that subset an ideal set of pseudo-measurements to be used in DLE?* For example; if you want to estimate the real-time load in a distribution network at say Jul. 19th, 2011 at 0300 hours and you have say three years (2008 – 2010) historical measured data for a distribution network. Unfortunately, you have only a few synchronized real-time measurements from various meters (like AMRs) placed in certain locations on the network. But you don't have any direct measurement for the particular node under study. Now do you use, Jul. 19th, 2008 at 0300 hours OR Jul. 19th, 2009 at 0300 hours OR

Jul. 19th, 2010 at 0300 hours as pseudo-measurements for the DLE process. Even if you want to, how do you prioritize your choice? Hence the reason for RLCs, to sort of integrate the properties of these three different times in the past into a single curve that represents them, characteristically taking a snapshot of the way load varies during this select period. With ANFIS, we hope to prioritize this selection and achieve this objective.

Two major categorizations exist upon reviewing various methods proposed in the estimation of RLCs;

Computational Intelligence and Artificial Intelligence Methods

- Fuzzy K-Means (FKM) Algorithm
- Hierarchical Clustering (HC) Algorithm
- Artificial Neural Network (ANN) in collaboration with Kohonen Clustering tool
- Knowledge-Based Expert Systems

Statistical Methods

- Using load survey systems according to some predefined consumers classes
- Multiple Discriminant Analysis (MDA)
- Top-Down Models and Bottom-Up Models
- Regression Models

Using the MATLAB software, we present an uncomplicated method to estimate RLCs using Adaptive-Neuro-Fuzzy Inference Systems (ANFIS).

CHAPTER THREE: DATA ANALYSIS/SURVEY

We present the sample 15-node distribution network used for this study. The source of various components of the data used is further explained and how it was modified to suit the desired purpose of this thesis. We'd present its network parameters in the appendix as well as computed statistical reports on the load data related to the system here.

3.1 Sample Data

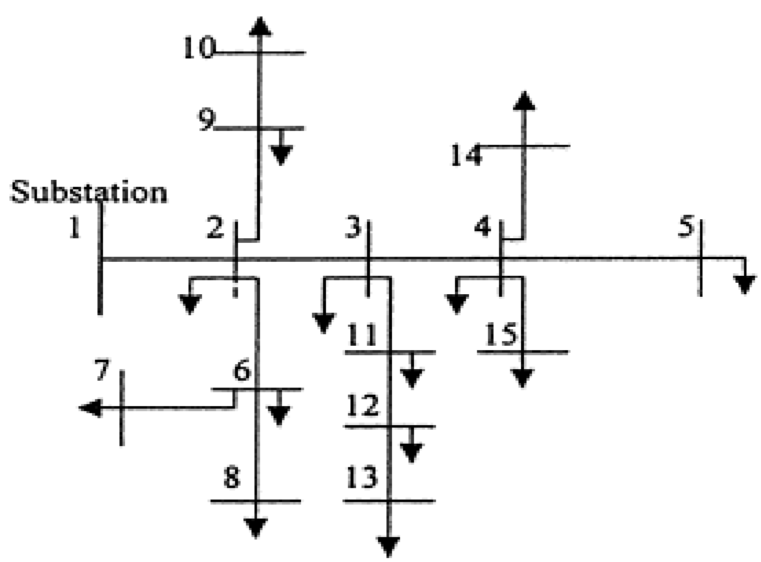

Figure 2: 11kV Test Distribution Network [29]

The figure shown above was extracted from [29] and is a single line equivalent diagram of an existing three-phase, 11kV radial distribution feeder. We assume this to be one of the feeders on a power company's distribution grid. The power distribution system can also be described as a group of buses which are interconnected through distribution lines, switches and transformers [4]. System loads are considered as constant (PQ) power or spot loads. The substation at bus 1 is considered as a slack bus with a constant voltage and is the only supply source in the system. Power factor of the load is taken as $\cos \phi = 0.70$. As is found in most cases, line shunt capacitance (different from shunt capacitor banks that are considered as loads) is considered to be negligible at the distribution voltage levels. Those are some of the assumptions made regarding the test network. Power Flow (PF) and Distribution State Estimation (DSE) can be used to find the voltage profile along the feeders. The line data for the network is presented in the appendix.

The load data used was mined and extracted data from [30] and was used for computational purposes in this thesis. Reason is because; one needs a huge data set for the ANFIS computations. Modifications were attainable on two years of historical measured data which was used for this study. The data source was primarily from PJM interconnection. PJM is a Regional Transmission Organization (RTO) that coordinates the movement of wholesale electricity in all or parts of 13 states and the District of Columbia. An RTO is responsible for moving electricity over large interstate areas, by coordinating, controlling and monitoring an electricity transmission grid that is larger with much higher voltages than a typical power company's distribution grid. Two years' data (2009, 2010) were mined and extracted from the data source to represent same two years (2009, 2010) for a sample network. Changing the historical metered readings from the data source from MW to kW and assuming this change to be effected to our test network is exactly what was done. The result is an adapted historical metered load for our sample network summarizing the kW real time demand.

The sum of individual demands of a diverse set of customers is the total demand met by the utility. The sample 11kV, 15-node distribution network is named/designed as follows: The sample 15-node distribution network is assumed to be one of the feeders on a power company's distribution grid. The substation feeds the remaining 14 nodes to be named as follows (Bus 2 – RV, Bus 3 – LAG, Bus 4 – DEL, Bus 5 – CR, Bus 6 – BAY, Bus 7 – ABJ, Bus 8 – KAT, Bus 9 – SOK, Bus 10 – IM, Bus 11 – AB, Bus 12 – KAD, Bus 13 – TAR, Bus 14 – EK, Bus 15 - OY). This is just a naming procedure and represents the loads fed from the substation. As was said earlier, all these assumptions were done to enable the ease of computation.

Some of the graphical statistics on the selected 2009 data was performed and a few of them are shown below. Reason we opt to show statistics on 2009 data is because this constitutes our training dataset for ANFIS. 2010 data on the other hand constitutes our checking dataset. Both sets of data would be used to validate the selected Fuzzy Inference System (FIS), and are both similar datasets. More insight would be provided in subsequent chapters. RTD signifies real time demand (historical data).

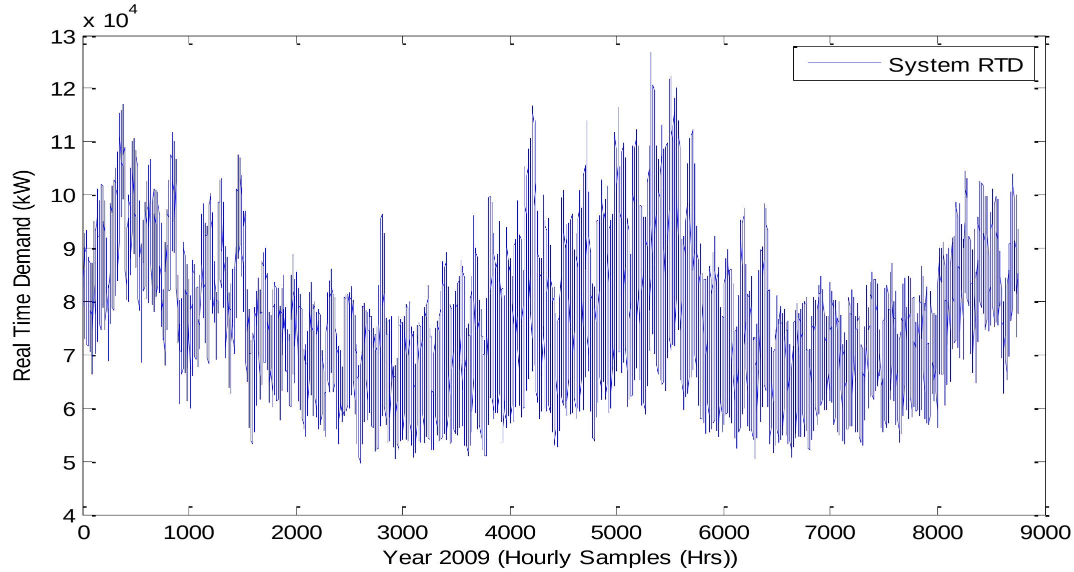

Figure 3: 2009 System Historical Real Time Demand (RTD)

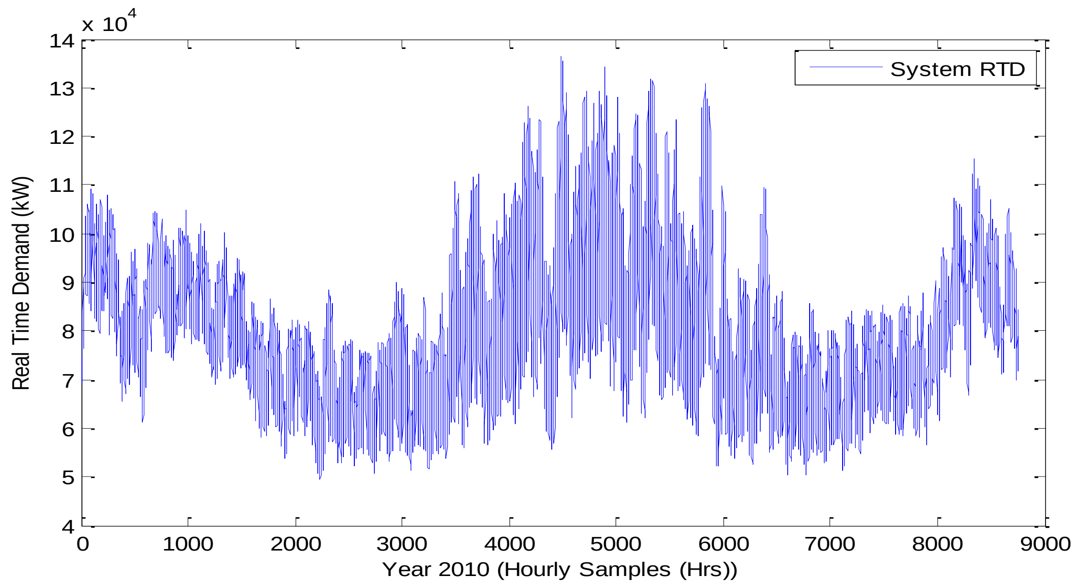

Figure 4: 2010 System Historical Real Time Demand (RTD)

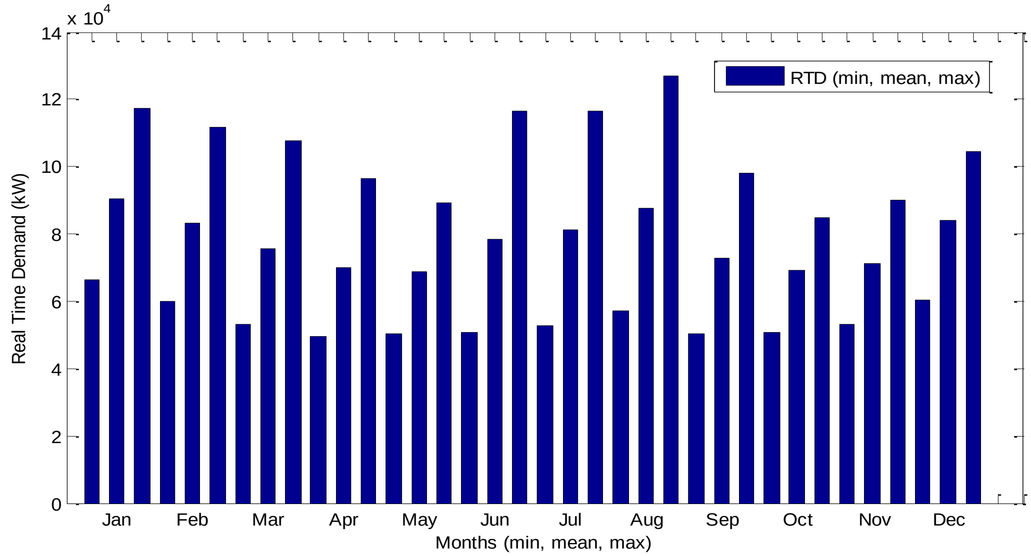

Figure 5: 2009 System Historical Monthly Summary

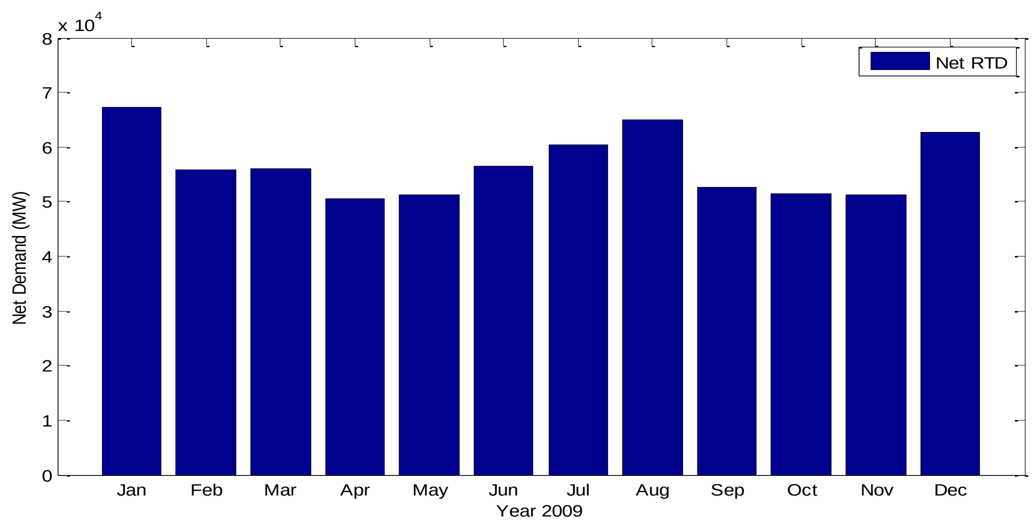

Figure 6: 2009 System Historical Monthly Net RTD

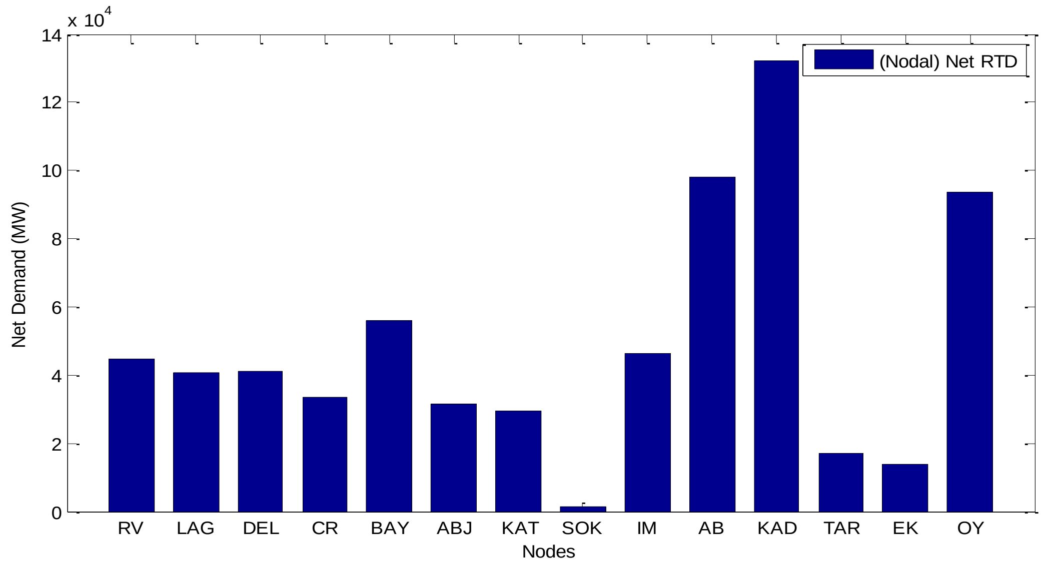

Figure 7: 2009 Nodal Net RTD (Excluding Substation Bus)

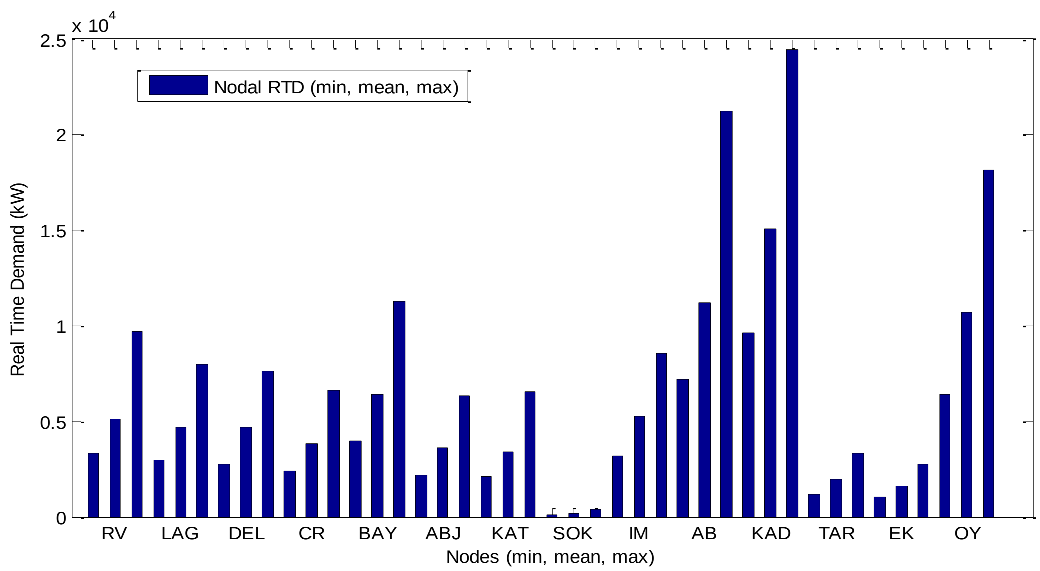

Figure 8: 2009 Nodal Historical Summary

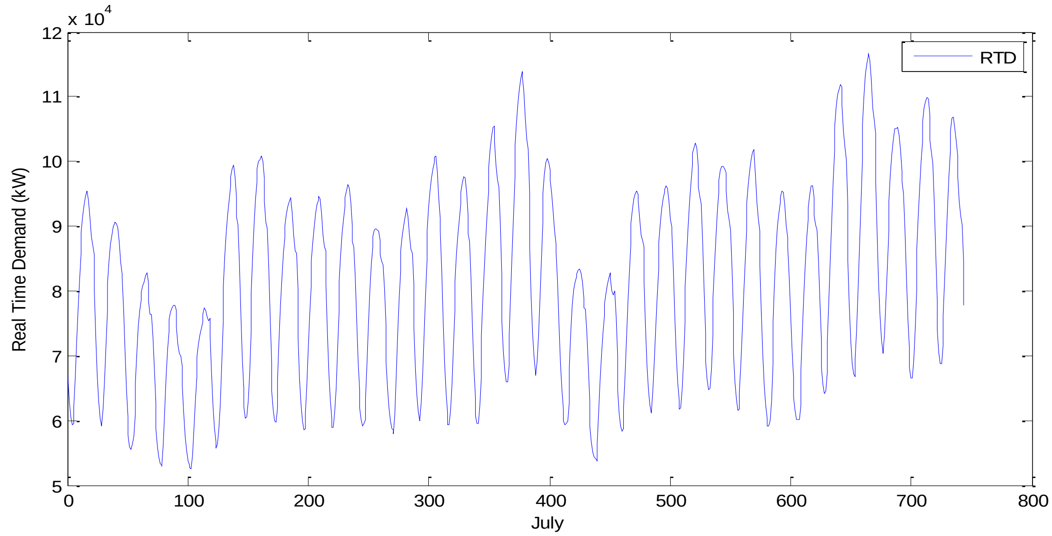

Figure 9: July 2009 System Historical RTD

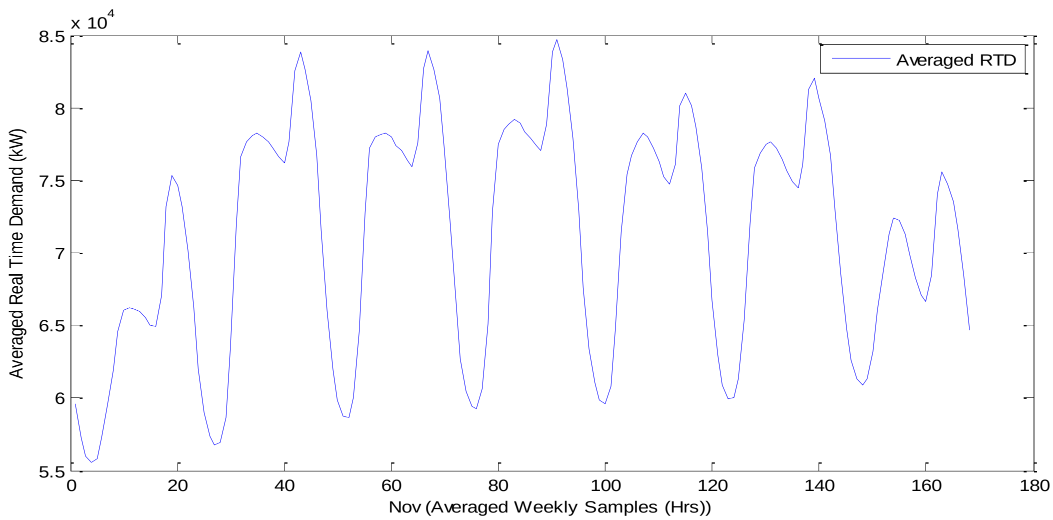

Figure 10: November 2010 System Weekly Mean RTD (1st 4 Weeks)

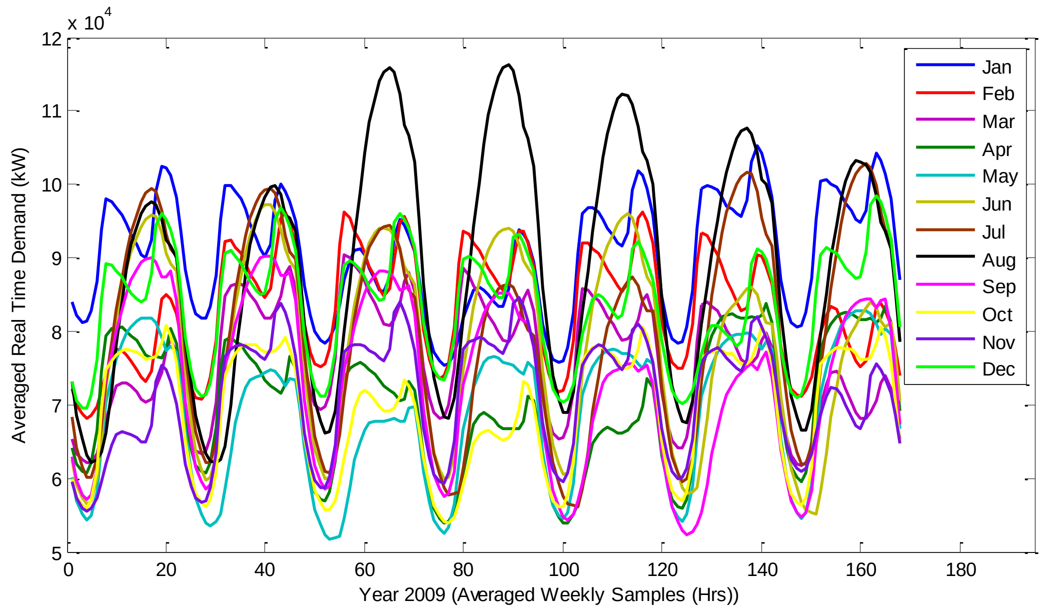

Figure 11: 2009 12 Month Weekly Mean RTD (1st 4 Weeks)

