SHORT-TERM HOURLY LOAD FORECASTING IN SOUTH AFRICA USING NEURAL NETWORKS

Masters Research Report

By

Elvis Tshiani Ilunga

Student Number: 1106539

SCHOOL OF STATISTICS AND ACTUARIAL SCIENCE

FACULTY OF SCIENCE

Supervisor

Dr Caston Sigauke

Co-supervisor

Dr Charles Chimedza

A Research Report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Science.

Johannesburg, 30 March 2018

Declaration

I declare that this Research Report is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

Abstract

Accuracy of the load forecasts is very critical in the power system industry, which is the lifeblood of the global economy to such an extent that its art-of-the-state management is the focus of the Short-Term Load Forecasting (STLF) models.

In the past few years, South Africa faced an unprecedented energy management crisis that could be addressed in advance, inter alia, by carefully forecasting the expected load demand. Moreover, inaccurate or erroneous forecasts may result in either costly over-scheduling or adventurous under-scheduling of energy that may induce heavy economic forfeits to power companies. Therefore, accurate and reliable models are critically needed.

Traditional statistical methods have been used in STLF but they have limited capacity to address nonlinearity and non-stationarity of electric loads. Also, such traditional methods cannot adapt to abrupt weather changes, thus they failed to produce reliable load forecasts in many situations.

In this research report, we built a STLF model using Artificial Neural Networks (ANNs) to address the accuracy problem in this field so as to assist energy management decisions makers to run efficiently and economically their daily operations. ANNs are a mathematical tool that imitate the biological neural network and produces very accurate outputs.

The built model is based on the Multilayer Perceptron (MLP), which is a class of feedforward ANNs using the backpropagation (BP) algorithm as its training algorithm, to produce accurate hourly load forecasts. We compared the MLP built model to a benchmark Seasonal Autoregressive Integrated Moving Average with Exogenous variables (SARIMAX) model using data from Eskom, a South African public utility. Results showed that the MLP model, with percentage error of 0.50%, in terms of the MAPE, outperformed the SARIMAX with 1.90% error performance.

Dedication

To my late father, Romain Bouyez, and especially to my dearest mother, Thérèse Kapinga, for her love, sacrifices and support, I dedicate this work.

Acknowledgements

I would like to thank Dr Caston Sigauke and Dr Charles Chimedza for their time, guidance, and input into this research report. Many thanks to Eskom for providing us with the data required for this research. Much appreciation is expressed to the School of Statistics and Actuarial Science, the University of the Witwatersrand for providing me with the needed facilities and administrative support leading to achieve this piece of work.

I extend my gratitude to all my family members, fellow students and friends for believing in me and encouraging to finish this research report.

Contents

List of Figures

List of Tables

List of Abbreviations, Acronyms and Initialisms

AI:	Artificial Intelligence
ANN:	Artificial Neural Network
APE:	Absolute Percentage Error
ARIMA:	Autoregressive Integrated Moving Average
BP:	Backpropagation
EMS:	Energy Management System
FL:	Forecasted Load
KBES	Knowledge-Based Expert Systems
LF:	Load Forecasting
MAE:	Mean Absolute Error
MAPE:	Mean Absolute Percentage Error
MIMO	Multi-Input Multi-Output
MISO	Multi-Input Single-Output
MLP:	Multilayer Perceptron
MSE:	Mean Squared Error
NN:	Neural Network
OLS	Ordinary Least Squares
SARIMAX:	Seasonal Autoregressive Integrated Moving Average with exogenous variables
SISO	Single-Input Single-Output
STLF	Short-Term Load Forecasting
VAC	Ventilation Air-Conditioning
VSTLF	Very-Short-Term Load Forecasting

CHAPTER 1

INTRODUCTION

1.1 Background

Accurate Load Forecasting (LF) is very important in the electric power industry. It is useful in power factory macroeconomic control and the power exchange plan, stated Bagnasco et al., (2014). Accurate LF can also assist to make the best decision on the optimised coordination and scheduling of generators (unit commitment problem), production and maintenance planning.