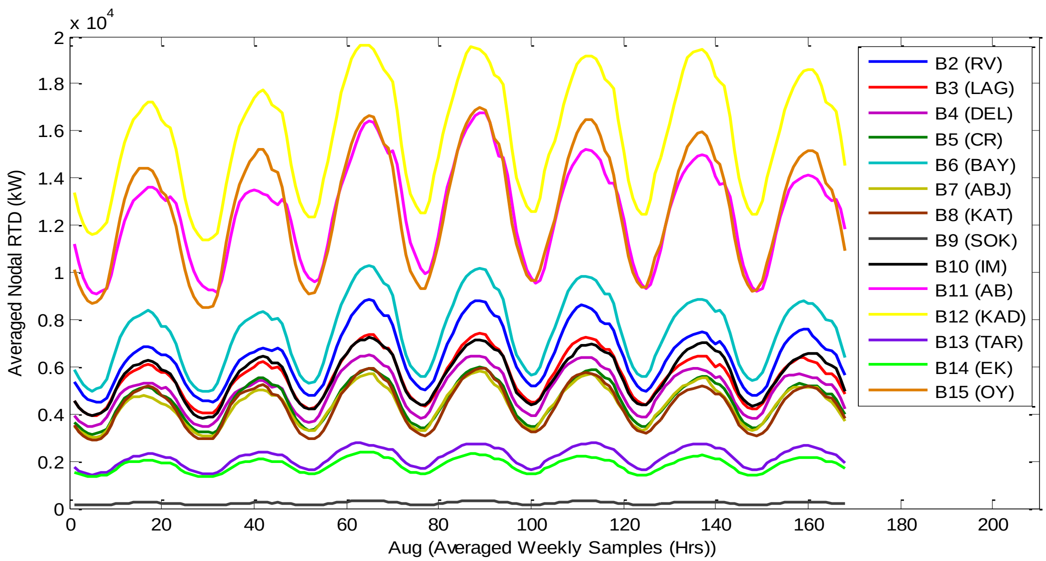

Figure 12: August 2009 Nodal Weekly Mean RTD (1st 4 Weeks)

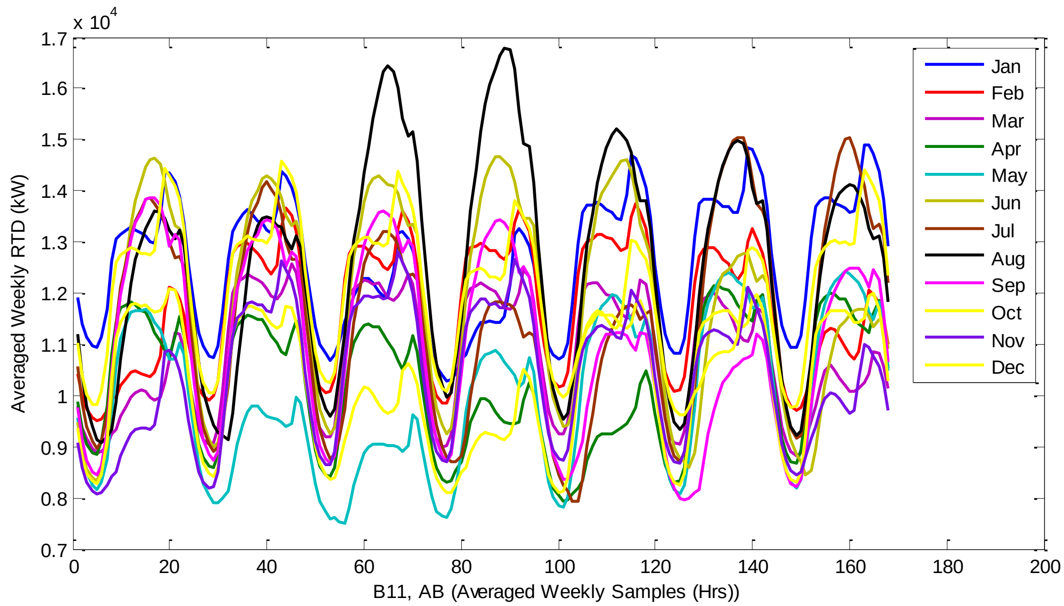

Figure 13: Bus 11 (AB) 2009 Weekly Mean RTD (1st 4 Weeks)

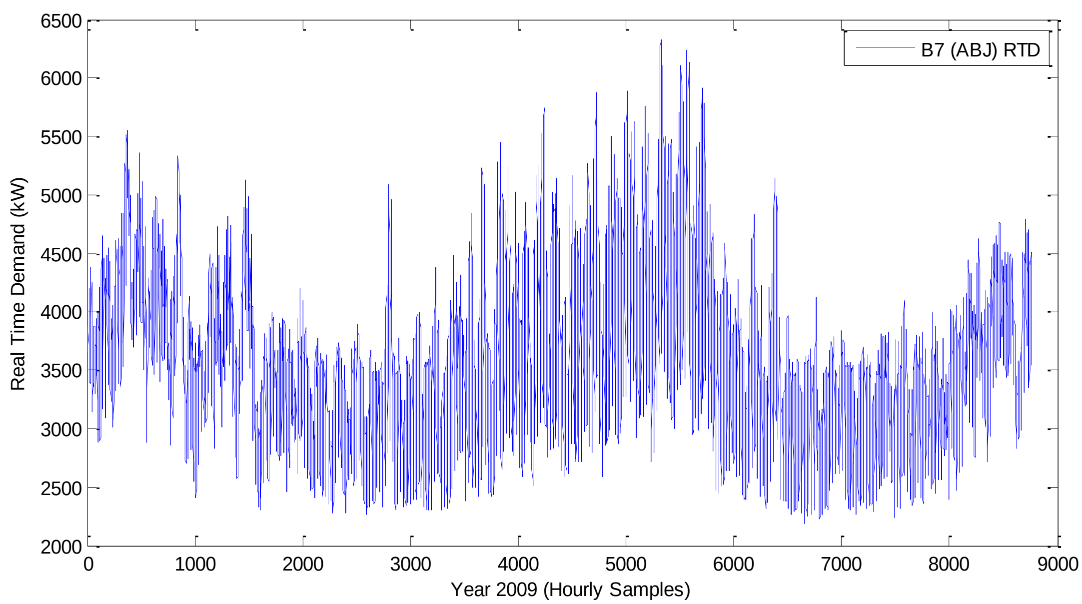

Figure 14: Bus 7 (ABJ) 2009 Load Profile Data

CHAPTER FOUR: ADAPTIVE NEURO-FUZZY INFERENCE SYSTEMS (ANFIS)

Proposed by J.-S.R Jang in 1993 [3], ANFIS integrates the best features of Fuzzy Systems (FS) and Artificial Neural Networks (ANNs) and can be used to learn information about a data set. We begin the section by presenting brief descriptions of ANN and FS and then proceed to ANFIS and how it can be used to estimate RLCs. This is somewhat of a data-fitting process.

4.1 Brief Overview of ANNs:

Inspired by biological nervous systems, Artificial Neural Networks (ANNs) consist of simple elements operating in parallel, where the connections between elements as in nature, largely determine the network function. By fine-tuning the values of the connections (weights) between elements, the ANN can be trained to perform a particular function [31]. ANNs are typically trained so that a specific input leads to a specific target output. Basically, the application of ANN is based on their capacity to mirror human behavior and neural structure to formulate a good approximation of functional relationships between input and output datasets. This is usually done using historic process data. In essence, many such input/target pairs are needed to train the network until its output and target match. ANN is therefore non-parametric and data-driven. The network learns the data and rewards the correct response of the system to an input by increasing the strength of the current matrix of nodal weights. This can be achieved with the aid of the MATLAB software [32], [33].

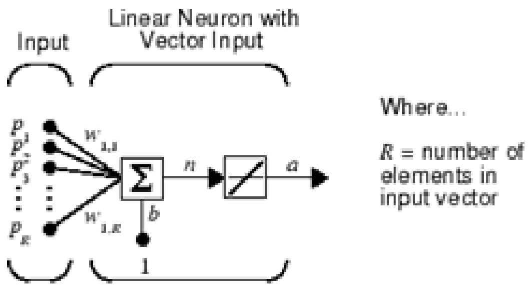

Figure 15: A Simple Neuron

There are three distinct processes that take place in a sample simple neuron as shown in the figure above:

- The weight function: a product of the weight, *w* and the input, *p*.
- The net input function: the sum of the weighted inputs, *wp* and the bias, *b*.
- The transfer function: produces the scalar output, *a* after the net input passes through the transfer function, *f*. Examples include; linear, tan-sigmoid and log-sigmoid amongst others.

$$a = f(wp + b) \tag{1}$$

The simple neuron can be extended to handle vector inputs, $\boldsymbol{p}$ in conjunction with a weight matrix, $\boldsymbol{W}$. Excluding the inputs, a layer in the network can include, the weights, the multiplication and summing operations, the bias and the transfer function. Two or more of the neurons can be combined in a layer, and a particular network could contain one or more such layers. A network can have several layers, where each layer has a weight matrix, a bias vector and an output vector. It is also common for the number of inputs to a layer be different from the number of neurons. For multilayer networks, the layer that produces the network output is called an output layer, whereas all other layers are called hidden layers. An example is shown below [32], [33].

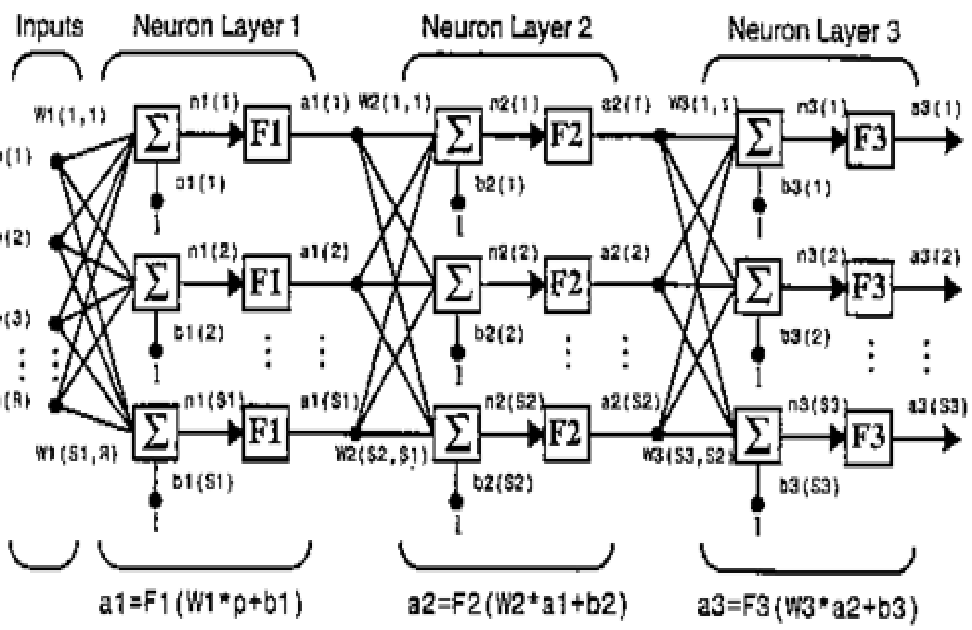

Figure 16: A Typical Feed-Forward Multi-Layer Neural Network

The parentheses help identify the layer elements, with respect to the input neuron, (1), (2) etc.; whereas the concatenated number alphabets identify the corresponding layer, (1), (2) etc. for example; *n1(2)* means the 2nd neuron in layer 1. *R* is the number of elements in the input vector. *S* is the number of neurons in each (layer); for example, *W2(S2,S1)*, implies a weight matrix in layer 2 of size *S2*-by-*S1*(i.e. number of neurons in layer 2 by number of neurons in layer 1). *F* is the transfer function used in the respective layer. A constant input 1 is normally fed to the bias for each neuron. The output of the neurons in the third layer for example is therefore the network output of interest and is:

$$a3 = F3(W3 * F2(W2 * F1(W1 * p + b1) + b2) + b3) \qquad (2)$$

The network can be trained for function approximation (nonlinear regression) or pattern recognition. Training involves adjusting the values of weights and biases of the network to optimize/improve network performance after network inputs, p and target outputs, t have been fed to the network. The *back-propagation* algorithm, amongst others is commonly used to optimize network performance by computing the gradient of the network performance with respect to the network weights, and the Jacobian of the network with respect to the weights [31]. The mean square error *mse* is the default performance function for feed-forward networks and is the average squared error between the network outputs, a and the target outputs t.

$$mse = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \qquad (3)$$

The multilayer feed-forward neural network often has one or more sigmoid hidden layer neurons followed by a linear output layer neurons. The multiple layered neural network with nonlinear transfer functions enable the network to learn nonlinear relationships between input and output vectors [32]. The dataset has to be properly representative of what the actual goal is. Essentially, ANNs have been applied in some of the literature mentioned earlier with regards to estimating the RLCs and it’s been noted that these RLCs were estimated quite well. It’s important the expert or user characterizes his network in terms of input/output relationships, as ANNs have many applications in the field of Electrical Engineering.

4.2 Brief Overview of Fuzzy Logic (FL):

Firstly, Fuzzy Logic (FL) using linguistic terms in its description is somewhat close to human thinking style. Membership degrees are designated to variables and are identical with fuzzy sets' theory which relates to classes of objects without crisp or clearly defined boundaries in which membership is a matter of degree. It's basically a logical system; an extension of multivalued logic. IF-THEN rules of the Fuzzy System (FS) are used to calculate different cases of each input's fuzzy sets. The optimum outputs obtained as a result of this operation are much closer to the target outputs. FL is all about the relative importance of precision. The building of the optimum results for the system depends on the experience of the expert [34], [35].

Some basic terminologies involving FL include: rules, fuzzy inference, membership functions and defuzzification, amongst others. IF-THEN statements used to map an input space to an output space are called rules and it's a primary mechanism for FL. All rules are evaluated in parallel, and the order of the rules is not important. A fuzzy inference translate the elements in the input vector and based on some set of rules, assigns values to the output vector. A curve defining how each point in the input space is mapped to a membership value (or degree of membership between 0 and 1) is a membership function (MF). Examples include triangular, trapezoidal, Gaussian, generalized bell membership functions amongst many others. Defuzzification implies obtaining a single number from the aggregated output fuzzy set [36].

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic and create IF-THEN rules that can be used to formulate the conditional statements that comprise FL. Example:

if x is A then y is B

where *A* and *B* are linguistic values defined by the fuzzy sets on the inputs X and Y, respectively. The IF-part is the *antecedent*, while the THEN-part is the *consequent*. The output of each rule is a fuzzy set [37], [38]. The fuzzy inference process as explained above involves five steps:

- Fuzzification of the input variables
- Application of the (AND or OR) fuzzy operator in the antecedent
- Implying the consequent from the antecedent
- Aggregation of the consequents across rules
- Deffuzification

Furthermore, two commonly used types of Fuzzy Inference Systems (FIS) include: *Mamdani* and *Sugeno*. Mamdani-type FIS [39] implies there's a fuzzy set for each output variable that needs defuzzification. Sugeno-type FIS [34] implies *singleton* output membership functions and can be thought of as a pre-defuzzified fuzzy set. Rather than integrating across the two-dimensional function to find the centroid as with Mamdami-type, Sugeno-type FIS uses the weighted average of a few data points, effectively simplifying the computation required. Moreover, Sugeno-type systems lend itself to the use of adaptive techniques for constructing fuzzy models. The membership functions can be customized using these adaptive techniques so that the FS best models the data [36]. Mamdani-type systems are more suited to human input.

In soft computing, a neuro-fuzzy system is one of the highly viewed methods in soft computing combining elements of fuzzy logic and neurocomputing. Jang developed ANFIS, serving an important role in the induction of rules from observations, can model nonlinear function of arbitrary complexity. We therefore present ANFIS in the next section.

4.3 Overview of ANFIS:

Essentially, a Fuzzy Inference System (FIS) in precise description is a model that maps input characteristics to input membership functions, input membership functions to rules, rules to a set of output characteristics, output characteristics to membership functions, and the output membership function to a single-valued output or a decision associated with the output. ANFIS comes into play when one can't ascertain what membership functions look like simply from looking at data. With ANFIS, one can construct a FIS whose membership function parameters are tailored using either a *back-propagation* algorithm or a *least-squares type* algorithm amongst others, on a well-represented input/output dataset in order to account for variations in the data values. The back-propagation algorithm is also known as the gradient descent algorithm. The MATLAB software aids in executing this task by providing an optimization scheme that best fit the dataset [36].

J.-S.R Jang in 1993 [3] proposed ANFIS which integrates the best features of Fuzzy Systems (FS) and Artificial Neural Networks (ANN) and can be used to learn information about a data set. This involves utilizing linguistic information from the FL as well as the learning capability of an ANN for automatic fuzzy IF-THEN rule generation and parameter optimization. To illustrate the system architecture he proposed, we assume the Fuzzy Inference System (FIS) which consists of five layers of adaptive network with two inputs x and y and one output f. Essentially there are five components: inputs and output database, a Fuzzy System generator, a Fuzzy Inference System (FIS), and an Adaptive Neural Network. The multilayer feed-forward network in which each node (neuron) performs a particular function on incoming signals is an adaptive network.

Each layer contains some nodes described by the node function. Each ANFIS layer has specific functions that are used in calculating input and output parameter sets. A few layers have the same number of nodes, and nodes in the same layer have similar functions. A fixed node is indicated with a circle, whereas an adaptive node is indicated with a square. An adaptive node has parameters while a circle node has none. This section supposes that the system consists of two fuzzy IF-THEN rules based on Takagi and Sugeno's type [40].

Rule 1: if x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$.

Rule 2: if x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$.

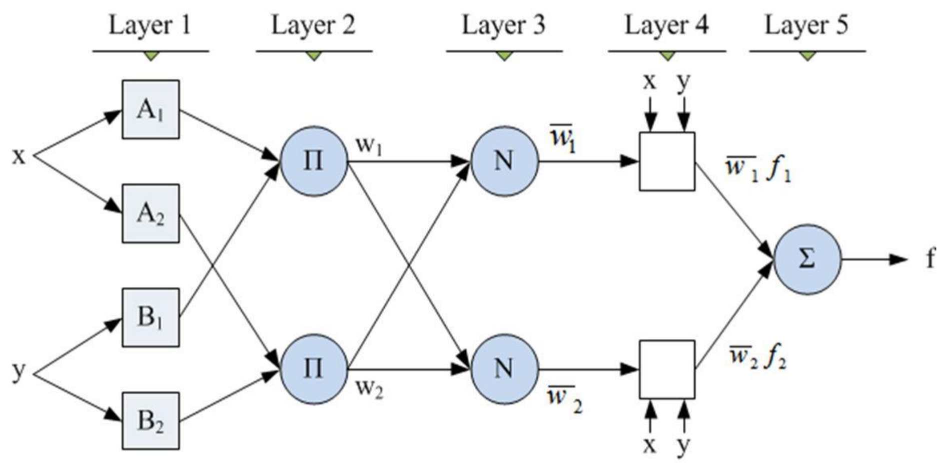

Figure 17: Proposed ANFIS by J.-S. R. Jang

The crisp inputs to the nodes are x and y; whereas A_1, B_1, A_2, B_2 are fuzzy sets, and f which is sometimes referred to as the weighted average. The node in the i-th position of the k-th layer is denoted as $O_{k,i}$, and the node functions in the same layer are of the same function family as described below:

Layer 1: is the input layer and every node *i* in this layer is a square (adaptive) node with a node function (eq.4). $O_{k,i}$ is the membership function of A_i, and it specifies the degree to which the given *x* satisfies the quantifier A_i. Usually, the bell-shaped membership function is selected by this method as the input membership function (eq.5) with maximum equal to 1 and minimum equal to 0. This membership function has been found to be commonly preferable and is employed from a computational point of view.

$$O_{1,i} = \mu A_i(x) \quad for\ i = 1,2 \tag{4}$$

$$\mu A_i(x) = \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}} \tag{5}$$

where a_i and b_i vary the curve's width, b_i is a positive value and c_i denotes the curve's center. Otherwise known as antecedent parameters of the FIS.

Layer 2: is the layer where every node here is a circle (fixed) node, marked by a circle and labeled Π, with the node function (eq.6) to be multiplied by input signals to serve as output signal.

$$O_{2,i} = w_i = \mu A_i(x) \times \mu B_i(y) \quad for\ i = 1,2 \tag{6}$$

where the output signal w_i represents the firing strength of a rule.

Layer 3: here also, every node is a fixed node, marked by a circle and labeled *N*, with the node function (eq.7) to normalize the firing strength by calculating the ratio of the *i*-th node firing strength to the sum of all rules' firing strength.

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2} \quad for\ i = 1,2 \tag{7}$$

Layer 4: all nodes i are adaptive nodes in this layer, marked by a square, with node function (eq.8). The parameters in this layer will be referred to as consequent parameters. This is where ANFIS applies least-squares technique to identify them.

$$O_{4,i} = \overline{w_i}\, f_i = \overline{w_i}\,(p_i x + q_i y + r_i) \tag{8}$$

where p_i, q_i, r_i are the parameters of linear function (in THEN part) in a Sugeno fuzzy model.

Layer 5: The single node in this layer is a fixed node and computes the overall output as the summation of all incoming signals (eq.9).

$$O_{5,i} = \sum_i \overline{w_i}\, f_i = \frac{\sum_{i=1} w_i f}{\sum_{i=1} w_i} = overall\ output \tag{9}$$

With the aid of MATLAB, and a collected input/output dataset of which one cannot easily discern membership functions arbitrarily or establish some sort of relationship/characterization between the input/output data pair; a Fuzzy Inference System (FIS) can be constructed whose membership function parameters are tuned (adjusted) using either a back-propagation (gradient-descent) algorithm alone or in combination with a least squares type of method (forming a *hybrid learning algorithm*) through the learning process. The ANFIS system is generally trained by the hybrid learning algorithm. This adjustment allows the fuzzy systems to learn from the data they are modeling and aimed at matching the ANFIS output with the training data. In the forward pass, the algorithm uses least-squares method to optimize the consequent parameters, and keeps the premise parameters are fixed. Once the optimal consequent parameters are obtained, the backward pass begins to optimize the premise parameters. However, in this stage the hybrid algorithm uses a gradient descent (back-propagation) method for updating and tuning optimally the premise parameters corresponding to the fuzzy sets in the input. This is done for each epoch (iteration). The signals in the forward pass are the node outputs, whereas in the backward pass, they are the error rates [3].

The FIS is a network type structure similar to that of a neural network, mapping inputs through output membership functions and associated parameters, and then through output membership functions and associated parameters to outputs, can be used to interpret the input/output map [36]. The MATLAB software presents a few restrictions to implement an ANFIS on an input/output dataset. They include:

- Current application of only Sugeno-type decision method
- Only one output can be defined
- Deffuzification is weighted mean value
- The output uses only constant and linear output membership functions

All these restrictions are still valid for the present test case we are trying to implement: estimate RLCs. As mentioned earlier, it’s key to note that the gradient method is applied to the calculation of input membership function parameters, and least-squares method is applied to the calculation of the output function parameters. Most research conclude that the effectiveness of the ANFIS depend on the input selection, the membership function (MF) selection and the rule generation.

CHAPTER FIVE: METHODOLOGY

An adaptive neuro-fuzzy inference system model, modified to suit different purposes, was used to estimate RLCs from historical load data. Data input selection was done by trial and error method, heuristically. Basically, ANFIS takes the initial fuzzy model generated by MATLAB functions and tunes it by means of a *hybrid learning algorithm*. At each epoch (iteration), an attempt is made to reduce the error measure, usually defined as the sum of the squared difference between actual and desired output. Training stops when either the predefined epoch number or error rate is obtained. The two passes in the hybrid learning procedure have been previously described. When the values of the premise parameters are learned, the overall output can be expressed as a linear combination of the consequent parameters.

The input variables of the ANFIS are selected based on preprocessing of the original data and guidelines for input selection for ANFIS learning is presented by Jang in [41]. Real-world modeling problems usually involves tens (or even hundreds) of potential inputs and use them accordingly. Therefore, we need to have a heuristic way to quickly determine the priorities of these potential inputs and use them accordingly. The rule bases of ANFIS are generated based on linear Sugeno fuzzy model. The hybrid algorithm is used. The Mean Absolute Percent Error (MAPE) and Root Mean Squared Error (RMSE) were both computed.

In our dataset described in Chapter 3, we have hourly samples of two years of data from Jan 1st, 2009 to Dec 31st, 2010. This implies there are 8760×2, (17520) samples of historical metered data for our 11kV, 15 node test distribution system under the simplifying assumption the network is balanced and can be represented using its single phase equivalent. The sample data includes both substation data and demand data for each individual node. The substation data is the overall system Real-Time-Demand (RTD). Before implementations begin in MATLAB, it's important the training data used to estimate membership functions has to be fully representative of the features of the data that the trained FIS is intended to model. The model has to be validated to ensure this statement is maintained.

The MATLAB software can be used to first hypothesize a parameterized model (FIS) structure (relating inputs to input membership functions to rules to output characteristics to output membership functions to a single-valued output or a decision associated with the output). This is done using a subset of the dataset, normally referred to as training data. Then the other input/output data-subset, normally referred to as checking data is used to train this model to mimic the training data presented to it by modifying the membership function parameters according to a chosen error criterion [36].

To validate the model estimated by ANFIS, the input/output dataset is divided into checking and training data. This is to ensure that their corresponding output datasets from the FIS model are similar and is fully representative of the network. In this thesis, the dataset was divided into two:

Training dataset = 1st 8760 samples of data (Year 2009).

Checking dataset = 2nd 8760 samples of data (Year 2010).

Both datasets are checked to ensure they are fully representative of the dataset by selecting a dataset (training) in which the trained model is intended to emulate, and yet sufficiently distinct from the dataset (checking) so as not to render the validation process trivial. This is useful so as to check for the presence of noisy measurements in the dataset. The training dataset checks the generalization capability of the resulting FIS, whereas the checking dataset is used to validate this FIS. This is done to observe any over-fitting concerns. The MATLAB software helps in this regard to select the membership function parameters associated with the minimum checking error of the FIS just prior to over-fitting. Over-fitting is accounted for by comparing the training and checking errors. Ideally, they must both decrease at the same time throughout the training period. If they don't, then this indicates over-fitting. But if the checking error begins increasing even at the first epoch (iteration), while the training error decreases, then the trained FIS has to be retrained because clearly, this membership function is not the best choice for modeling the entire dataset. You may have to use other membership function choices or increase the size of the dataset. This is indicated by plots shown in Chapter 6 where the results are placed. The errors computed are actually RMSE.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}} \qquad (10)$$

where $\hat{y}_t$ are the estimated values for times t of the specific target y_t for n samples.

MATLAB commands such as *genfis1*, *genfis2* and *anfis* were used to generate an initial FIS and train it. *Time series estimation* was attempted heuristically on the input/output dataset to estimate RLCs. We begin with estimating RLCs for the substation and then proceed to estimating RLCs for select nodes on the test system.

In estimating the RLCs, we need to use known values of the load profile (overall system OR select node) up to say a point in time t to estimate the value at some future point in time, say (t + p) within the same load profile. In this accord, the output to our ANFIS is say (t + 24) and must be within the dataset (the load profile for 2009-2010), whereas the inputs are various adaptations of the load at that same time, t. Also remember that our dataset represents historical hourly metered load; this implies hourly samples. This model configuration is somewhat of a *time series estimation*. Various adaptations could include: present load (t), previous 24 hour load (t – 24), previous week same hour load (t – 168), previous 24 hour averaged load (($\sum Y(t-i)$) ÷ 24) for i = 1 to 24, weekly model factor (MF), daily model factor (DMF), RTD(t). RTD(t) represents the overall system load (RTD) at the point in time within the same nodal load profile. These adaptations are chosen heuristically for different cases with the sole objective of seeking a model that generates the least Minimum Absolute Percent Error (MAPE).

We heuristically selected 4 weeks in each month to be used to determine some sort of Weekly and Daily RLCs for each month with regards to the overall system load (RTD) or nodal demand. The MATLAB implementation of ANFIS helped estimate that RLC. A daily and/or weekly RLC was chosen on one basic criterion:

The Mean Absolute Percent Error (MAPE) between the corresponding windows (between the chosen estimated RLC and the network's target vector) must be the least.

$$MAPE = \frac{1}{N}\sum_{t=1}^{n}\left|\frac{T_t - E_t}{T_t}\right| \quad (11)$$

Example: In March, there are more than 4 weeks of data. But only four weeks are to be used in this analysis. In selecting the weekly RLC, from the estimated RLC (fitted ANFIS output), the estimated RLC and the original target vector are divided into four sections to represent the selected four weeks (672 hours' samples of data). Hence we have two sets of data broken into 168 samples. That's four sections/parts with 168 hours' samples each. So we compare each section and select the section of the estimated RLC that has the smallest MAPE when compared with the original target data. This selection then represents our weekly RLC. But since we'd be dealing with 2 year's data, we'd be examining eight weeks of data, rather than four. March 2008 and March 2009 is an example. Also a plot of the selected weekly average is compared with the averaged RTD for the specified month.

Same analogy is applied in selecting the daily RLC for each month. Four weeks of data. Seven days each. After heuristically selecting our input dataset and generating an output from ANFIS, we select the corresponding days in those four weeks and compute their respective MAPEs. The section with the smallest MAPE represents the day's RLC for the select 4 weeks in the selected month. This can also be done to multiple data sets (stacked yearly). March 2009, 2010 means we select eight weeks and proceed in a similar manner. We also used day D1 to D7 to represent Sunday to Saturday respectively. This hugely favoured how we selected our dataset to be used in the ANFIS implementation.

5.1 Weekly RLC for Overall System Load

Heuristically, after trying different adaptations of the respective target output, the model below achieved the least MAPE when estimating weekly RLCs to represent monthly historical data for the overall system load. Preferably used when one wants to have a snapshot of how load varies weekly, in a selected month and can also be used as pseudo-measurements along with real time synchronized measurements in DLE. Our Inputs (X) and Targets (Y) for implementing ANFIS are as follows:

Inputs (X): Weekly System Model Factor (WSMF), Previous Week Same Hour Load ($Y(t - 168)$), Previous 24 Hour Load ($Y(t - 24)$), Present Load ($Y(t)$).

Target (Y): $Y(t + 24)$; (time series estimation available within the load profile dataset).

WSMF stands for weekly system model factor and was computed using historical substation data by dividing the average load in its specific week by the maximum load in that same select week. This factor somewhat aided in improving the results of ANFIS.

The modeling criterion shown in the table below is used to estimate weekly and daily RLCs for the overall system load.

Table 1: Modeling Criterion for Estimating Weekly/Daily RLCs for the System

S/N	Custom ANFIS	Variables
1	Number of Inputs	Four
2	Membership Function Type	Generalized Bell
3	Number of Membership Functions	Varied from 2 to 3
4	Learning Algorithm	Hybrid Learning Algorithm
5	Epoch Size	Varied from 100 to 150
6	Data Size	Data per Hour (17520 Samples)
7	Sugeno-Type System	First Order
8	Output Type	Linear
9	Initial Step Increase/Decrease Size	1.1/0.9
10	Number of Linear Parameters	80
11	Number of Nonlinear Parameters	24
12	Number of Nodes	55
13	Number of Fuzzy Rules	16
14	Data/Parameter Ratio	≈ 84

The generated FIS structure with the help of the MATLAB software contains 16 fuzzy rules with a total of 104 parameters. It's also recommended that the number of training data points be several times larger than the number of parameters being estimated. In our case, that ratio is approximately 84.

Essentially, because we needed previous week's data as one of our selected inputs, the RLCs obtained from the ANFIS fit excluded the first week's data (168 samples) in the estimation/data fitting process. Hence a reason why we have Not-A-Number (NaN) values for the 1st week of January. This will be seen in the next Chapter.

An initial FIS is generated with the training dataset and validated with the checking dataset. The ANFIS output which fits this dataset with minimum error is what we designate as the RLC. Because of the inconsistencies in various sections of the fitted data with respect to the real data, we propose that weekly and daily RLCs be selected based on the section/window with the least MAPE.

5.2 Daily RLC for Overall System Load

Heuristically, after trying different adaptations of the respective target output, the model below achieved the least MAPE when estimating daily RLCs to represent weekly historical data for the overall system load. This is preferably used as pseudo-measurements along with real time synchronized measurements in DLE. Our Inputs (X) and Targets (Y) for implementing ANFIS are as follows:

Inputs (X): Daily System Model Factor (DSMF), Previous Week Same Hour Load (Y(t – 168)), Previous 24 Hour Load (Y(t – 24)), Present Load (Y(t)).

Target (Y): Y(t + 24); (time series estimation available within the load profile dataset).

DSMF stands for daily system model factor and was computed using historical substation data by dividing the average load on each day by the maximum load on that same select day. This factor somewhat aided in improving the results of ANFIS. Its modeling criterion is also same as with estimating weekly RLCs for the overall system load shown in table 1. Only difference is in the selection of inputs. Essentially, because we needed previous week's data as one of our selected inputs, the RLCs obtained from the ANFIS fit excluded the first week's data (168 samples) in the estimation/data fitting process. Replacing them with NaN in the ANFIS output.

An initial FIS is generated with the training dataset and validated with the checking dataset. The ANFIS output which fits this dataset with minimum error is what we designate as the RLC. Because of the inconsistencies in various sections of the fitted data with respect to the real data, we propose that weekly and daily RLCs be selected based on the section/window with the least MAPE.

5.3 Weekly RLC for Selected Node 5 (CR)

Heuristically, after trying different adaptations of the respective target output, the model below achieved the least MAPE when estimating weekly RLCs to represent monthly historical data for the overall system load. Preferably used when one wants to have a snapshot of how load varies weekly, in a selected month and can also be used as pseudo-measurements along with real time synchronized measurements in DLE. In estimating nodal weekly RLCs for select nodes, we proceed in a similar manner as with weekly RLCs for overall system load but with different input/output data set possessing certain characteristics of the particular node in question. Our Inputs (X) and Targets (Y) for implementing ANFIS are as follows:

Inputs (X): Weekly Nodal Model Factor (WNMF), Previous Week Nodal Same Hour Load (Y(t – 168)), Previous 24 Hour Nodal Load (Y(t – 24)), Present Nodal Load (Y(t)).

Target (Y): Y(t + 24); (time series estimation available within the node's load profile dataset).

WNMF stands for weekly nodal model factor and was computed using historical nodal data by dividing the average load in its specific week by the maximum load in that same select week for the particular node in question. This factor somewhat aided in improving the results of ANFIS.

Each node is computed independently since we are using a single output Sugeno-type system. This means that all selected inputs/outputs dataset are with respect to the select node. The modeling criterion shown in the table below is used to estimate weekly and daily RLCs for the select nodal load.

Table 2: Modeling Criterion for Estimating Weekly/Daily RLCs for a Select Node

S/N	Custom ANFIS	Variables
1	Number of Inputs	Four
2	Membership Function Type	Generalized Bell
3	Number of Membership Functions	Varied from 2 to 3
4	Learning Algorithm	Hybrid Learning Algorithm
5	Epoch Size	Varied from 50 to 100
6	Data Size	Data per Hour (17520 Samples)

S/N	Custom ANFIS	Variables
7	Sugeno-Type System	First Order
8	Output Type	Linear
9	Initial Step Increase/Decrease Size	1.1/0.9
10	Number of Linear Parameters	80
11	Number of Nonlinear Parameters	24
12	Number of Nodes	55
13	Number of Fuzzy Rules	16
14	Data/Parameter Ratio	≈ 84

The generated FIS structure with the help of the MATLAB software contains 16 fuzzy rules with a total of 104 parameters. It's also recommended that the number of training data points be several times larger than the number of parameters being estimated. In our case, that ratio is approximately 84. Also again, because we needed previous week's data as one of our selected inputs, the RLCs obtained from the ANFIS fit excluded the first week's data (168 samples) in the estimation/data fitting process. Replacing those entries with NaNs in the ANFIS output as seen in the next Chapter.

An initial FIS is generated with the training dataset and validated with the checking dataset. The ANFIS output which fits this dataset with minimum error is what we designate as the RLC. Because of the inconsistencies in various sections of the fitted data with respect to the real data, we propose that weekly and daily RLCs be selected based on the section/window with the least MAPE.

5.4 Daily RLC for Selected Node 5 (CR)

Heuristically, after trying different adaptations of the respective target output, the model below achieved the least MAPE when estimating daily RLCs to represent weekly historical data for the overall system load. This is preferably used as pseudo-measurements along with real time synchronized measurements in DLE. In estimating nodal daily RLCs for select nodes, we proceed in a similar manner as with daily RLCs for overall system load but with different input/output data set possessing certain characteristics of the particular node in question. Our Inputs (X) and Targets (Y) for implementing ANFIS are as follows:

Inputs (X): Daily Nodal Model Factor (DNMF), RTD(t), Previous 24 hour Nodal Averaged Load $((\sum Y(t-i)) \div 24)$ for $i = 1$ to 24, Previous Week Same Hour Load $(Y(t-168))$, Previous 24 Hour Load $(Y(t-24))$, Present Load $(Y(t))$.

Target (Y): $Y(t + 24)$; (time series estimation available within the load profile dataset).

DNMF stands for daily nodal model factor and was computed using historical substation data by dividing the average load on each day by the maximum load on that same select day for the particular node in question. This factor somewhat aided in improving the results of ANFIS.

Its modeling criterion is also same as with estimating weekly RLCs for a select nodal load as shown in table 2. Only difference is in the selection of inputs. Also, because we needed previous week's data as one of our selected inputs, the RLCs obtained from the ANFIS fit excluded the first week's data (168 samples) in the estimation/data fitting process. Each node is also computed independently since we are using a single output Sugeno-type system. This means that all selected inputs/output are with respect to the select node except RTD(t).

An initial FIS is generated with the training dataset and validated with the checking dataset. The ANFIS output which fits this dataset with minimum error is what we designate as the RLC. Because of the inconsistencies in various sections of the fitted data with respect to the real data, we propose that weekly and daily RLCs be selected based on the section/window with the least MAPE. Essentially, all methods involve understanding the ANFIS architecture/structure and tailoring our input/output dataset to achieve our objective; the estimated RLC. We present our results in the next section. The basic flow diagram of computations is described using the following flowchart:

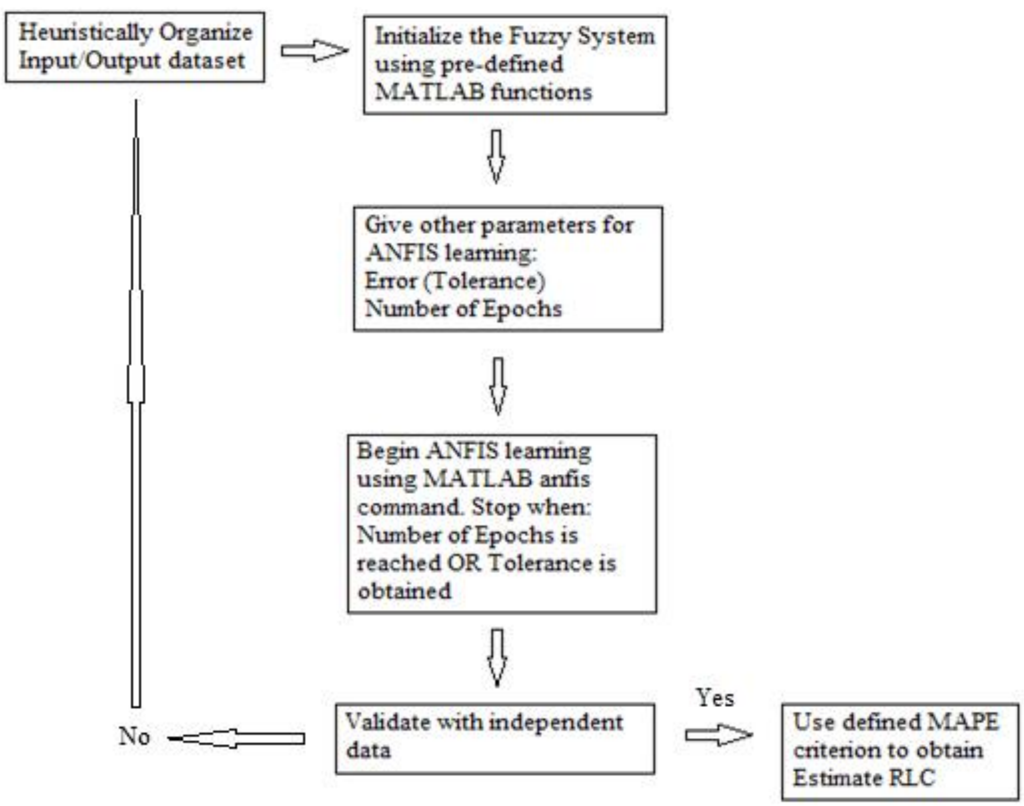

Figure 18: Basic Flow Diagram Using ANFIS Computation

CHAPTER SIX: RESULTS

We first present results for estimating weekly and daily RLCs for the overall system load at the substation. And then proceed to present results for estimating weekly and daily RLCs for the select node 5 (CR). These results were consistent even at other select periods in the dataset and other buses on the test system. But not all results will be shown here. We begin each subsection by displaying our target dataset and also how the fitted ANFIS output (estimated RLC) compares with it.

6.1 Estimated Weekly RLCs for the Overall System Load

The 2009-2010 Historical System Real Time Demand (RTD) is presented in the figure below.

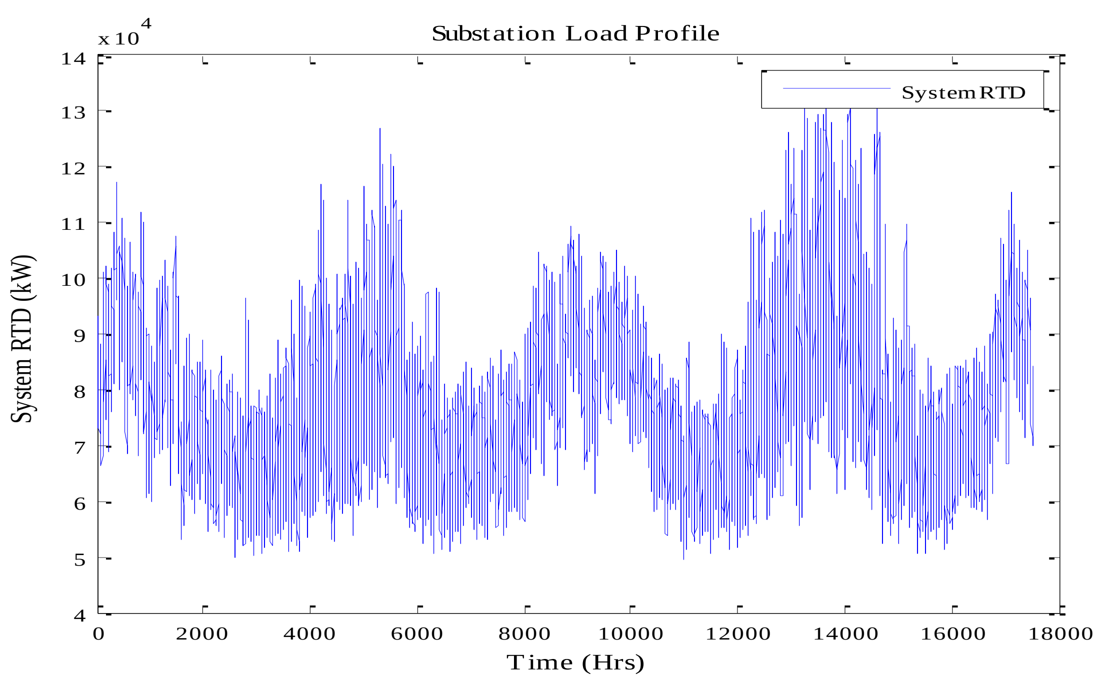

Figure 19: 2009-2010 Historical System Real Time Demand (RTD)

Upon training and testing the ANFIS, the following results were obtained. The system RTD being out target vector, whereas the estimated RLC is the output of the ANFIS implementation. The ANFIS architecture/structure used for this implementation is shown below and is also the same architecture that will be used in estimating the daily RLCs for the overall system with the exception on the selection of inputs. The estimated RLC and system RTD is then plotted.

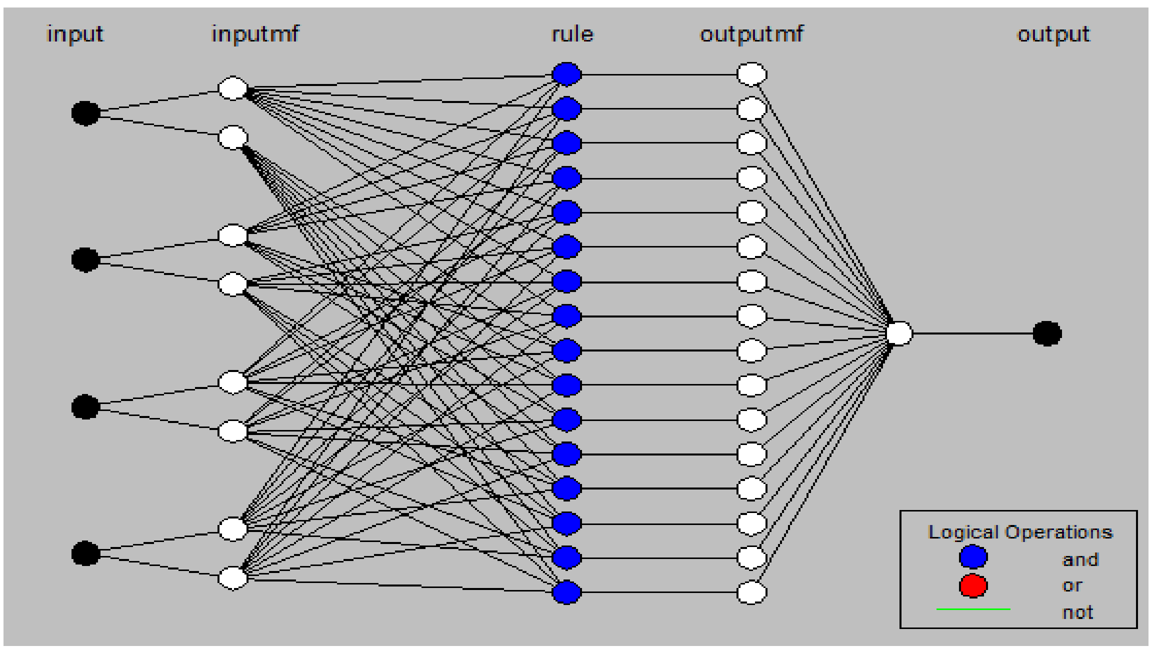

Figure 20: ANFIS Structure for Estimating System RLC

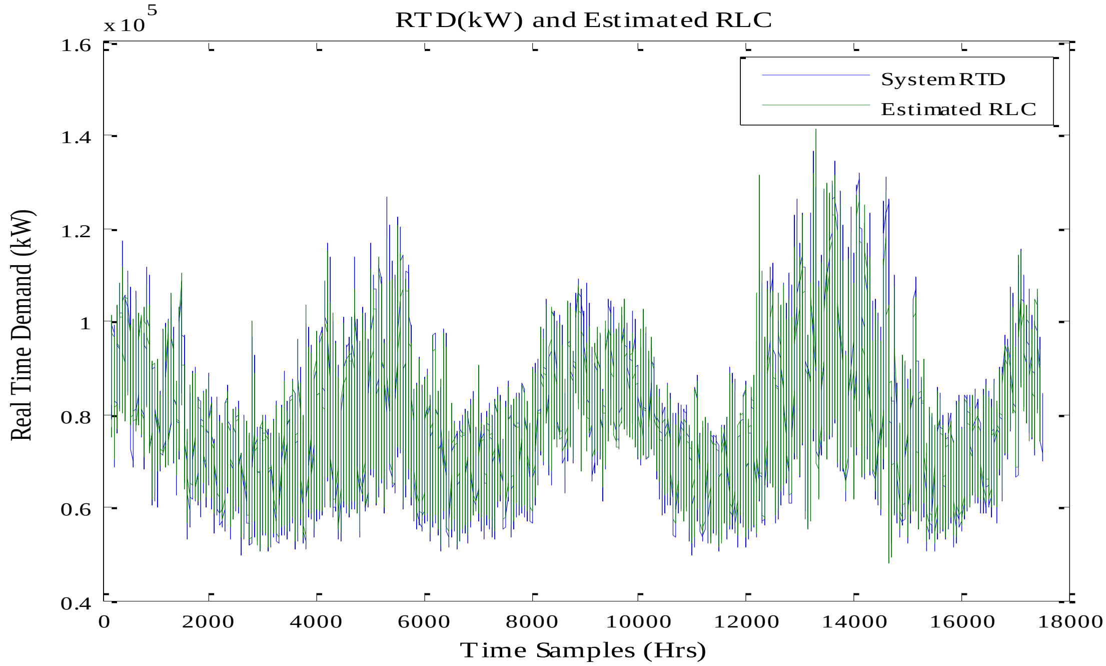

Figure 21: Estimate RLC and System RTD (intended for Weekly RLC)

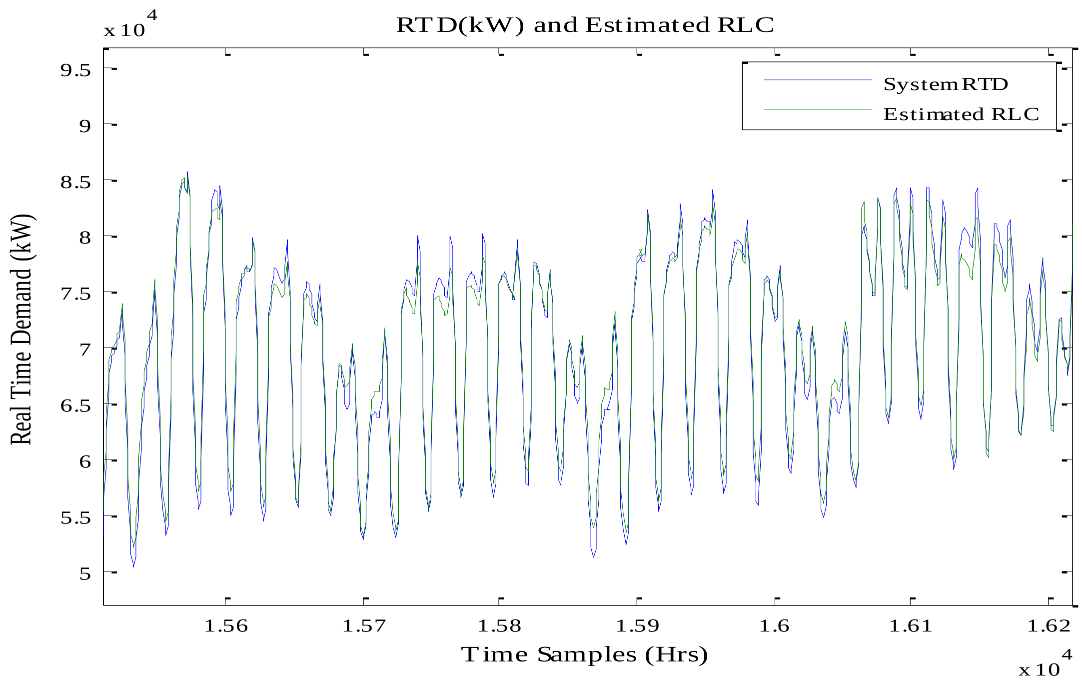

Figure 22: Magnified Estimate RLC and System RTD (intended for Weekly RLC)

Overall MAPE between the estimated RLC and the system RTD is 2.5071%. The corresponding RMSE between estimated RLC and system RTD averaged at 6.2790×10^3 kW. The table below summarizes the methodology explained in the previous chapter where the weekly/daily estimated RLC is selected based on the weekly window/section with the least MAPE. The ones boldly highlighted represents the estimate RLC for that period in the dataset.