Gupta (2012) added that forecasting the electric load is a critical process in the management of utilities. One has to make sure that the energy produced meets the demand. The author also emphasised the fact that LF is massively crucial for power producers and stakeholders in the energy management system (EMS) where it is used to monitor daily operations, such as dispatch and fuel allocation. Well-timed and relevant decisions regarding LF result in a profitable and reliable network, reduce machine breakdowns and avoid blackouts.

Hedden (2015) underlined the fact that South Africa suffered from heavy power cuts caused by a supply shortage. This was an unprecedented energy crisis that damaged the South African economy to the core. The aforementioned author added that fixing this problem was not just a matter of generating more electricity. On the contrary, this required decision makers in the energy management to anticipate rather than to react. Among other means, this problem could be addressed by building models that could give accurate forecasts so as to attain a planned, efficient and smarter grid in the short-term.

In the STLF literature, various researchers (Momoh, Wang and Elfayoumy, 1997; da Silva and Moulin, 2000; Amral, King and Ozveren, 2008; Buhari and Adamu, 2012; Kumar, 2014, among others) corroborated emphatically that accuracy in the LF is very crucial in the power industry because of important various influential factors that often lead to erroneous load forecasts. A high forecasting error rate may result in either costly over-scheduling or adventurous under-scheduling of energy inducing heavy economic forfeits to power companies. Therefore, there is a strong need for accurate and reliable LF models.

The literature states that the nature of the link or the relationship between the load and its affecting factors is composite and nonlinear, making it difficult to model by means of conventional or traditional statistical methods.

Buhari and Adamu (2012) stated that conventional methods were not robust enough, noise tolerant, and they failed to give accurate forecasts when quick weather changes occurred. These traditional statistical methods may have their own advantages, but they have limited capacity to take control of nonlinear and non-stationary attributes of the hourly load series.

On the other hand the ANN methods have been successfully applied to deal with the nonlinearity in load forecasting and produced very accurate and reliable forecasts as reported in the literature (Park, El-Sharkawi and Marks II, 1991; Lee, Cha and Park, 1992; Papalexopoulos et al., 1994; Khontazad et al., 1996; Yoo and Pimmel, 1998; Senjyu, Takara, Uezato, and Funabashi, 2002).

ANN is a mathematical tool that imitates biological neural networks. ANNs are able to extract more complex relationships among input patterns by learning from training data. ANNs can learn the load patterns that would otherwise require highly complex statistical analysis methods to find. These are properties that allow the ANNs to obtain more accurate forecasts than traditional methods, and this is the reason why we used and applied them to forecast the South African power system.

To obtain better results on forecasts, we used an MLP, which is a feedforward ANNs class using the BP algorithm during its training phase, to produce accurate hourly load forecasts each time new data are available.

In this research work, we use “neural networks” to refer to ANNs or interchangeably make use of the term “network” as is done in most of the surveyed literature that also reported the use of the word ‘load’ meaning electric load. We did the same in this research report using interchangeably load or electric load.

Hong and Fan (2016) highlighted that the forecasting of the electric load and the forecasting of other utilities such as water and gas shared a lot of common properties in terms of forecasting techniques and principles. Let us then underline that “load forecasting” in this research report refers to “electric load forecasting”.

The literature and in particular Murto (1998) divided the LF methods into three groups, depending on the length of the forecasting time period, namely short-term, medium-term and long-term forecasting.

Short-term load forecasting (STLF) normally goes from one hour up to a week. Medium-term forecasting deals with the load from seven to thirty days, and long-term forecasting often predicts the electric load from one year to a few years or even up to several decades.

This research report is focused on the STLF, which is mainly used to schedule maintenance, assist in unit commitment, control the power system distribution and security, giving information to dispatchers and market operators, as pointed out by Ramezani et al. (2005).

1.2 The Electric Load

The electric load is the consumption of power energy by any piece of equipment, or anything that has a strictly positive current flow from an electric source, which is an element capable of providing electricity under right conditions. LF is a method used in the power energy management to predict the energy consumption needed by a power utility.