Table 3: Tabular MAPE Results Comparing Each Week for Weekly Estimate Substation RLC

	Year 2009				Year 2010			
	January							
Week/Window	1	2	3	4	1	2	3	4
Estimate RLC MAPE	NaN	3.0439	3.6580	2.0672	3.5247	4.2699	**1.9837**	3.4545
	February							
Week/Window	1	2	3	4	1	2	3	4
Estimate RLC MAPE	2.9859	2.1857	**1.2555**	1.8272	3.5504	2.4485	3.5648	1.9084
	March							
Week/Window	1	2	3	4	1	2	3	4
Estimate RLC MAPE	2.7775	2.9444	1.4536	1.9165	**1.2002**	2.7785	1.9662	2.5529

We proceed to show results for the estimated substation RLC for the month of March. Bear in mind that RLC are used only in load variation studies and can be used as pseudo-measurements alongside real-time measurements for DLE. The initial and adjusted input membership functions before and after ANFIS training respectively are shown below:

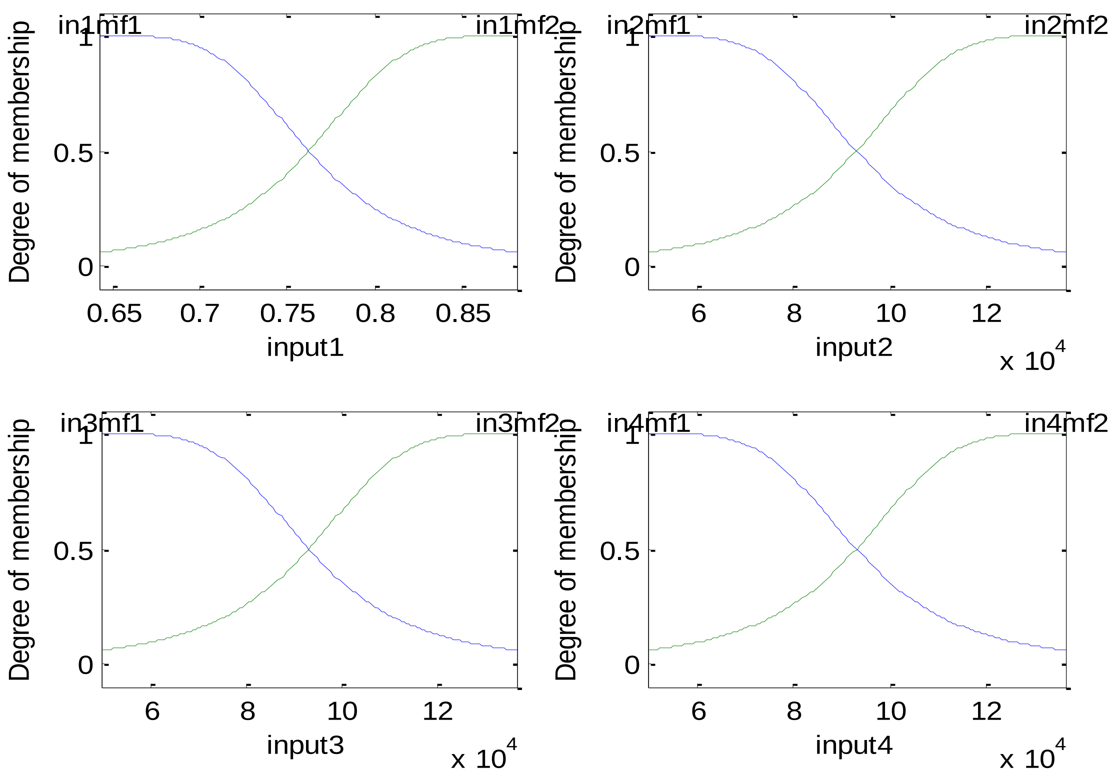

Figure 23: Initial Input Membership Functions for Estimating System Weekly RLC

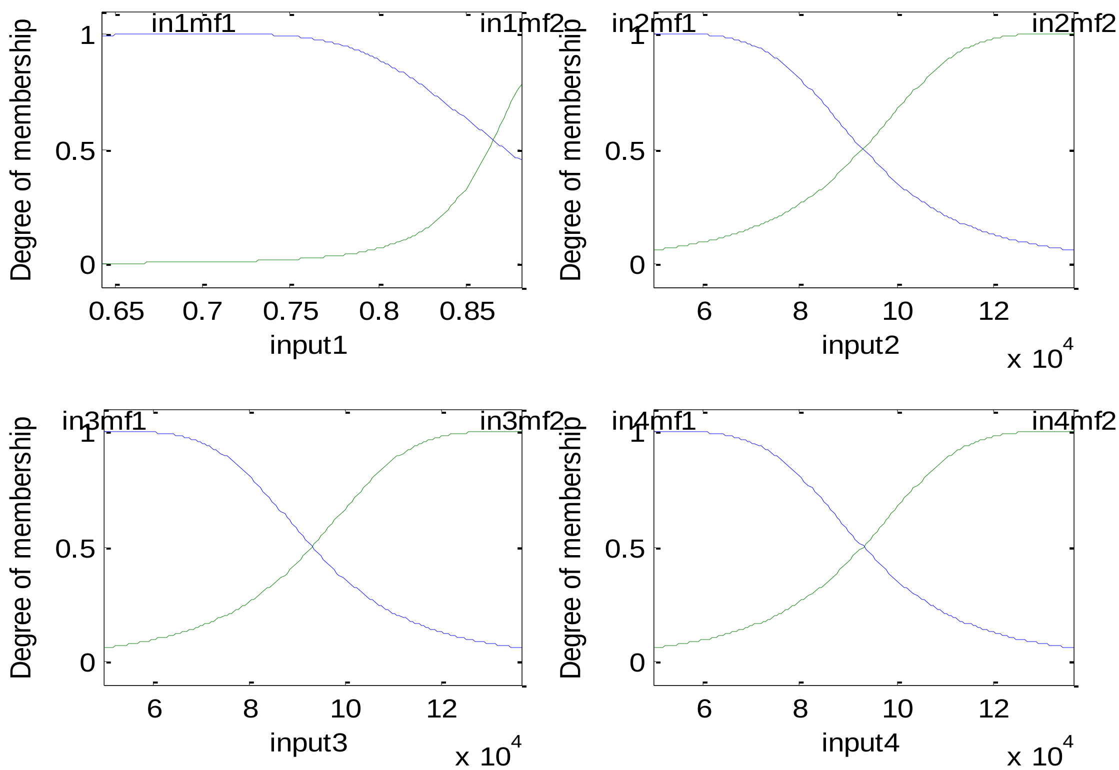

Figure 24: Adjusted Input Membership Functions for Estimating System Weekly RLC

Furthermore, the figure below shows how the training and checking errors interact with each other during ANFIS training.

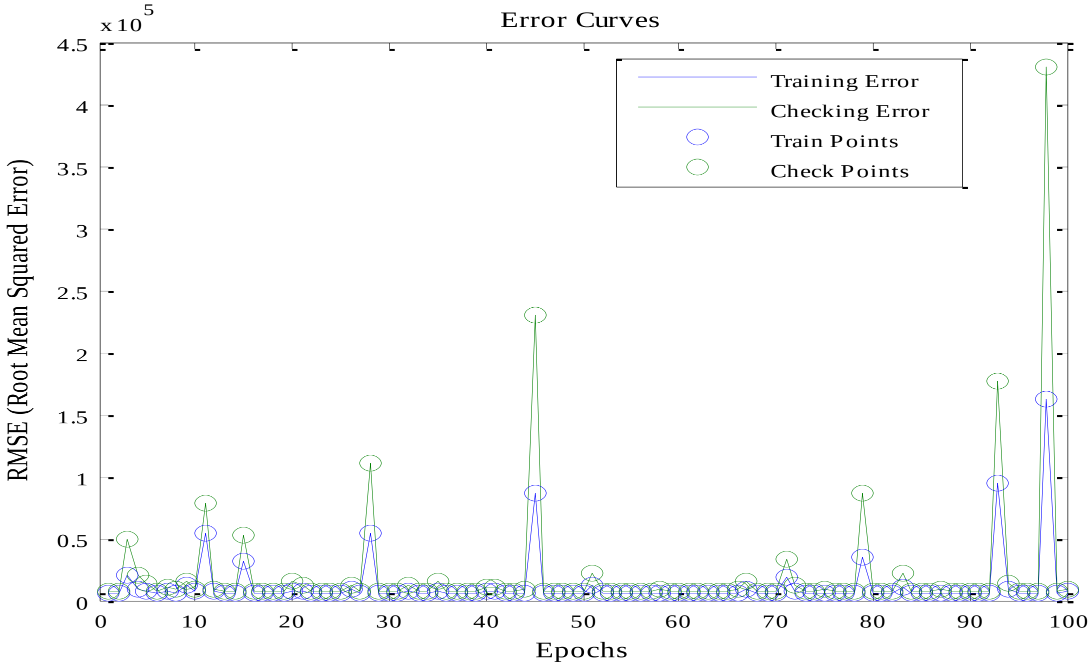

Figure 25: Error Curves (in Estimating System Weekly RLC)

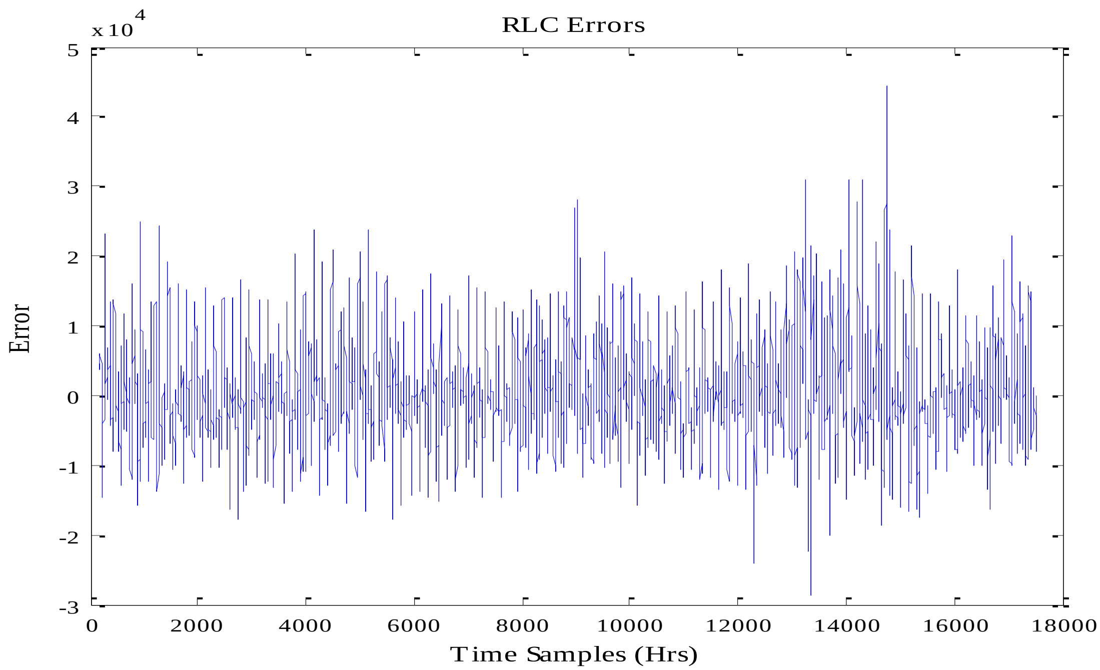

Figure 26: Error between System Load and Estimate RLC (intended for Weekly RLC)

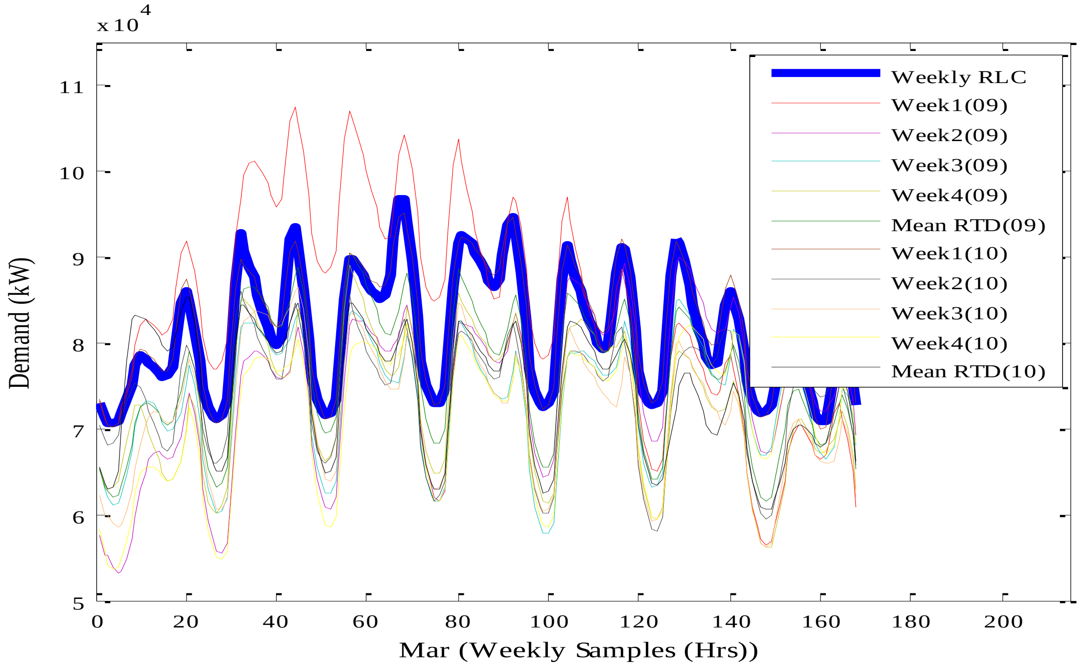

Figure 27: System March Estimate Weekly RLC

The figure above shows the estimate weekly RLC for the month of March for the overall system load using the historical dataset. This is based on our selection criterion which is to choose the section of the ANFIS output with the least MAPE. For this case, the 3rd week's estimate ANFIS output was used as the estimate RLC. As you can see, the estimate RLC can be used to represent the loads at those distinct periods during the year. A good recommendation to be used as pseudo-measurements in the DLE process.

6.2 Estimated Daily RLCs for the Overall System Load:

The ANFIS architecture/structure used is also the same architecture that was used in estimating the weekly RLCs for the overall system with the exception on the selection of inputs. The estimated RLC and system RTD is then shown in the figures below:

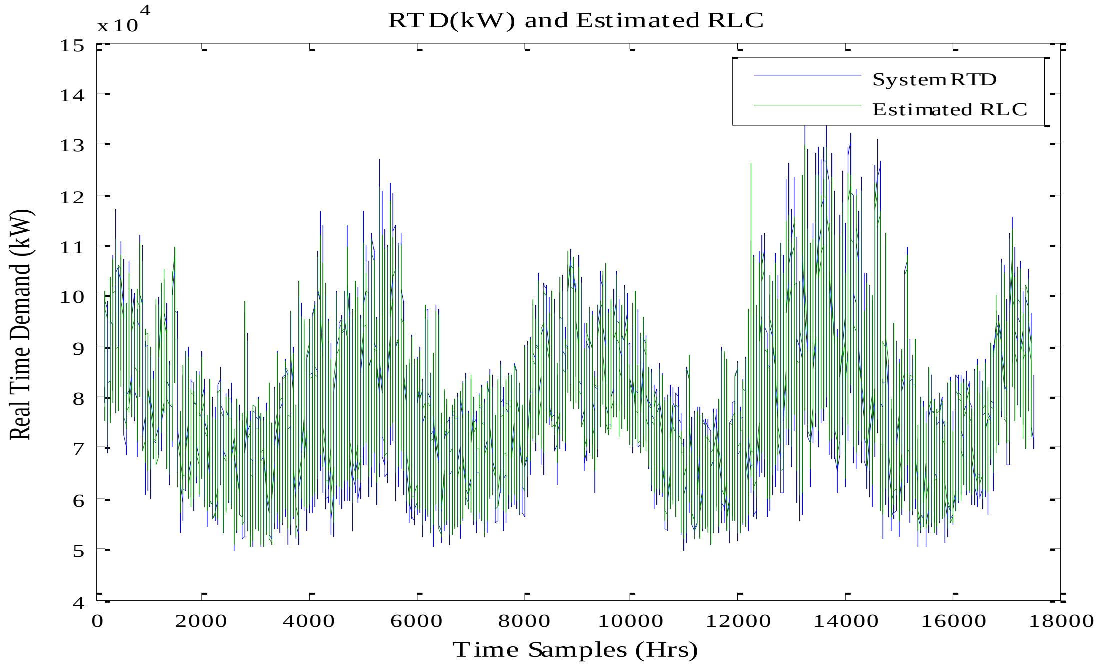

Figure 28: Estimate RLC and System RTD (intended for Daily RLC)

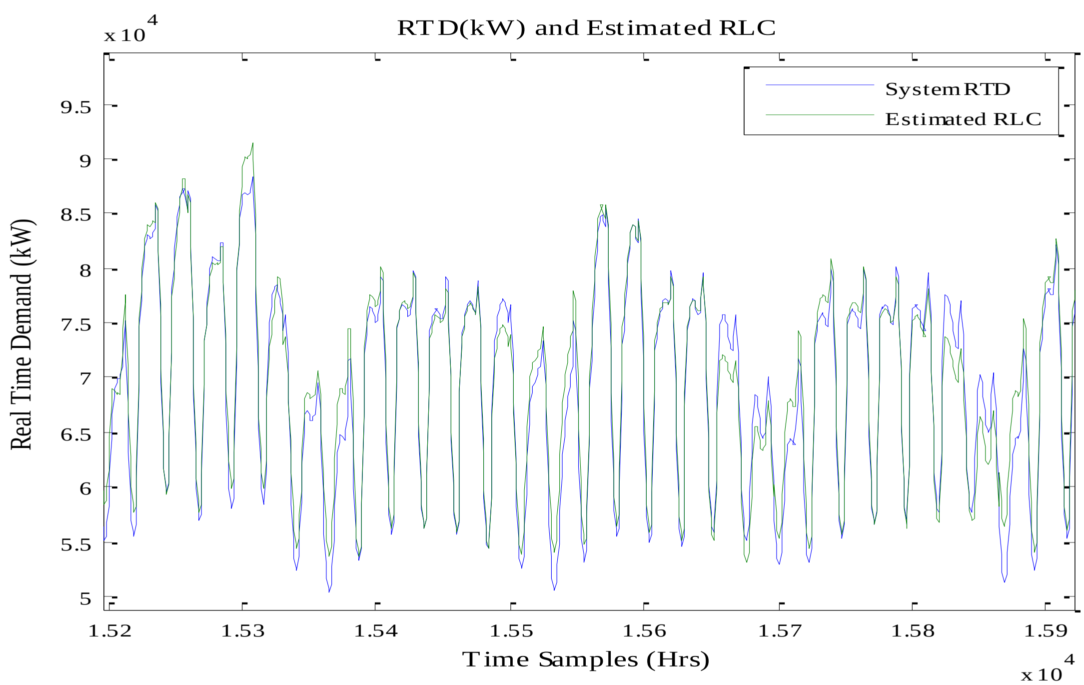

Figure 29: Magnified Estimate RLC and System RTD (intended for Daily RLC)

Overall MAPE between the estimated RLC and the system RTD is 2.9336%. The corresponding RMSE between estimated RLC and system RTD averaged at 5.7713×10^3 kW. The table below summarizes the methodology explained in the previous chapter where the weekly/daily estimated RLC is selected based on the weekly window/section with the least MAPE. The entries in the table show the different MAPEs in each week when the estimate RLC is compared with the system RTD. The ones boldly highlighted represents the estimate RLC for that period in the dataset.