Murto (1998) defined the electric load of a utility as being constituted of complex consumption units. A big portion of the power energy is used by industrial companies, another portion is consumed by public services, such as traffic lights and street lighting, railway traffic, to name only a few. Private consumers use another part for daily household activities, such as cooking, lighting, ironing, etc., including appliances of agricultural irrigation.

The electric load we are talking about in this research report is the load provided by Eskom. This is, in fact, the hourly aggregated load data. In other words, this is a sequence of aggregated real numbers, the average load consumptions of hour by hour each day for eleven years.

1.2.1 The Source of the Data

The data used in this research report are hourly, aggregated load and temperature data from 2000 to 2010, provided by Eskom, a South African public utility.

1.2.2 Overview of Eskom

In 1923 the government of South Africa founded the Electricity Supply Commission (ESCOM) with regard to the Electricity Act (1922). The Afrikaans equivalence of ESCOM is Elektrisiteitsvoorsieningskommissie (EVKOM). In 1986, the fusion of ESCOM and EVKOM gave Eskom.

Eskom uses quite a number of remarkable power stations among which is the Koeberg nuclear power station (the unique nuclear power plant in Africa). The company has three main branches: Generation, Transmission, and Distribution division. In total, Eskom contributes roughly 95% of electricity in South Africa, and more than 45% in Africa. Generation's total installed capacity is about 45145 MW besides the 61 MW from Colley Wobbles, First Falls, Ncora and Second Falls hydro station managed by the Distribution Division (Eskom Holdings SOC Limited Integrated Report, 2013).

1.2.3 Factors Affecting Load Forecasting Accuracy

The LF literature points out that accuracy of the load forecasts has considerable effects on the economy since the control of the Energy Management System (EMS) may be quite sensitive to erroneous forecasting. High forecasting error rate will have a negative impact on daily EMS operations and the economy.

Hamid and Rohman (2010) claimed that factors influencing the LF accuracy depend on the specific unit of consumption. In the industrial companies, the load is generally determined by the production capacity. In this category, the load is steady most of the time. Uncertainty in the forecasting of the load of this nature comes from unexpected events, such as production equipment failure or strikes resulting in serious unpredictable turbulence in the load.

Murto (1998) argued that for the private consumers, it is quite difficult to identify precisely the factors influencing the load, since each household behaves in their own particular way. Parameters such as human psychology, social events, seasons of the year, etc. are included in the consumption decision. To reduce the number of factors influencing the load, the aggregated load of the entire utility is usually considered. This is the angle from which we looked at the load of Eskom utility in this research report.

Gross and Galiana (1987) stated that there are four main factors that influence load forecasting techniques as described below.

- Weather factors: Kothanzard et al. (1996) acknowledged that this is the most important individual factor since there is a correlation between weather and the load. Changes in the meteorological conditions affect the behaviour of consumers in the sense that weather-sensitive loads due to Heating Ventilation and Air-Conditioning (HVAC) tend to have a great impact on the power system. In the regions where there is a huge difference between summer and winter weather, load patterns will exhibit an irregular curve. Regarding forecasted weather variables, the most important ones in STLF are the temperature, humidity, and wind speed.
- Time factors: Gupta (2012) pointed out that from the forecasting angle, time factors are very essential. These include various seasonal effects and cyclical behaviours like daily and weekly oscillations, as well as the occurrence of public holidays. There is a difference between weekdays and weekend loads (the weekend or holiday load curve is lower than the weekday curve). The load variation with time reflects people's lives, like working time, leisure time and sleeping time.
- Random factors: all other factors causing disturbances, such as strike, inclement weather, or even popular TV-programs, are classified as random factors. They add uncertainty in the forecasts that cannot be explained by the previous three factors and making prediction very difficult.