Table 4: Tabular MAPE Results Comparing Each Week for Jan. Daily Estimate Substation RLC

	Year 2009				Year 2010			
	January							
Week/Windows	1	2	3	4	1	2	3	4
Day D1	NaN	2.100	4.0980	1.5486	1.2470	1.2843	2.1259	**0.8995**
Day D2	NaN	**1.3803**	2.3907	2.2058	2.4057	2.2927	3.1947	1.9121
Day D3	NaN	1.6378	1.2680	2.2933	2.8051	3.5780	1.0710	**0.8014**
Day D4	NaN	1.4141	3.3049	3.7446	2.6992	2.7294	**1.4017**	2.3400
Day D5	NaN	2.0072	5.0193	2.5138	2.5405	2.5680	**1.1774**	1.8731
Day D6	NaN	4.0507	3.3767	4.6439	**1.4748**	1.7575	4.4968	4.9008
Day D7	NaN	5.8436	3.9484	7.8143	6.2799	3.6019	5.9621	**3.0751**

Table 5: Tabular MAPE Results Comparing Each Week for Feb. Daily Estimate Substation RLC

	Year 2009				Year 2010			
	February							
Week/Windows	1	2	3	4	1	2	3	4
Day D1	4.6056	6.1339	**0.7002**	2.0433	5.9114	3.2603	5.5478	0.8293
Day D2	4.0153	1.1517	**0.7069**	1.5943	2.2975	3.2981	1.9862	2.0904
Day D3	2.1715	1.1615	1.4084	2.2315	4.6313	5.0774	1.0404	**0.5343**
Day D4	1.2082	1.8548	5.5954	2.0999	3.4324	1.3689	3.1634	**0.8018**
Day D5	1.5410	5.1379	1.9197	4.0904	2.8915	4.3186	4.4354	**0.5442**
Day D6	5.3638	1.9361	1.9096	2.2172	**1.5082**	5.0358	3.0692	9.7610
Day D7	5.5343	4.9675	2.8519	2.0770	**1.8719**	5.6203	3.5927	3.5516

Table 6: Tabular MAPE Results Comparing Each Week for Mar. Daily Estimate Substation RLC

	Year 2009				Year 2010			
	March							
Week/Windows	1	2	3	4	1	2	3	4
Day D1	**0.5754**	8.3058	1.9381	3.1936	1.0387	2.2705	4.1553	5.8714
Day D2	2.2323	6.5073	2.3728	0.5882	1.4388	**0.5136**	1.2570	1.1718
Day D3	2.6334	1.8336	1.7444	**0.8100**	1.1385	0.8662	2.7274	2.1249
Day D4	3.7641	1.9019	**0.9628**	2.9537	3.3542	1.6558	2.1178	2.2182
Day D5	2.1410	1.8792	1.2420	5.2401	3.2512	1.6016	**0.8776**	2.3837

	Year 2009				Year 2010			
Day D6	4.3748	2.9377	**0.9842**	1.8526	1.3084	2.1406	1.1592	3.4495
Day D7	**1.1082**	7.0723	2.7085	2.5397	4.1632	3.7401	2.2744	4.3073

We again proceed to show results only for the month of March as we won't display all the graphical representations within this thesis. The initial and adjusted membership functions for the inputs during ANFIS training are shown below:

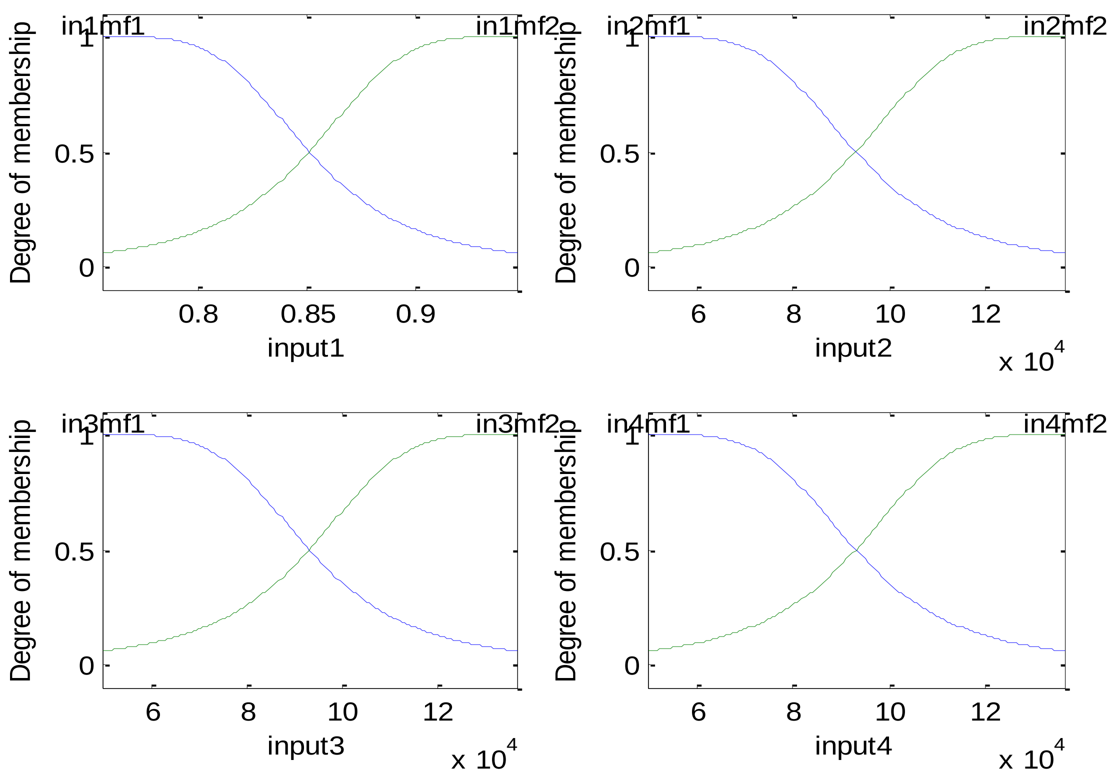

Figure 30: Input Membership Functions for Estimating System Daily RLC

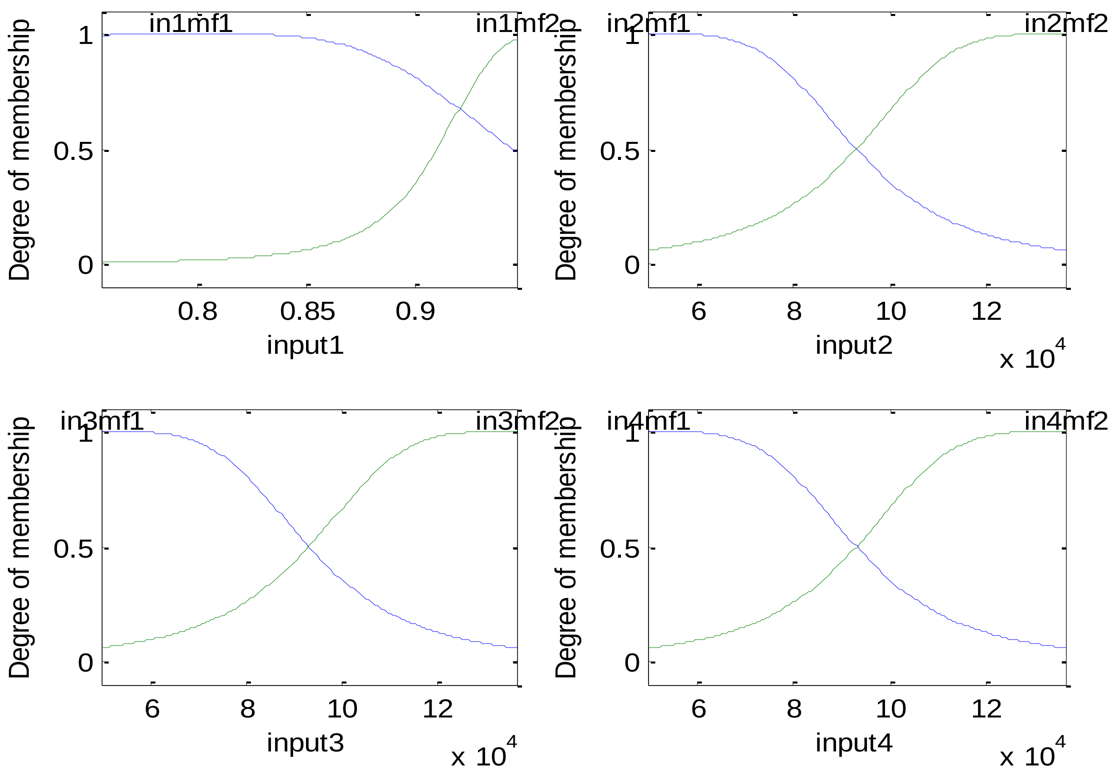

Figure 31: Adjusted Input Membership Functions for Estimating System Daily RLC

The figure below then shows how the errors of the training and checking datasets interact during the course of the ANFIS implementation.

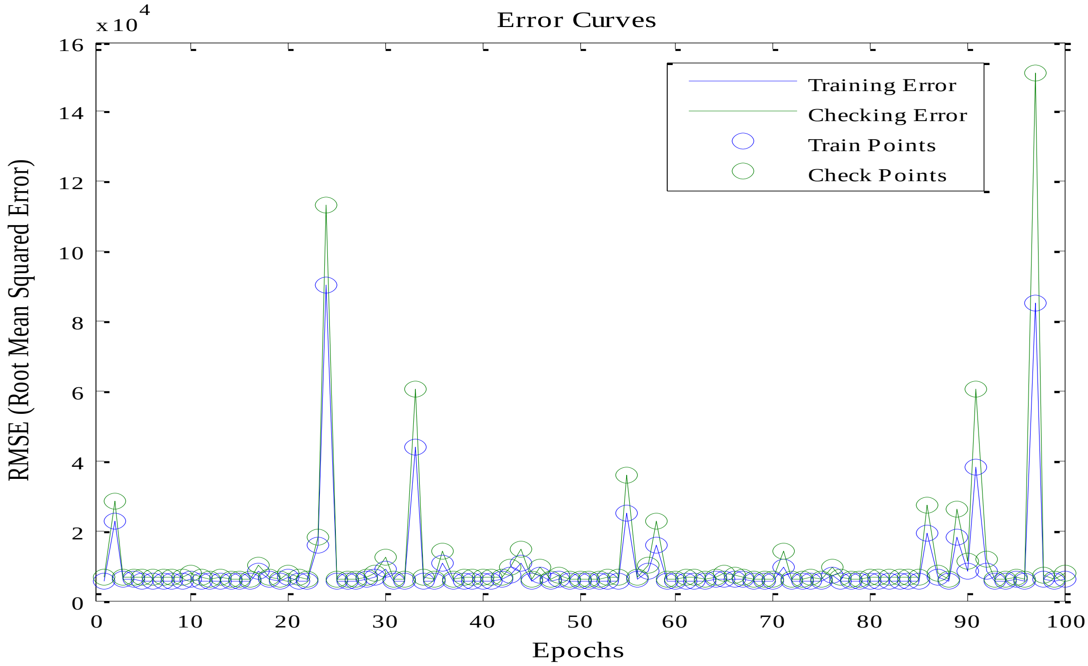

Figure 32: Error Curves (in Estimating System Daily RLC)

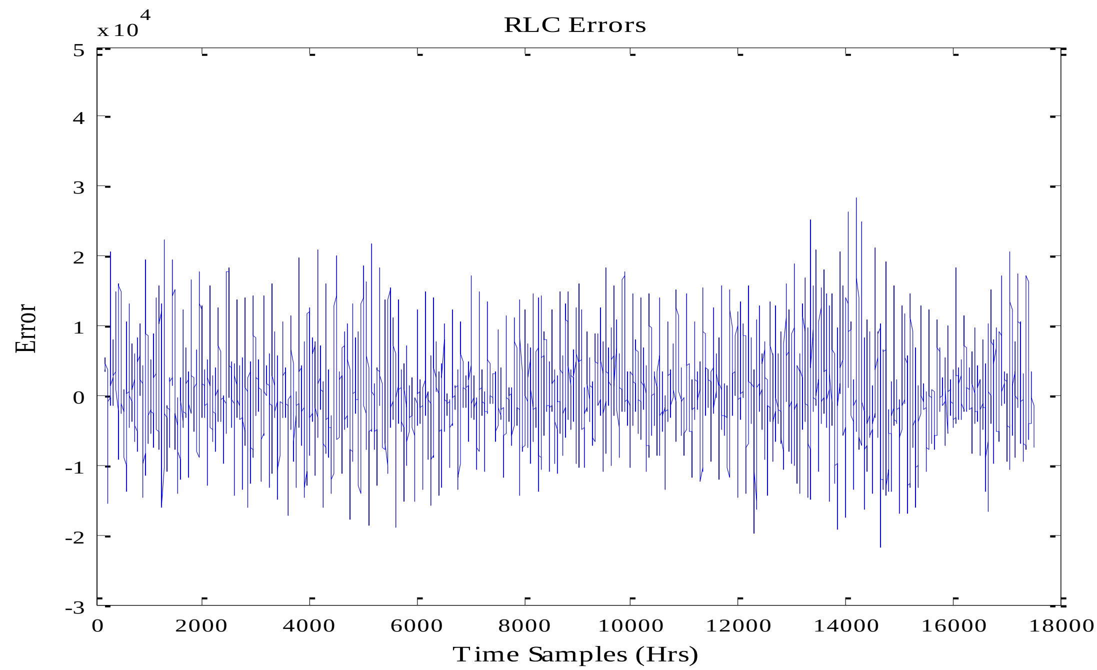

Figure 33: Error between System Load and Estimate RLC (intended for Daily RLC)

Now, we present the estimate daily RLCs for the month of March. Another recommended set as possible candidates as pseudo-measurements for DLE. Again the selection criterion in on the window/week with the least MAPE between the ANFIS output and original dataset for that specific period (March) in the dataset.

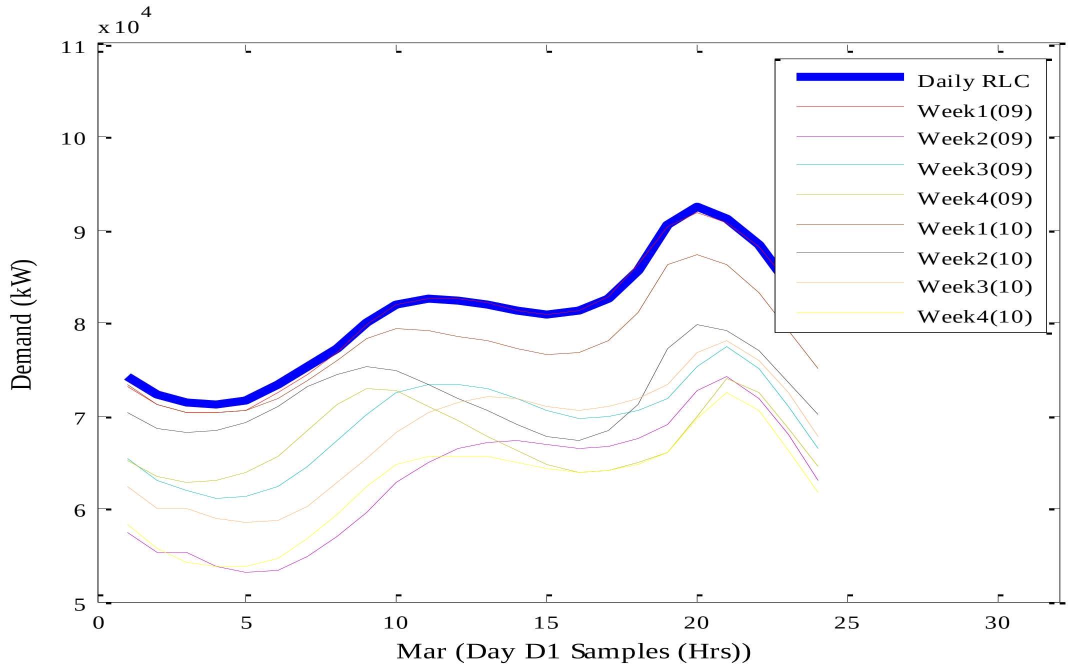

Figure 34: System, March Day D1 Estimate RLC

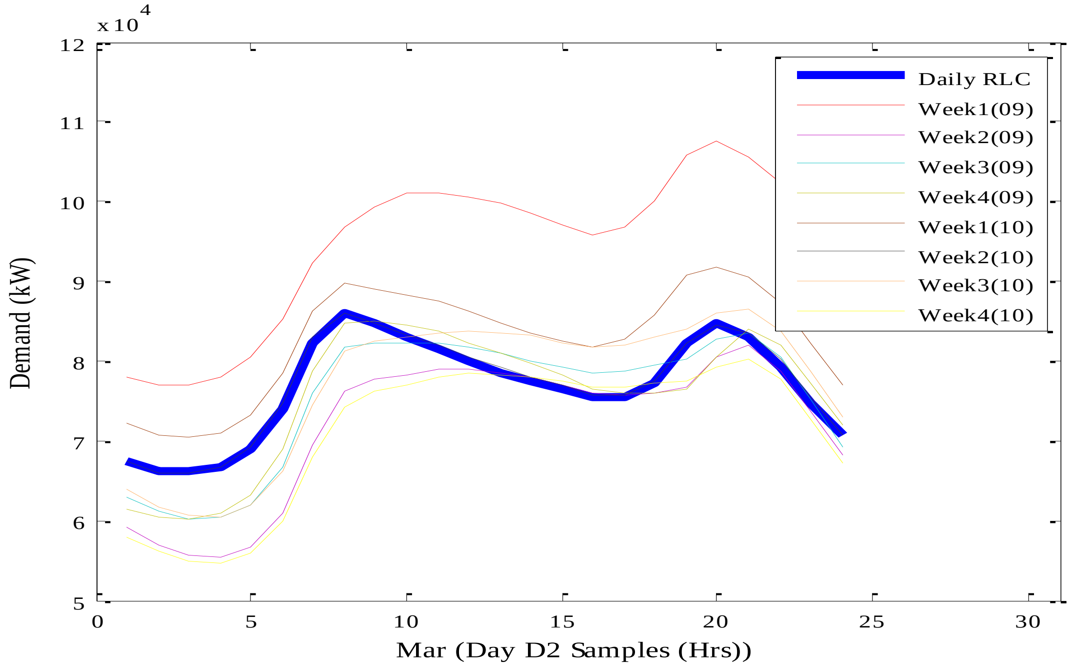

Figure 35: System, March Day D2 Estimate RLC

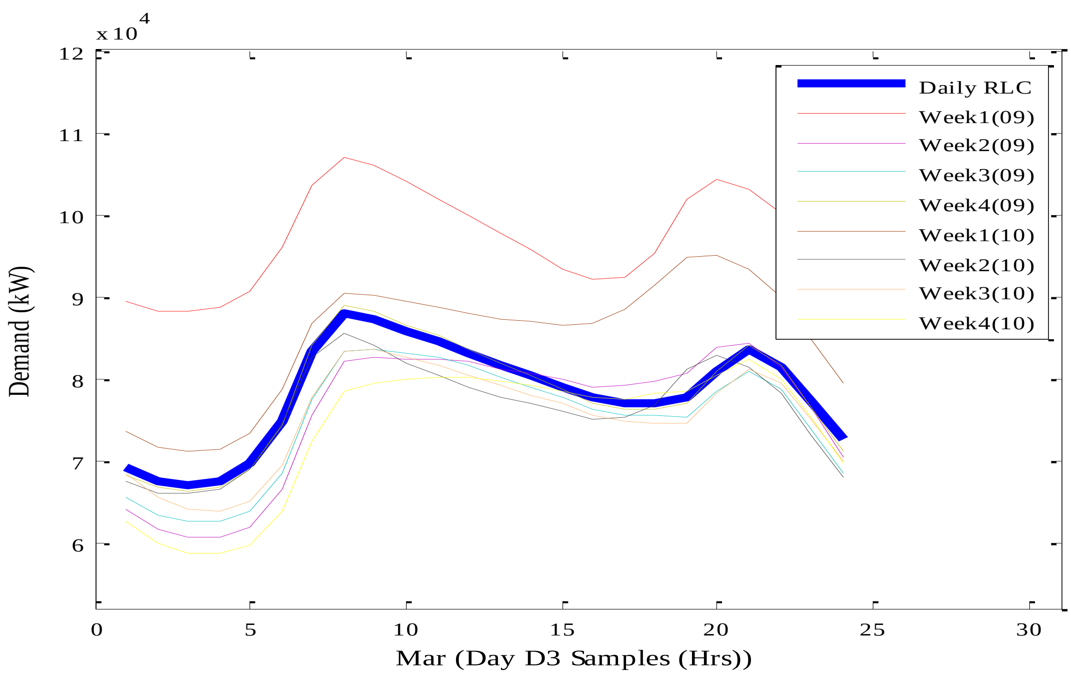

Figure 36: System, March Day D3 Estimate RLC

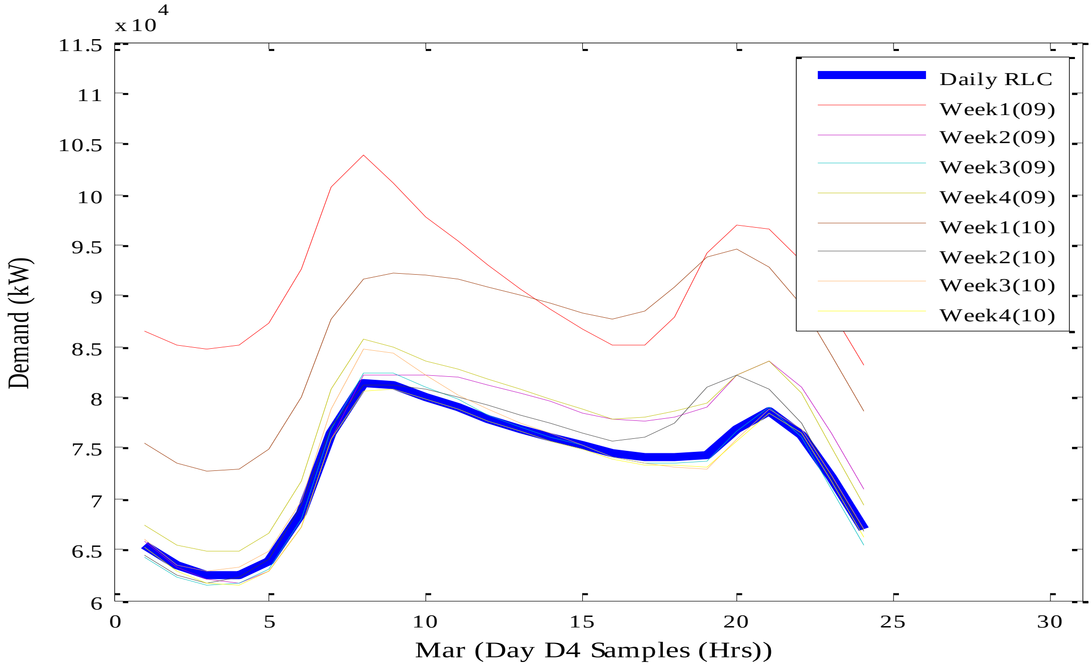

Figure 37: System, March Day D4 Estimate RLC

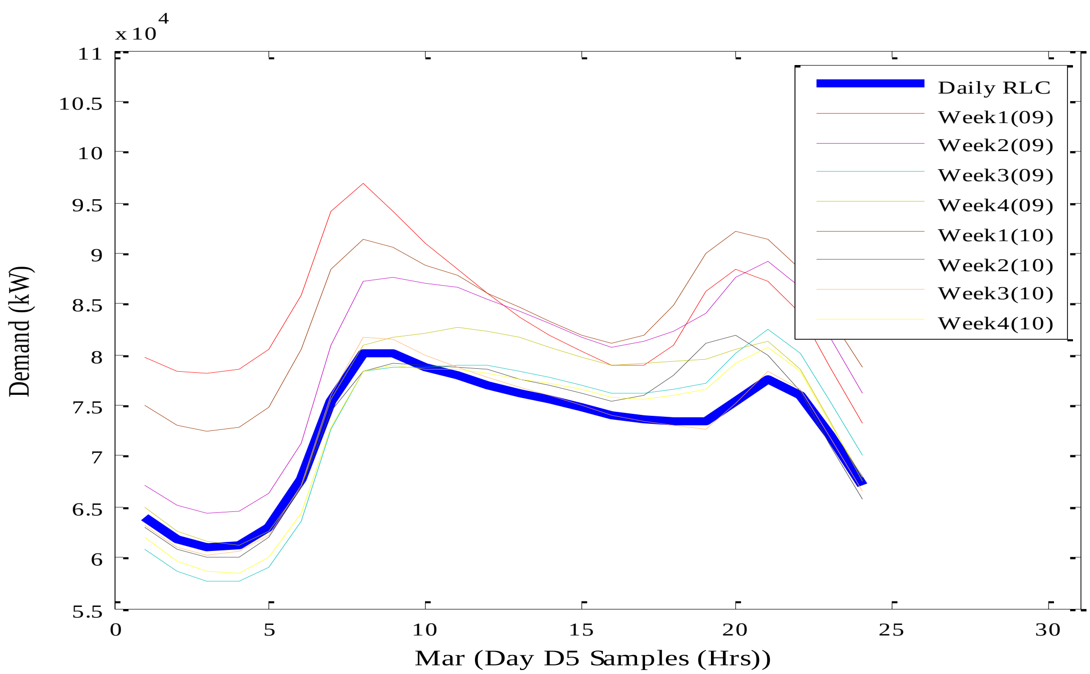

Figure 38: System, March Day D5 Estimate RLC

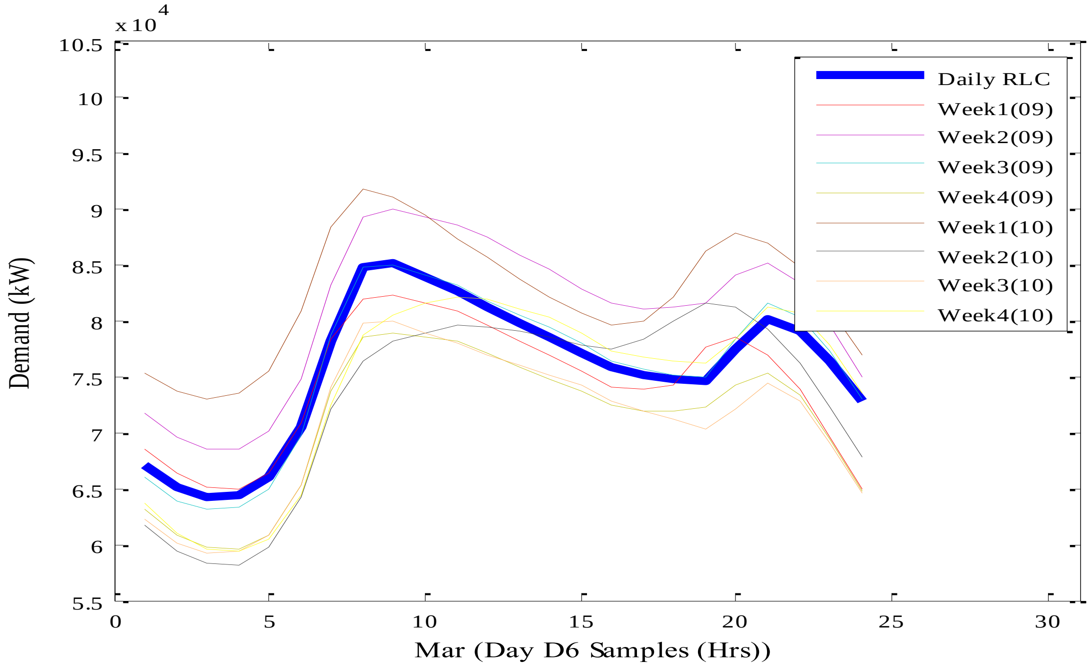

Figure 39: System, March Day D6 Estimate RLC

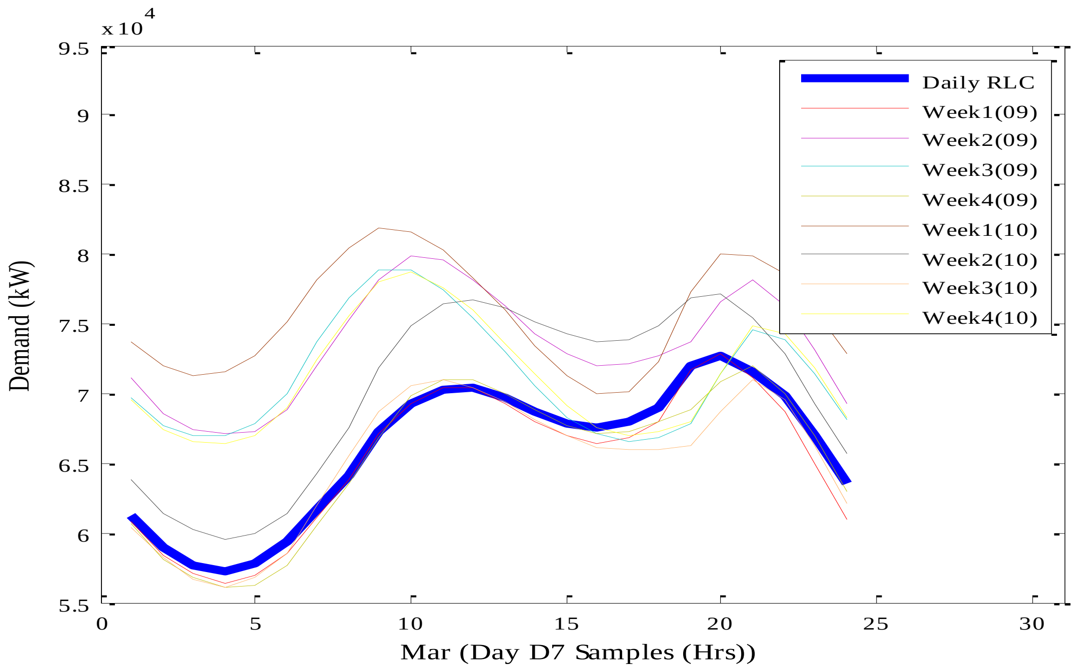

Figure 40: System, March Day D7 Estimate RLC

6.3 Estimated Weekly RLCs for the Selected Node 5 (CR)

The 2009-2010 Historical Node 5 (CR) Real Time Demand (RTD) is presented in the figure below.

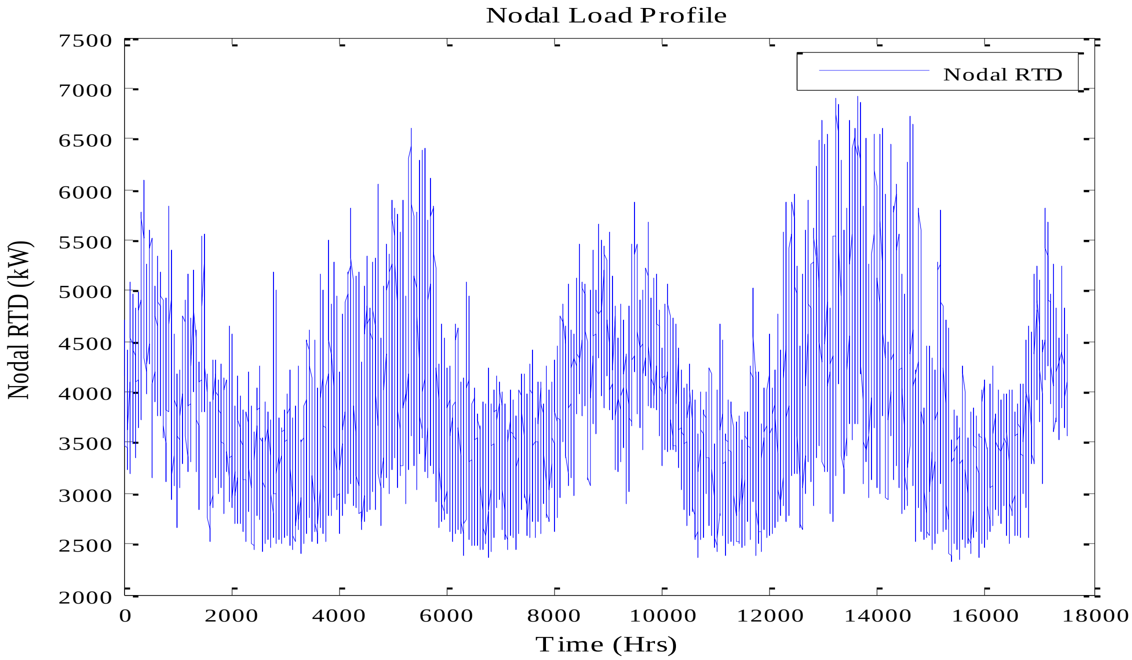

Figure 41: 2009-2010 Historical Node 5 (CR) Real Time Demand (RTD)

Upon training and testing the ANFIS, the following results were obtained. The historical RTD of the selected node being our target vector, whereas the estimated nodal RLC is the output of the ANFIS as explained in the previous chapter when its attainment was explained. The ANFIS architecture/structure used for this implementation is shown below and is also the same architecture that will be used in estimating the daily RLCs for the selected node (5) with the exception on the selection of inputs. This architecture can also be extended to other nodes. The estimated RLC and system RTD is then plotted.

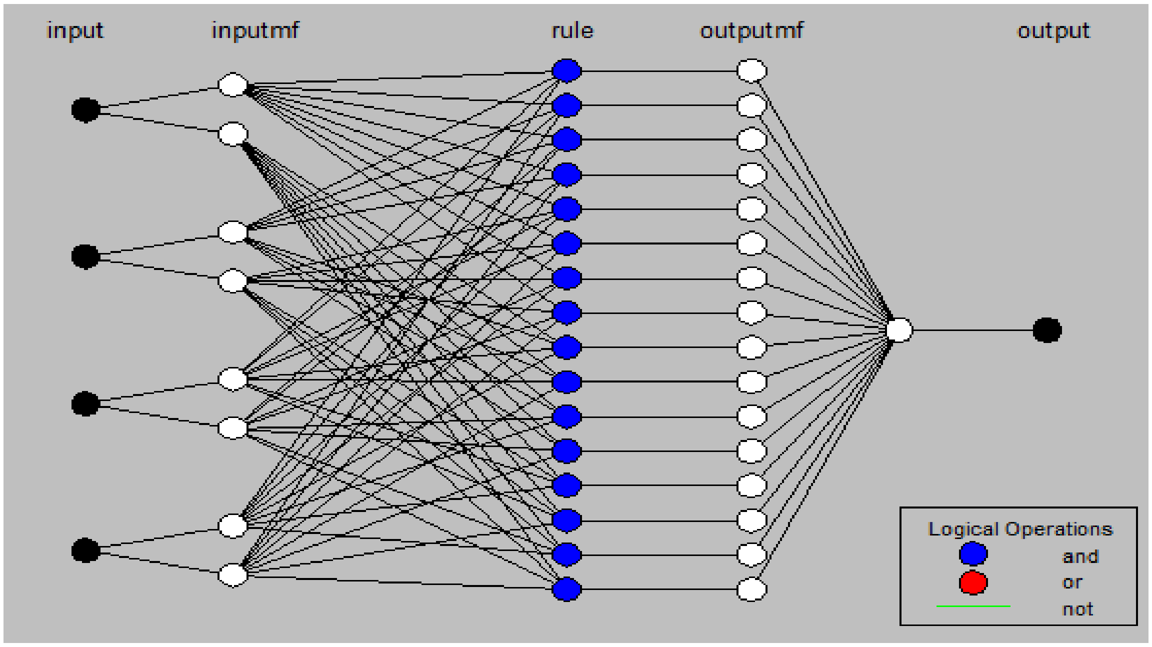

Figure 42: ANFIS Structure for Estimating RLCs for Node 5 (CR)

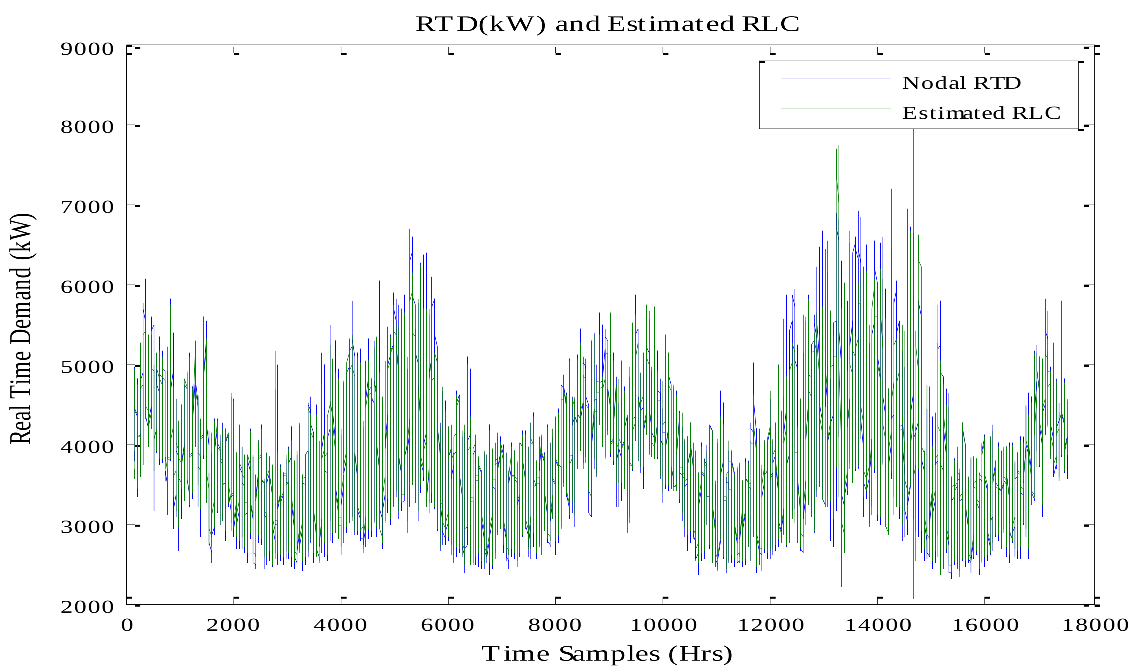

Figure 43: Estimate Nodal RLC and Nodal RTD (intended for Weekly RLC)

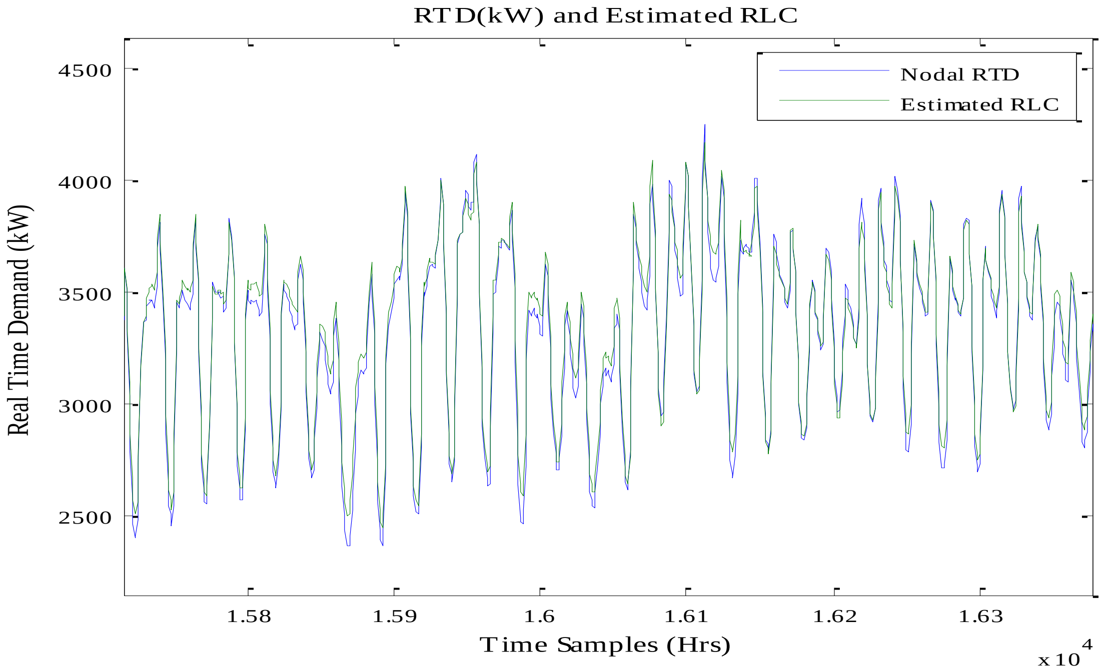

Figure 44: Magnified Estimate Nodal RLC and Nodal RTD (intended for Weekly RLC)

Overall MAPE between the estimated node 5 RLC and its RTD is 2.9950%. The corresponding RMSE between estimated RLC and nodal RTD averaged at 365.7375 kW. The table below summarizes the methodology explained in the previous chapter where the weekly/daily estimated RLC is selected based on the weekly window/section with the least MAPE. The entries in the table show the different MAPEs in each week when the estimate RLC is compared with the nodal RTD. The ones boldly highlighted represents the estimate RLC for that period in the dataset.

Table 7: Tabular MAPE Results Comparing Each Week for Weekly Estimate Nodal RLC

	Year 2009				Year 2010			
	January							
Week/Window	1	2	3	4	1	2	3	4
Estimate RLC MAPE	NaN	3.3852	3.3457	1.6465	3.3411	2.0590	2.7015	**1.5196**
	February							
Week/Window	1	2	3	4	1	2	3	4
Estimate RLC MAPE	4.3430	4.3857	1.6303	2.3567	2.1592	2.1784	**1.5875**	1.6407
	March							
Week/Window	1	2	3	4	1	2	3	4
Estimate RLC MAPE	4.3457	3.0880	1.1465	1.6570	**0.8333**	1.8832	2.0675	1.9950

We proceed to show results for node 5's estimated RLC for the month of February. Bear in mind that RLC are used only in load variation studies and can be used as pseudo-measurements alongside real-time measurements for DLE. The initial and adjusted input membership functions for the ANFIS training is shown in the figures below:

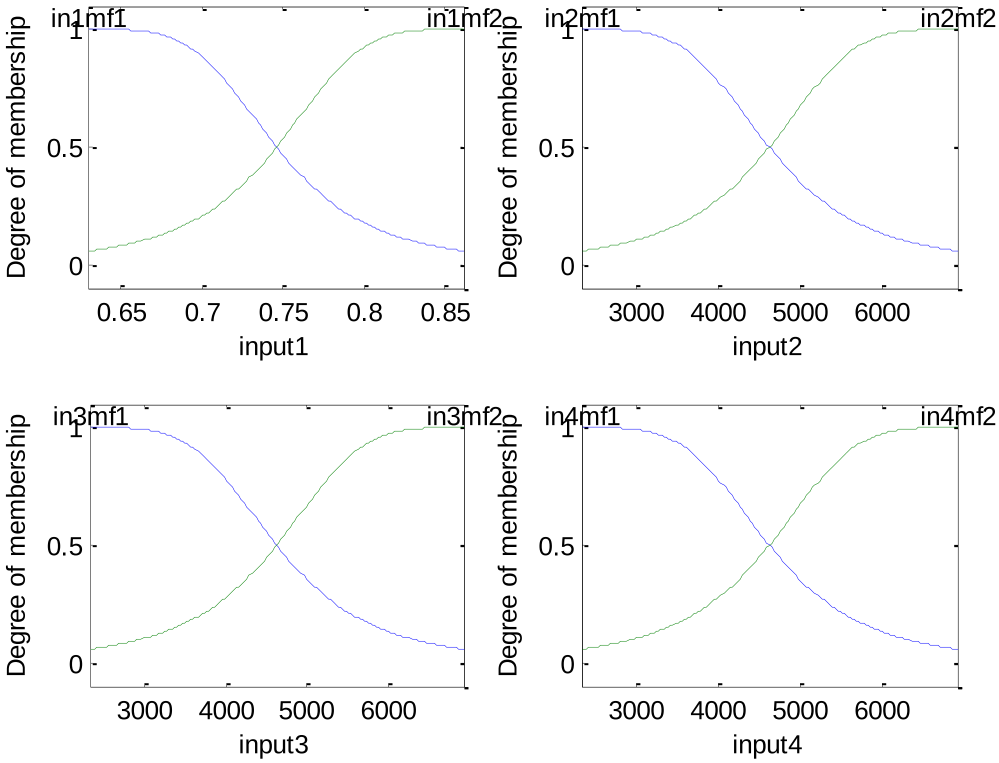

Figure 45: Initial Input Membership Functions for Estimating Node 5 (CR) Weekly RLC

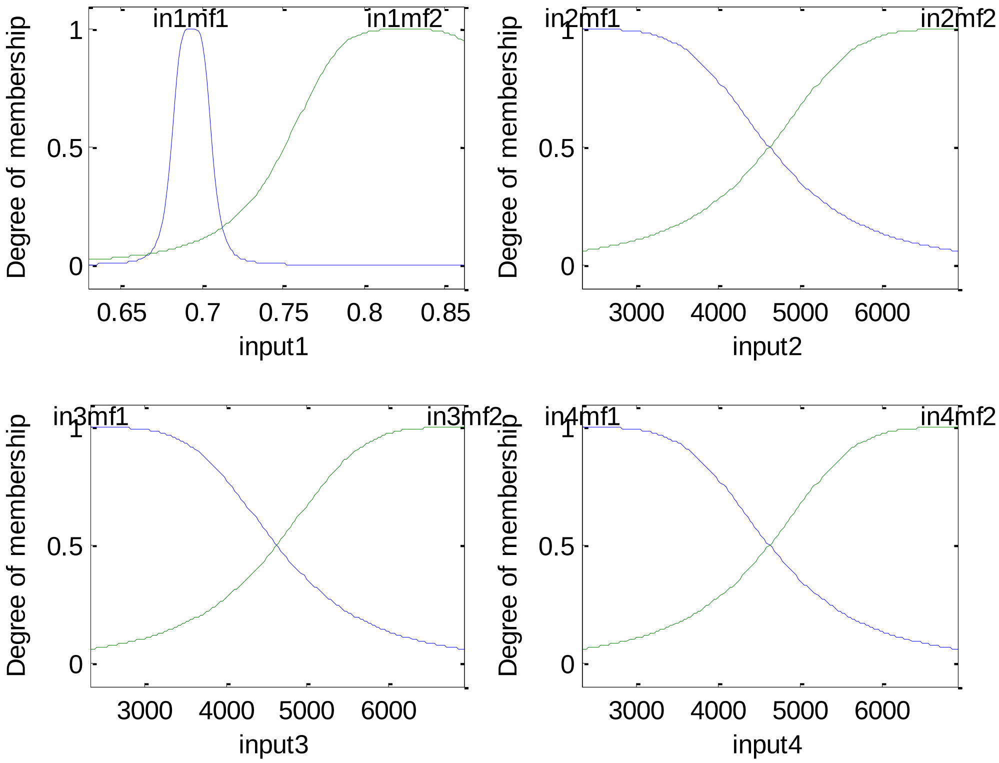

Figure 46: Adjusted Input Membership Functions for Estimating Node 5 (CR) Weekly RLC

The figure shown below further shows how the errors of the training and checking dataset interact with each other. The FIS is selected using the MATLAB software just before over-fitting occurs.

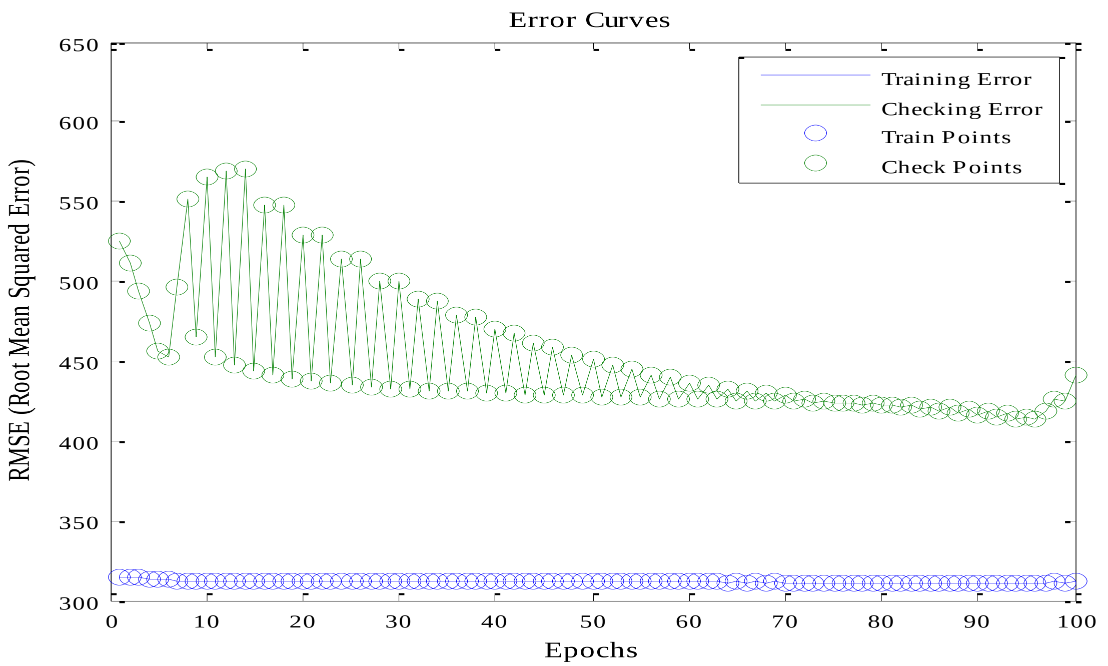

Figure 47: Error Curves (in Estimating Node 5 (CR) Weekly RLC)

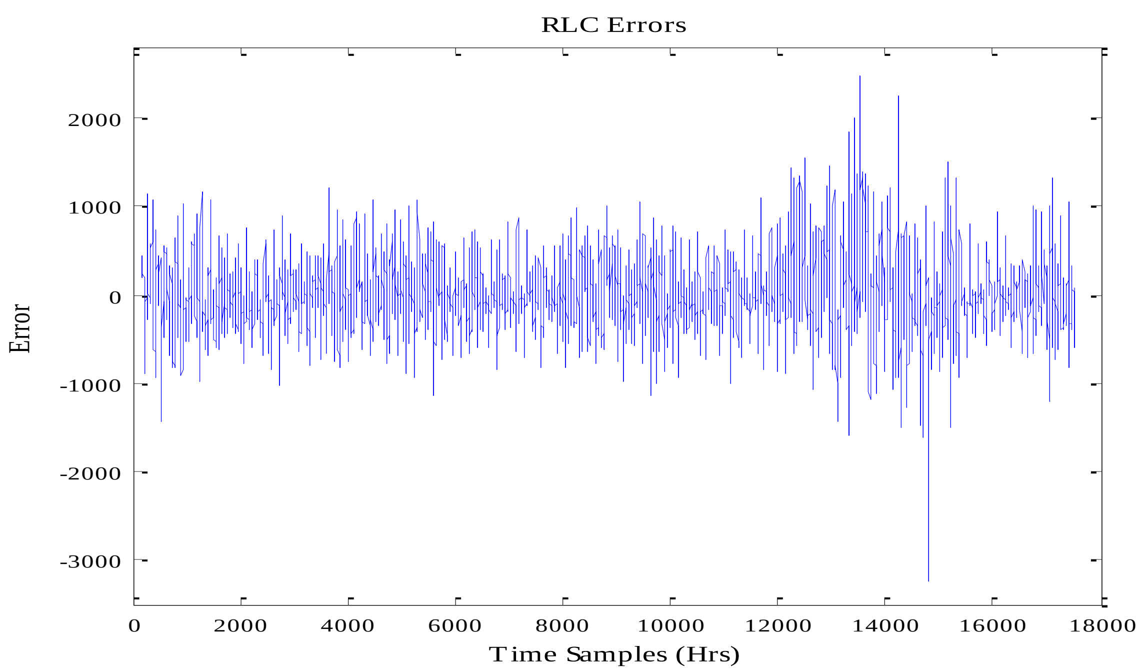

Figure 48: Error between Nodal Load and Nodal Estimate RLC (intended for Weekly RLC)