1.2.4 Overview of Load Forecasting Methods

According to Alfares and Nazeeruddin (2002) the LF methods and models can be classified into nine categories as follows:

1. Multiple linear regression,
2. Exponential smoothing,
3. Iteratively reweighted least-squares,
4. Stochastic time series,
5. Autoregressive moving average model with exogenous inputs (ARIMAX),
6. Models based on genetic algorithm,
7. Fuzzy logic,
8. Neural networks and
9. Expert systems.

Some of the most popular methods in LF such as multiple linear regression, stochastic time series, knowledge-based expert system, and fuzzy logic will be explored further. The NNs technique is described in detail in chapter 3. The first five categories are considered as statistical methods and the remainder categories are data mining or machine learning techniques, a particular approach of Artificial Intelligent (AI).

1.3 Aims and Objectives of the Study

1.3.1 Aims

The principal aim of this research report is to construct a reliable NN-based model that produces hourly load forecasts up to 24 hours ahead.

After constructing such an LF NN-based model, the subsequent aims are to:

- ❖ Provide decision makers with necessary information regarding the load demand to help them run their daily operations more efficiently and economically,
- ❖ Solve the unit commitment problem and minimise the operating costs,
- ❖ Prevent overloading and reduce occurrence of equipment failures,
- ❖ Schedule spinning reserve (back-up energy production) allocation properly and
- ❖ Schedule routine maintenance.

1.3.2 Objectives

This research report should constitute the basis for an MLP model application to predict accurately the electric load in a real-time environment. The specific objectives of this study are mainly to accredit the built MLP model with the following important properties:

- ➢ Accuracy: model should be very accurate as required in the literature and compared favourably to a benchmark model (SARIMAX),
- ➢ Robustness and adaptability: the model should adapt to quick changes in the load consumptions (due to whimsical weather, for example),
- ➢ Reliability: unpredictable events should not result in highly erroneous forecasts, and
- ➢ Up to date: the model should be able to forecast with new available data.

1.4 Organisation of the Research Report

Chapter 1 gives the background of the LF and enumerates its major techniques. This chapter also states the aims and objectives of this research report, describes the data and gives a brief overview of Eskom;

Chapter 2 outlines the common methods and surveys the literature on the STLF;

Chapter 3 introduces and describes the neural networks method;

Chapter 4 is about the materials and methodology used to build the proposed MLP model;

Chapter 5 analyses the load profile, presents and discusses the load forecasting results;

Chapter 6 is dedicated to conclusions and recommendations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The STLF literature is focused on different aspects of the problem as a whole, with some literature covering statistical approaches, and some looking at ANNs as a Data Mining or Machine Learning technique. A number of techniques developed for LF were surveyed and the most common are presented below.

2.2 Load Forecasting Techniques

There are several techniques developed for LF in the literature, but in this report, we only looked at a few of them.

2.2.1 Multiple Linear Regression (MLR)

Papalexopoulos and Hesterberg (1990) stated that regression is one of the most widely used of the statistical techniques, which assumes that there is a linear dependence between the load components and some explanatory variables. This approach uses weather and non-weather variables, such as temperature, humidity, day types, and customer class as predictors of the load at a particular time. The model can be written as follows.

$$z(t) = a_0 + a_1 x_1(t) + \cdots + a_n x_n(t) + a(t), \tag{2.1}$$

where $z(t)$ is the electric load, $a(t)$ is a white noise component with zero mean and constant variance, $x_i(t)$ are the explanatory variables, and a_0 and a_i are regression coefficients, with $i = 1, \ldots, n$.

In general, to choose the most typical explanatory variables of this model, we use the correlation analysis. Estimation of the regression parameters is carried out by means of least-squares technique. MLR can be easily implemented and updated, but its typical sensitivity serial correlation of weather variables disturbances can be a major problem as highlighted by Murto (1998).