The estimate weekly RLC for node 5 (CR) for the month of February is shown in the figure below. As suggested earlier, another good candidate as pseudo-measurements in the DLE process. Based on the same selection criterion; the MAPE between the ANFIS output and original dataset must be the least for the select period (February in this case)

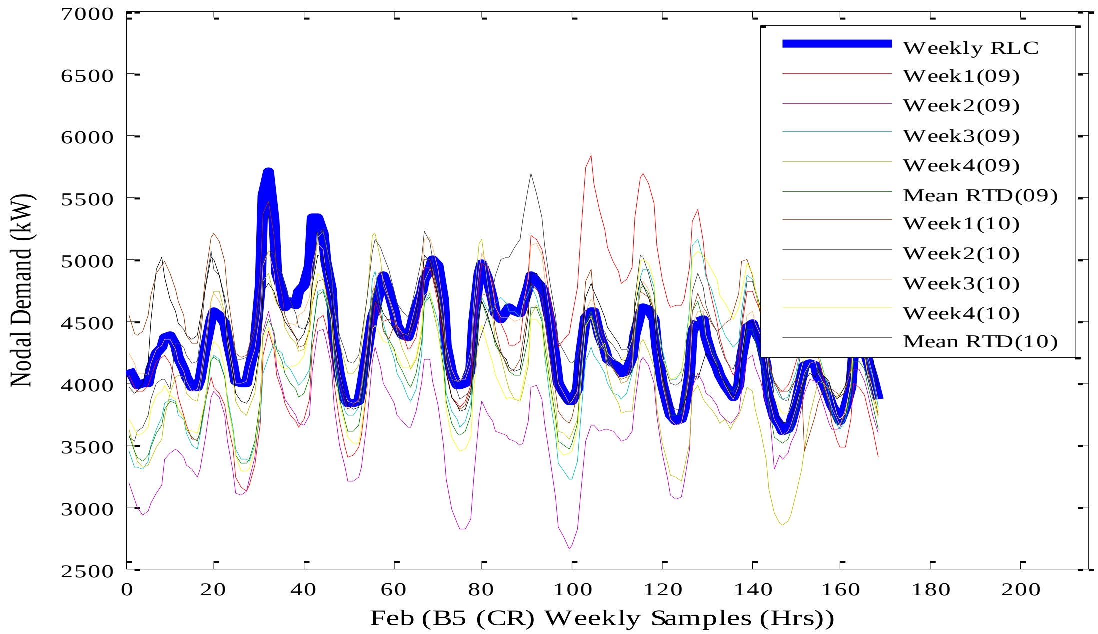

Figure 49: Node 5 (CR) February Estimate Weekly RLC

6.4 Estimated Daily RLCs for the Selected Node 5 (CR)

The ANFIS architecture/structure used is also the same architecture that was used in estimating the daily RLCs for node 5 with the exception on the selection of inputs. This architecture can also be extended to other nodes. The estimated RLC and system RTD is then shown in the figures below:

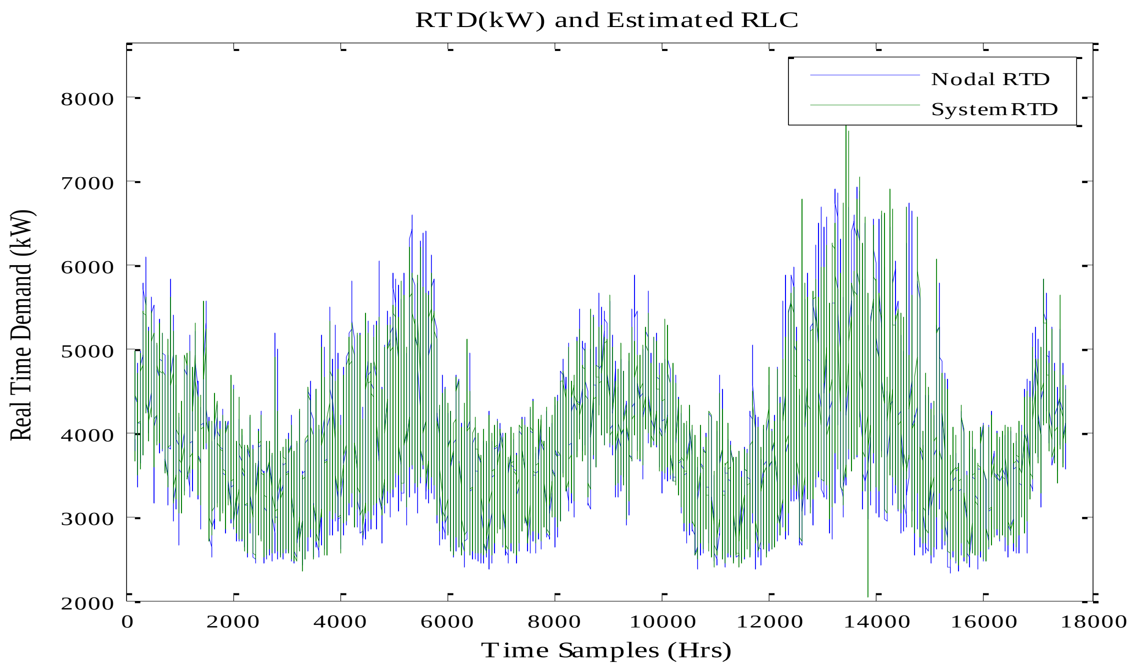

Figure 50: Estimate Nodal RLC and Nodal RTD (intended for Daily RLC)

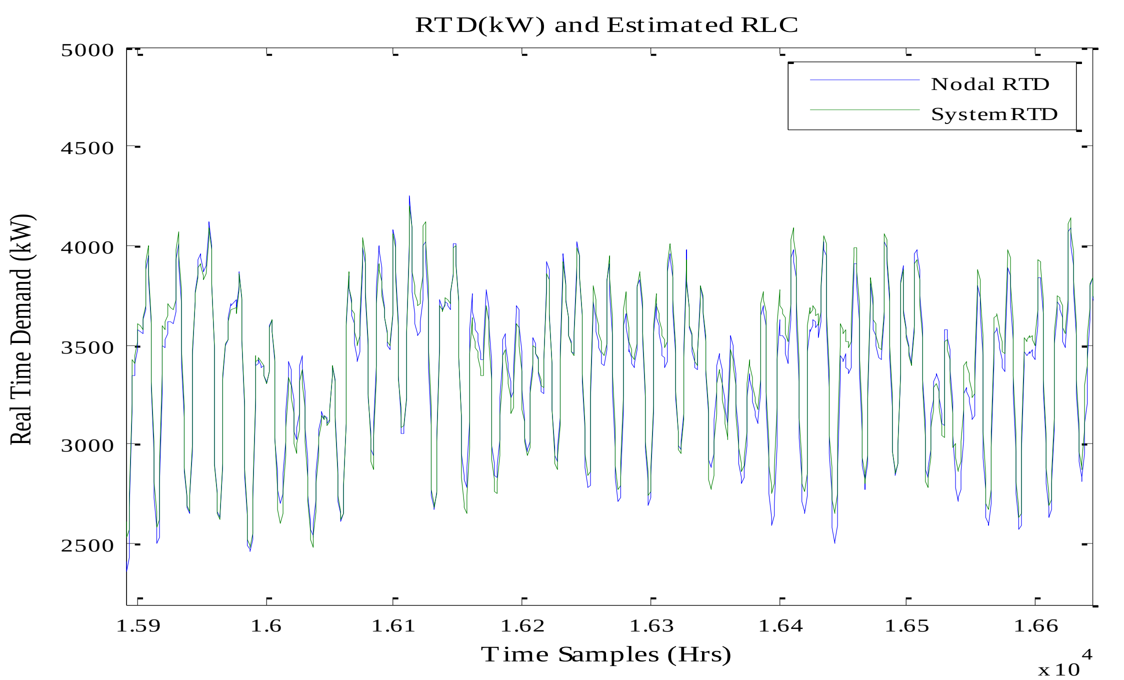

Figure 51: Magnified Estimate Nodal RLC and Nodal RTD (intended for Daily RLC)

Overall MAPE between the estimated RLC and the nodal RTD is 3.3523%. The corresponding RMSE between node 5's estimated RLC and its historical RTD averaged at 363.0236 kW. The table below summarizes the methodology explained in the previous chapter where the weekly/daily estimated RLC is selected based on the weekly window/section with the least MAPE. The entries in the table show the different MAPEs in each week when the estimate RLC is compared with the system RTD. The ones boldly highlighted represents the estimate RLC for that period in the dataset.

Table 8: Tabular MAPE Results Comparing Each Week for Node 5's Jan. Daily Estimate RLC

	Year 2009				Year 2010			
	January							
Week/Windows	1	2	3	4	1	2	3	4
Day D1	NaN	1.9540	**1.1120**	2.6517	9.5528	1.1234	2.3791	1.4082
Day D2	NaN	1.9059	1.7919	**0.7331**	5.1481	1.9840	2.8293	1.2859
Day D3	NaN	2.0474	3.4258	**1.2924**	2.0392	1.3343	3.5157	3.1603
Day D4	NaN	1.3174	2.0333	2.6016	1.3700	1.6750	4.5142	**0.6895**
Day D5	NaN	3.4837	3.9690	1.4075	2.2991	1.9586	1.6994	**0.6998**
Day D6	NaN	**1.5937**	5.7308	1.5629	3.2256	3.6198	2.4584	4.5544
Day D7	NaN	2.0871	8.6491	4.1422	2.9894	**1.4300**	1.8548	9.1912

Table 9: Tabular MAPE Results Comparing Each Week for Node 5's Feb. Daily Estimate RLC

	Year 2009				Year 2010			
	February							
Week/Windows	1	2	3	4	1	2	3	4
Day D1	1.7732	4.2669	1.4944	5.0127	6.0279	2.9700	2.3207	**0.9795**
Day D2	6.6388	3.9354	1.4577	1.1103	6.2320	2.2350	**0.6429**	7.0284
Day D3	3.4514	2.3805	1.2573	1.4168	2.2866	2.8781	**1.0093**	1.2379
Day D4	**1.1083**	5.8363	5.0720	2.1094	1.7781	2.8253	2.1944	1.1841
Day D5	2.3480	9.2324	4.2115	1.3516	**0.5825**	2.3396	1.2290	3.5723
Day D6	3.0805	**1.0992**	3.4146	1.5397	1.4747	2.0677	1.7191	3.9697
Day D7	1.5741	3.0554	1.6518	4.7604	2.8461	2.6736	1.2455	**1.1198**

Table 10: Tabular MAPE Results Comparing Each Week for Node 5's Mar. Daily Estimate RLC

	Year 2009				Year 2010			
	March							
Week/Windows	1	2	3	4	1	2	3	4
Day D1	1.1667	6.8987	2.4013	1.5234	0.7038	**0.6951**	3.1086	6.4498
Day D2	1.8415	7.4273	1.2013	1.3149	**0.7513**	2.4538	1.8362	3.2604
Day D3	3.6501	3.8302	0.8186	1.3025	1.0904	2.8879	**0.3902**	3.8914
Day D4	3.7633	3.1597	1.0669	1.9249	1.9682	2.0470	2.9145	**0.9124**
Day D5	3.3806	4.5007	1.0980	1.9485	**0.8805**	2.4557	2.9120	0.8768

	Year 2009				Year 2010			
Day D6	0.9463	2.9860	**0.9342**	1.2291	1.0238	2.6320	0.9712	1.0596
Day D7	2.0936	4.0323	1.3098	1.9184	2.4958	1.2767	**0.6556**	3.7581

We again proceed to show results only for the month of February as we won't display all the graphical representations within this thesis. We'd begin by showing the input membership functions of the initially generated FIS, and its corresponding adjusted input membership functions after ANFIS training.

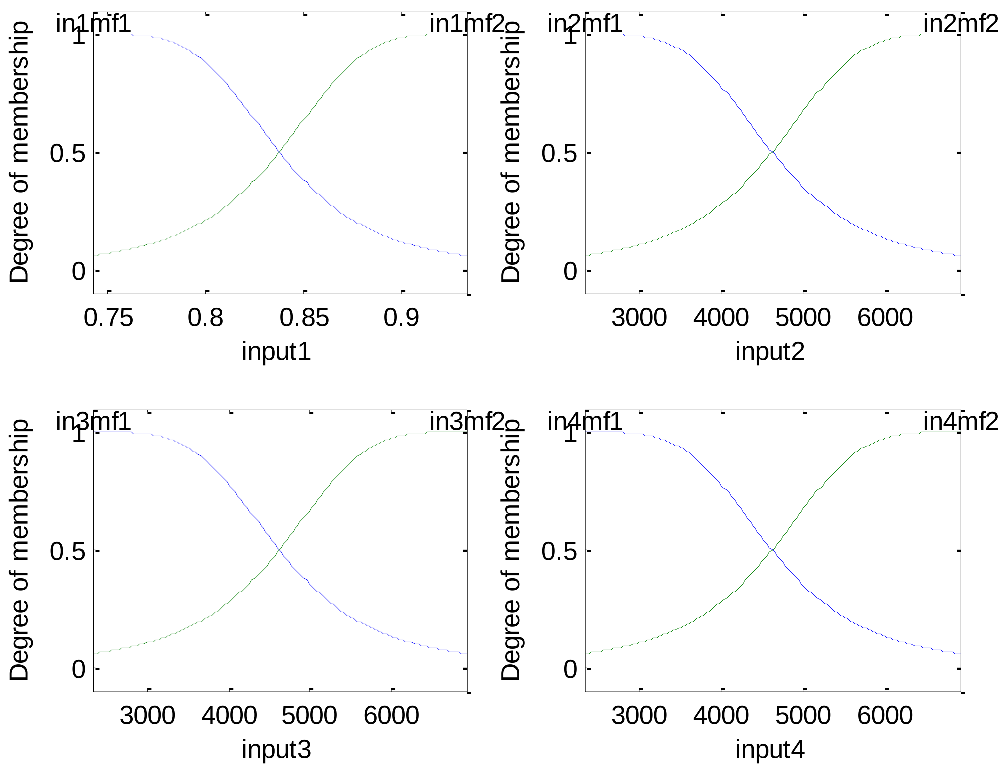

Figure 52: Initial Input Membership Functions for Estimating Node 5 (CR) Daily RLC

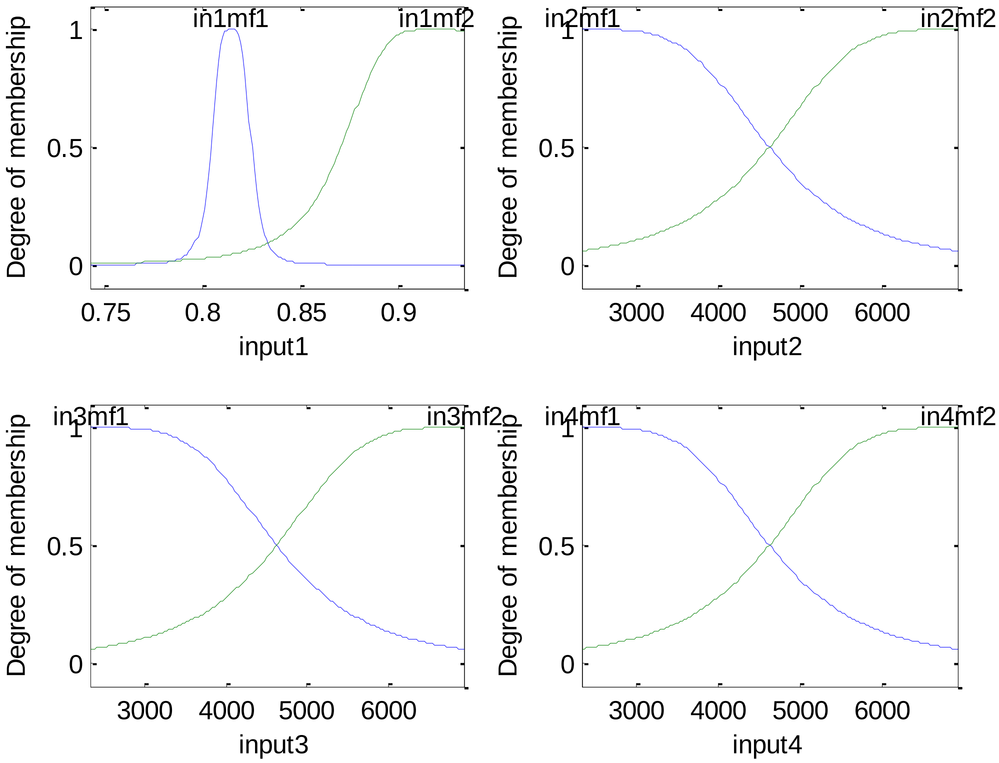

Figure 53: Adjusted Input Membership Functions for Estimating Node 5 (CR) Daily RLC

During the course of the ANFIS training, we'd also how the training and checking errors interact with each other. It's important to know that with the aid of MATLAB, the snapshot FIS optimized is the one just before over-fitting occurs.

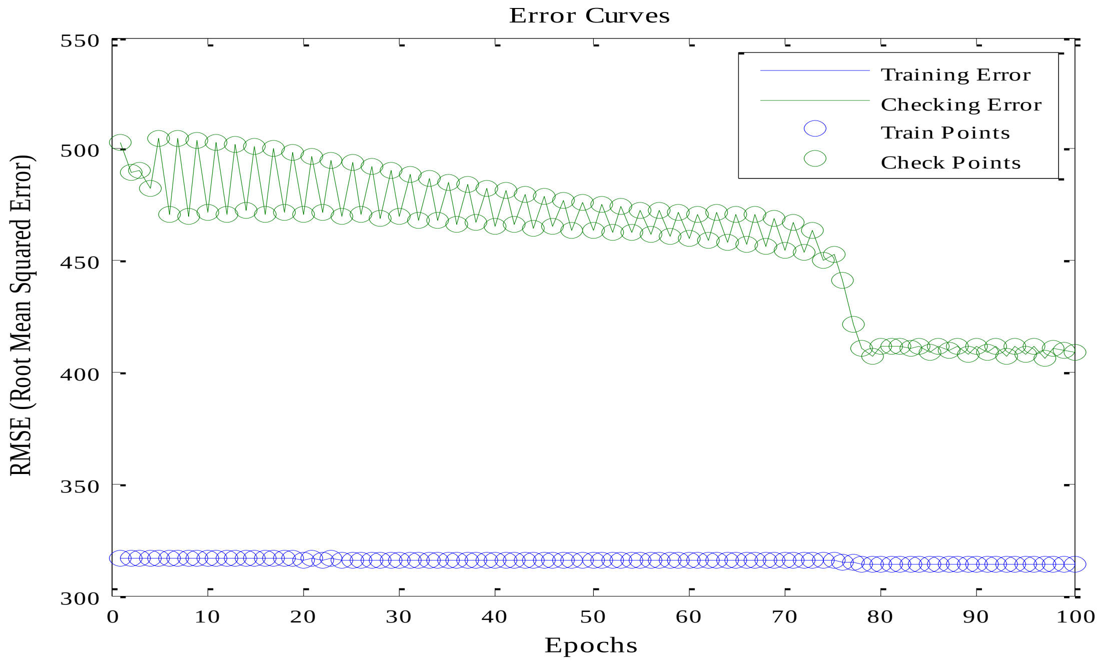

Figure 54: Error Curves (in Estimating Node 5 (CR) Daily RLC)

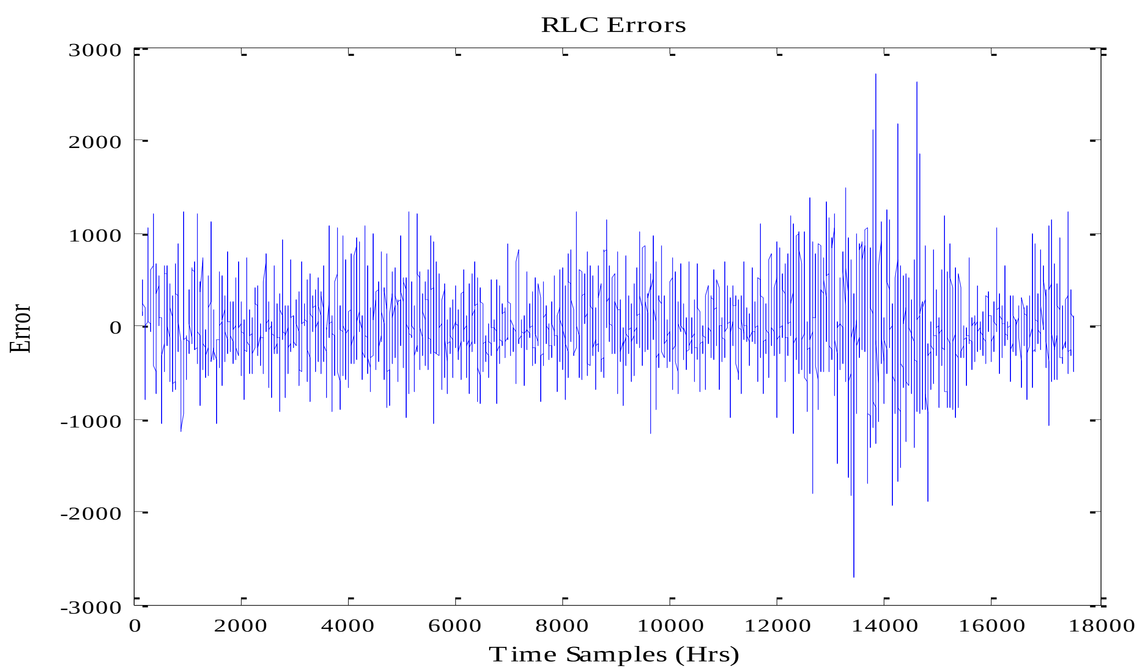

Figure 55: Error between Nodal Load and Nodal Estimate RLC (intended for Daily RLC)

The figures below show the daily estimated RLCs for node 5 (CR) in the month of February. Again, possible candidates for pseudo-measurements in the DLE process. Same critrion as before: the MAPE between the ANFIS output (estimate RLC) and original dataset has to be the least for the select period.

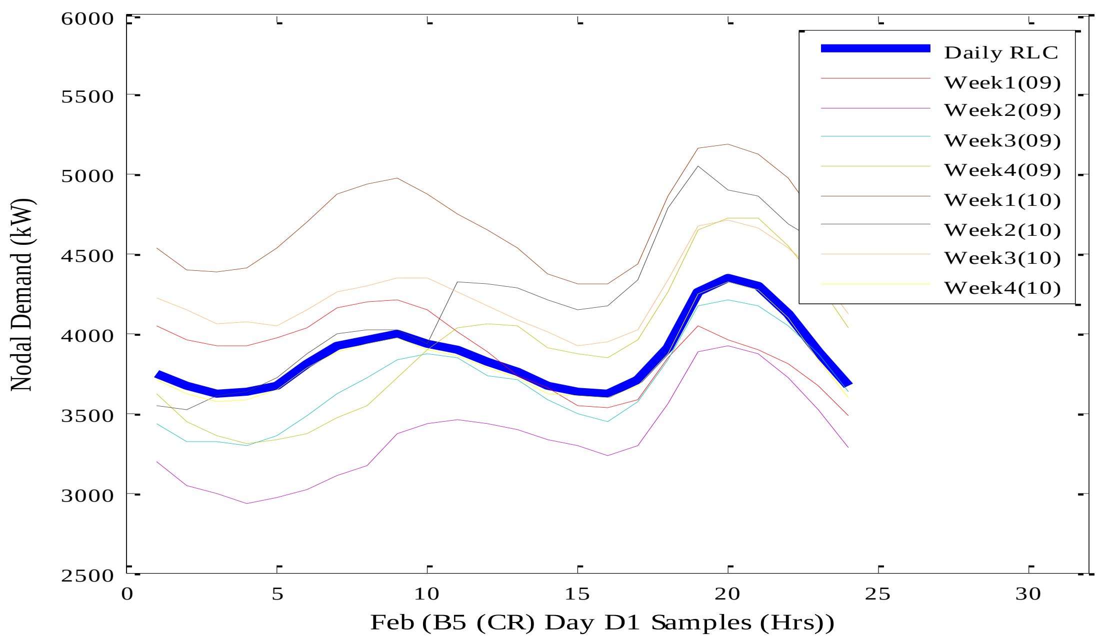

Figure 56: Node 5 (CR), February Day D1 Estimate RLC

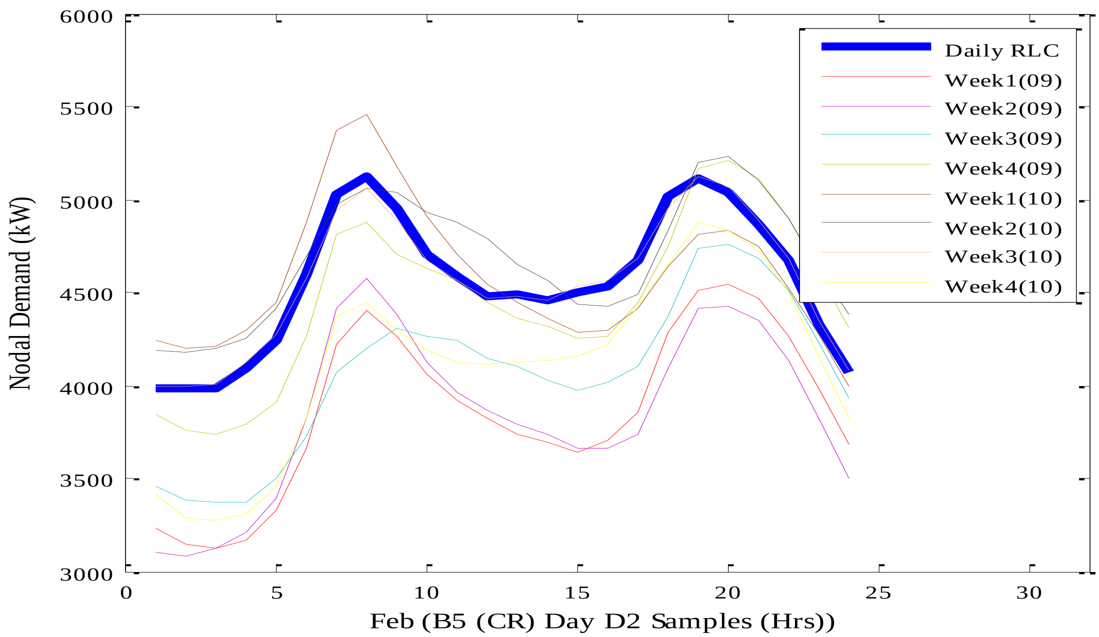

Figure 57: Node 5 (CR), February Day D2 Estimate RLC

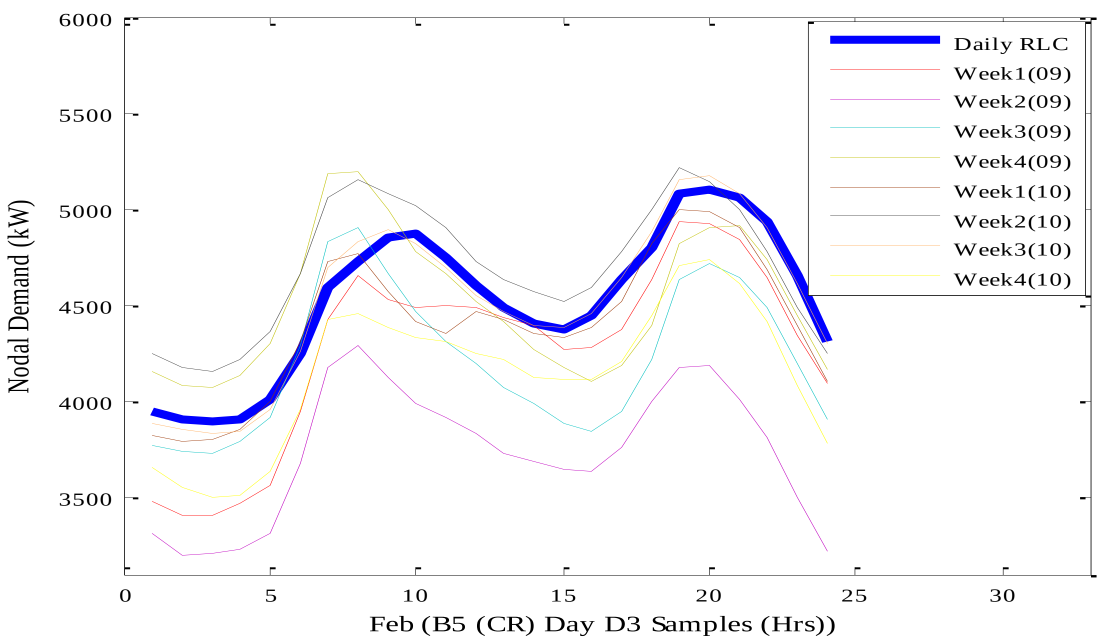

Figure 58: Node 5 (CR), February Day D3 Estimate RLC

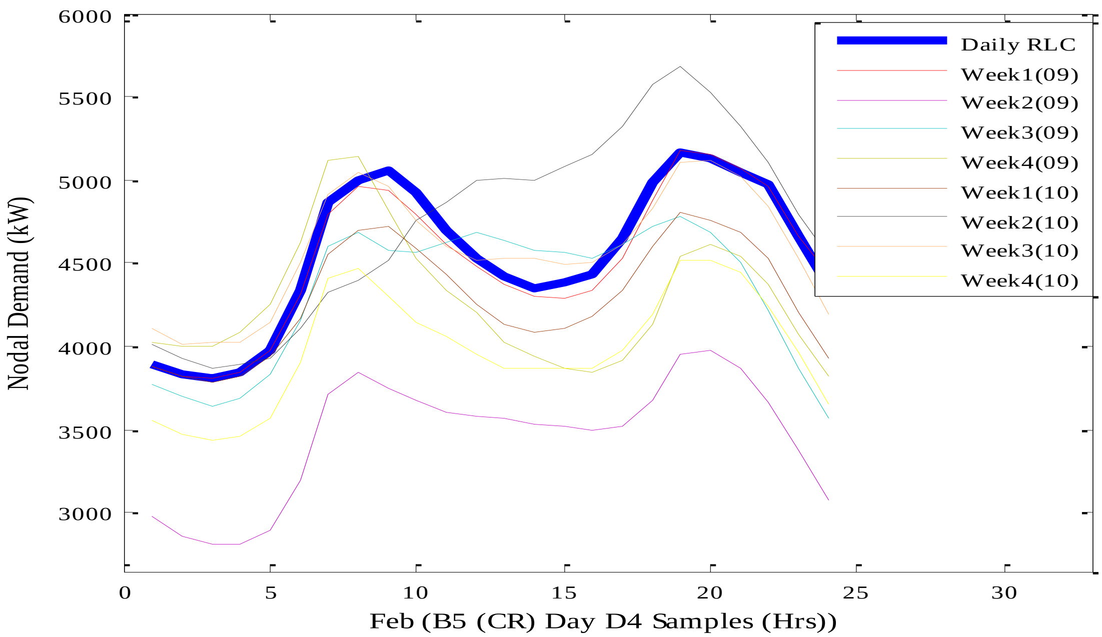

Figure 59: Node 5 (CR), February Day D4 Estimate RLC

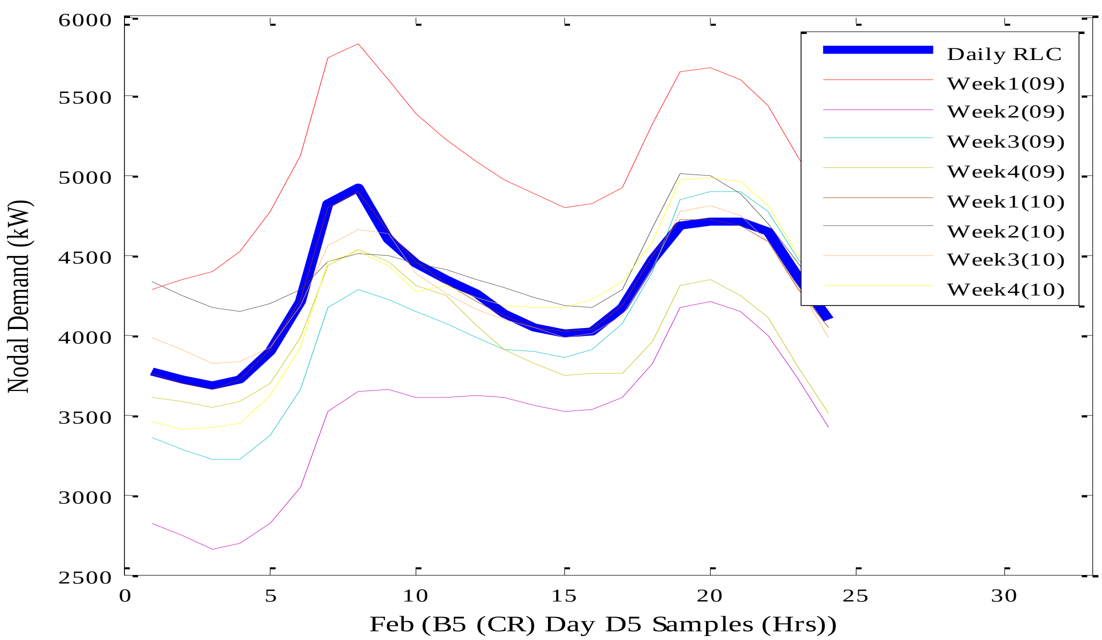

Figure 60: Node 5 (CR), February Day D5 Estimate RLC

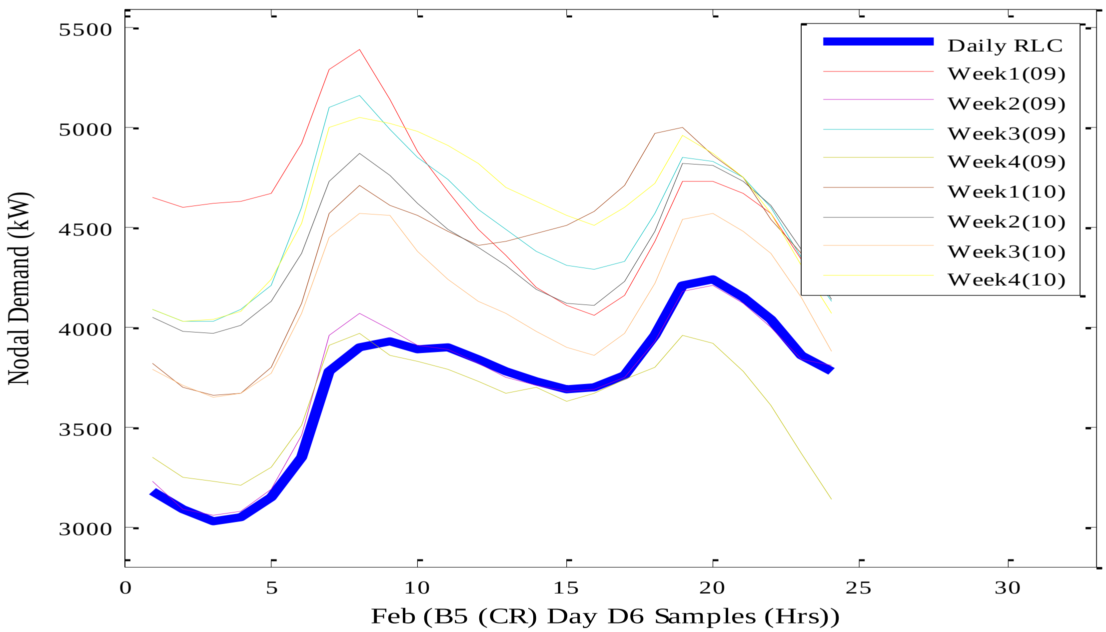

Figure 61: Node 5 (CR), February Day D6 Estimate RLC

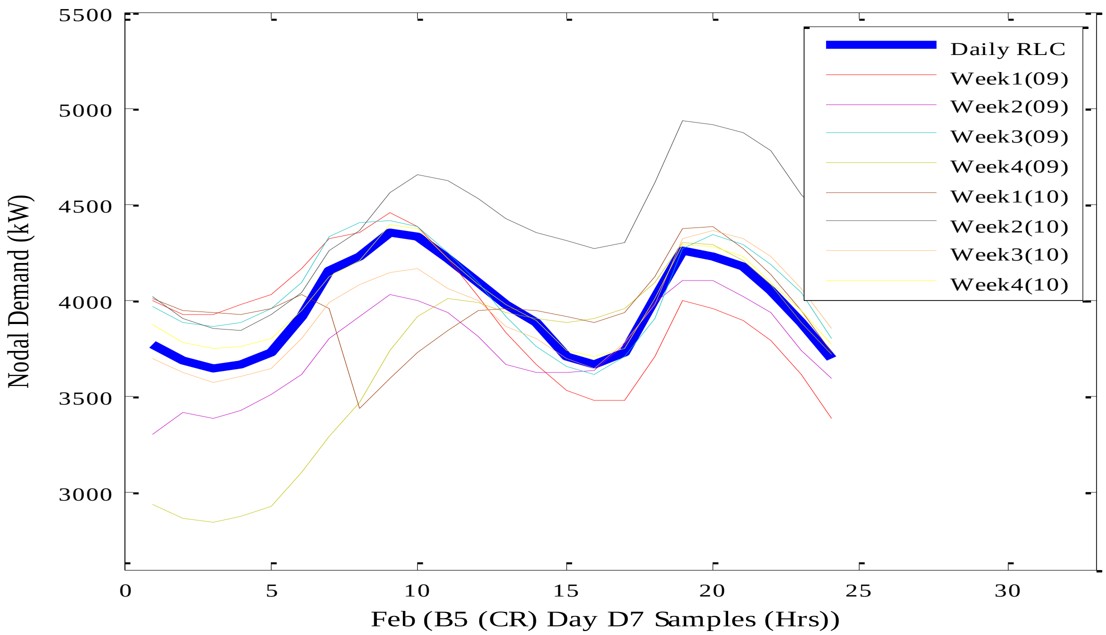

Figure 62: Node 5 (CR), February Day D7 Estimate RLC

CHAPTER SEVEN: CONCLUSION/FUTURE RESEARCH

RLCs can be done in classified time spans; examples include: winter, summer, weekends, workdays etc. amongst various classifications occurring yearly. With the aid of the MATLAB software, we've been able to answer the question posed in Chapter two. We have shown how a group of load curves can be represented using a single RLC to resolve the ambiguity of selecting a subset of the historical data that is good enough for the DLE process. Hopefully, integrating RLCs alongside synchronized real-time measurements would further aid in estimating real-time load with improved accuracy.

It's possible we can also estimate monthly RLCs. This implies we'd need a much bigger dataset. Say 6 – 10 years of historical load data. This would just be more like an estimated generalization RLC. It's also possible that we nullify our earlier assumptions and proceed with three-phase unbalanced systems to test the generalization capability of this simple method. But then again, real system properties would have to be integrated in the models. That's a possible future research.

To conclude; it's important we remember that these estimated RLCs and not load estimates themselves, but rather, possible candidates for pseudo-measurements in distribution load estimation and even distribution state estimation. And it comes with an added advantage that it sort of imitates the load curves it represents and also makes a possible case in load variation studies.

APPENDIX: LINE DATA FOR TEST SYSTEM

Table 11: Line Data for Test System

Branch Number	Sending end node	Receiving end node	R (ohm)	X (ohm)
1	1	2	1.35309	1.32349
2	2	3	1.17024	1.14464
3	3	4	0.84111	0.82271
4	4	5	1.52348	1.02760
5	2	9	2.01317	1.35790
6	9	10	1.68671	1.13770
7	2	6	2.55727	1.72490
8	6	7	1.08820	0.73400
9	6	8	1.25143	0.84410
10	3	11	1.79553	1.21110
11	11	12	2.44845	1.65150
12	12	13	2.01317	1.35790
13	4	14	2.23081	1.50470
14	4	15	1.19702	0.80740

LIST OF REFERENCES

[1] C.H. Marton, A. Elkanel, T.A. Duever, “An order-specific clustering algorithm for the determination of representative demand curves”, Elsevier Computer and Chemical Engineering, Vol. 32, pp. 1365-1372, Jun. 2007.

[2] P. Balachandra and V. Chandru, “Modelling electricity demand with representative load curves”, Elsevier Energy, Vol. 24, Iss. 3, pp. 219-230, Mar. 1999.

[3] J.S. Jang, “ANFIS: adaptive-network-based fuzzy inference systems”, *IEEE Trans. Sys. Man. Cybern.*, Vol. 23, Iss. 3, pp. 665-685, May/Jun 1993.

[4] W.H. Kersting, *Distribution System Modeling and Analysis*, CRC Press Inc, New Mexico, 2002.

[5] Jie Wan, “Nodal Load Estimation for Electric Power Distribution Systems”, *Ph.D Dissertation*, Drexel University, 2003.

[6] J. Wan, K.N. Miu, “A Zonal-Load Estimation Method for Unbalanced Radial Distribution Networks”, *IEEE Trans. on Power Delivery*, Vol. 17, No. 4, pp. 1106-1112, Oct. 2002.

[7] J. Wan, K.N. Miu, “A WLS Method for Load Estimation in Unbalanced Distribution Networks”, *IEEE Power Engineering Society Winter Meeting*, Vol. 2, pp. 969-974, Jan. 2002.

[8] V. Borozan and N. Rajakovic, “Load estimation for distribution systems with minimum information,” *Euro. Trans. On Electr. Power*, Iss. 14, pp. 331-345, Aug. 2004.

[9] V. Borozan and N. Rajakovic, “Minimum loss distribution network configuration: analyses and management,” Electricity Distribution. Part 1: Contributions. CIRED. *14th*

International Conference and Exhibition on IEE Conference, No. 438, Vol. 6, 1997, pp. 18/1 - 18/5.

[10] R. P. Broadwater, A.H. Khan, H. E. Shaalan and R. E. Lee, “Time Varying Load Analysis To Reduce Distribution Losses Through Reconfiguration,” *IEEE Trans. Power Delivery*, Vol. 8, No. 1, pp. 294-300, Jan. 1993.

[11] M. E. Baran and A. W. Kelley, “State Estimation for Real-Time Monitoring of Distribution Systems,” *IEEE Trans. Power Syst.*, Vol. 9, No. 3, pp. 1601-1609, Aug. 1994.

[12] M. E. Baran, “Load Estimation for Distribution System analysis,” Panel Summary, *Proceedings of the 2001 IEEE Power Engineering Society Summer Meeting*, Vancouver, Canada.

[13] T. Wang and M. Fan, “A novel load estimation method in distribution network”, in *Proc. Power Systems Technology*, Vol. 1.1, 1998, pp. 567-571.

[14] A. K. Ghosh, S. L. Lubkeman and R. H. Jones, “Load Modeling for Distribution Circuit State Estimation,” *IEEE Trans. Power Delivery*, Vol. 12, No. 2, pp. 999-1005, Apr. 1996.

[15] J. Nazarko and W. Zalewski, “The Fuzzy Regression Approach to Peak Load Estimation in Power Distribution Systems”, *IEEE Trans. on Power Systems*, Vol. 14, No.3, pp. 809-814, Aug. 1999.

[16] H. Kuo and Y. Hsu, “Distribution System Load Estimation And Service Restoration using A Fuzzy Approach”, *IEEE Trans. On Power Delivery*. Vol. 8, No. 4, Oct. 1993.

[17] M.R. Irving and C.N. Macqueen, “Robust Algorithm for Load Estimation in Distribution Networks”, IEE Proc.-Gener. Transm. Distri., Vol. 145, No. 5, pp. 499-504, Sept. 1998.

[18] D. M. Falcao and H. O. Henriques, “Load Estimation in Radial Distribution Systems Using Neural Networks and Fuzzy Set Techniques,” Panel Summary, *Proceedings of the 2001 IEEE Power Engineering Society Summer Meeting*, Vancouver, Canada.

[19] J. Wan, K.N. Miu, “Load Estimation in Radial Electric Power Distribution Networks using Limited Measurements”, in *Proc. IEEE International Symposium on Circuits and Systems*, Vol.2, Geneva, Switzerland, 2000, pp. 517-520.

[20] J. Wu, Y. Yu, “CBR-based Load Estimation for Distribution Networks”, in *Proc. IEEE Electrotechnical Conference*, MELECON, Malaga, Spain, May 2006, pp. 952-955.

[21] G. Grigoraş, F. Scarlatache and G. Cârţină, “Load estimation for distribution systems using clustering techniques”, *Optimization of Electrical and Electronic Equipment (OPTIM), 2012 13th International Conference*, pp. 301 – 306, May 2012.

[22] T. Konjic, V. Miranda and I. Kapetanovic, “Fuzzy Inference Systems Applied to LV Substation Load Estimation”, *IEEE. Trans. On Power Sys.*, Vol. 20, No. 2, pp. 742-749, May 2005.

[23] P.T.T. Binh, N.H. Ha, T.C. Tuan and L.D.Khoa, “Determination of Representative Load Curve based on Fuzzy K-Means” in *Proc. 4th Intl. Power Engineering and Optimization Conf (PEOCO2010)*, Malaysia, pp. 281-286, Jun. 2010.

[24] J. Hossain, A.N.M. Enamul Kabir, M. Rahman, B. Kabir and R. Islam, “Determination of Typical Load Profile of Consumers Using Fuzzy C-Means Clustering Algorithm”, *International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 1, iss. 5, pp. 169-173, Nov. 2011.

[25] D. Gerbec, S. Gasperic, I. Simon and F. Gubina, "Consumer's Load Profile Determination Based on Different Classification Methods", in *Proc. Power Engineering Society General Meeting*, Vol. 2, pp. 900-905, Jul. 2003.

[26] A. Grandjean, J. Adnot and G. Binet, "A review and an analysis of the residential electric load curve models", Elsevier Renewable and Sustainable Energy Reviews, Vol. 16, Iss. 9, pp. 6539-6565, Dec. 2012.

[27] J.A. Jardini, C.M.V. Tahan, M.R. Gouvea, S.U. Ahn and F.M. Figueiredo, "Daily Load Profiles for Residential, Commercial and Industrial Low Voltage Consumers", *IEEE Trans. On Power Delivery*, Vol. 15, No. 1, pp. 375-380, Jan. 2000.

[28] J.N. Fidalgo, "Load Curve Estimation For Distribution Systems using ANN", in *Proc. Of the 7th WSEAS Int. Conf. on CIMMACS*, pp. 197-202, 2008.

[29] D. Das, D.P. Kothari and A. Kalam, "Simple and efficient method for load flow solution of radial distribution networks", Elsevier Electric Power and Energy Systems, Vol. 17, No. 5, pp. 335-346, 1995.

[30] Historical Metered Load. www.pjm.com/markets-and-operations/ops-analysis/historical-load-data.aspx Available [Online].

[31] J.M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, Minnesota, 1992.

[32] Neural Network Toolbox™ User's Guide. www.mathworks.com Available [Online].

[33] Neural Network Toolbox™ Getting Started Guide. www.mathworks.com Available [Online].

[34] M. Sugeno, G.T. Kang, "Structure identification of fuzzy model", Elsevier Fuzzy Sets and Systems, Vol. 28, Iss. 1, pp. 15-33, Oct. 1988.

[35] A. Kandel, *Fuzzy Expert Systems*, CRC Press Inc, Boca Raton, Florida, 1992.

[36] Fuzzy Logic Toolbox™ User's Guide. www.mathworks.com Available [Online].

[37] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, Vol. 8, pp. 338-353, 1965.

[38] L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Trans. On Sys., Man & Cyber*, Vol. SMC-3, Iss. 1, pp. 28-44, Jan 1973.

[39] E.H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers", *Intl. Jour. Of Man-Machine Stud.*, Vol. 8, Iss. 6, pp. 669-678, Nov. 1976.

[40] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions", in *Proc. IFAC Symp Fuzzy Inform, Knowledge Representation and Decision Analysis*, pp. 55-60.

[41] J.-S. R. Jang, "Input Selection for ANFIS learning", *in Proc. 5th IEEE Intl. Conf. on Fuzzy Systems*, Vol. 2, pp.1493-1499, Sep. 1996.