2.2.2 Stochastic Time Series

Cabrera, Guiterrez-Alcaraz and Gil (2013) claimed that the stochastic time series approach is one of the very popular LF models. The method is based on the assumption that the data is structured in a way that exhibits autocorrelation, trend and/or seasonal patterns. Historical data are used to forecast the future. The literature of this type of techniques, such as ARMA (Autoregressive Moving Average), ARIMA (Autoregressive Integrated Moving Average), the Box and Jenkins method and their seasonal versions abounds in the LF field. Janacek and Swift (1993) discussed most of these classical time series methods in detail. The philosophy of these methods lies in the fact that the load time series is first transformed into a stationary load by a differencing operator and/or a Box-Cox transformation. Then the newly obtained stationary series is modelled as the output of a linear filtered model with a white noise input, claimed Murto (1998). The ARIMA model can be written as follows.

$$\emptyset(L)\nabla^d z_t = \theta(L)a_t, \tag{2.2}$$

where z_t is the time series to model and a_t is the white noise process, L is the lag operator or backward shift and ∇ is the difference operator such that $\nabla = 1 - L$, and $t = 1, \ldots, N$. The Autoregressive (AR) process is given by the following expression.

$$\emptyset(L) = 1 - \emptyset_1 L - \cdots - \emptyset_p L^p, \tag{2.3}$$

and the Moving Average process can be expressed as follows.

$$\theta(L) = 1 - \theta_1 L - \cdots - \theta_q L^q, \tag{2.4}$$

where $\theta \ and \ \emptyset$ are constant parameters. The two processes above can be combined to form an Autoregressive Moving Average (ARMA) process. But researchers, in general, and in particular Paretkar et al. (2010) agreed that this ARMA model is not convenient to describe properly the load time series, which includes seasonal patterns due to hourly, weekly and monthly behaviours. To deal with the seasonal patterns in the load time series we resorted to a Seasonal Autoregressive Moving Average (SARMA) process, which, most of the time, causes the series to be nonstationary. Therefore, we modified, beforehand, the ARMA model by a differencing process to obtain an Autoregressive Integrated Moving Average (ARIMA) model in equation (2.2). To cater for the seasonal patterns, the new model is known as the Seasonal Autoregressive Integrated Moving Average (SARIMA) model as developed by Box and Jenkins in 1970s. The SARIMA model can be expressed as follows.

$$\emptyset(L)\Phi_s L^s \nabla^d \nabla_D^s z_t = \theta(L)\theta_s(L^s)z_t, \tag{2.5}$$

where $\nabla_D^s = (1 - L^s)^D$, d is the order of differencing, s is the seasonal period variation (per week, month, year, etc.), and D is the order of seasonal differencing. When this SARIMA model is applied to load forecasting with data including weather variables such as the temperature, which is seen as an external input variable, the model is called SARIMAX.

Besides the given equation in (2.2), we explored the ARIMAX model as a transfer function model which assumes two time series denoted Y_t and X_t, to be both stationary. Then the transfer function model (TFM) is given by

$$Y_t = C + v(B)X_t + N_t\ , \tag{2.6}$$

where: Y_t is the output series (dependent variable),

X_t is the input series (independent variable),

C is a constant term, N_t is the stochastic disturbance,

$v(B)X_t$ is the transfer function (or impulse response function), which allows *X* to influence *Y* via a distributed lag.

B is the backshift operator.

When X_t and N_t are assumed to follow ARMA model, equation (2.6) is known as the ARMAX model.

The transfer function can be written as the rational polynomial distributed lag model of finite order as the ratio of a low order polynomial in *B*:

$$v(B)X_t = \frac{\omega_h(B)B^b}{\delta_r(B)}X_t\ \ , \tag{2.7}$$

where, $\omega_h(B) = \omega_0 + \omega_1 B + \cdots + \omega_h B^h$; $\delta_r(B) = 1 - \delta_1 B - \cdots - \delta_r B^r$. The function $\omega_h(B)$ and $\delta_r(B)$, and parameter b are then determined from the cross-correlation between Y_t and X_t.

The weakness of this class of SARIMA models lies in failing to adapt to some quick changes of the load behaviour during a year. Since ARIMA models forecast is a function of all the previous loads, then it would be very difficult for them to adapt quickly to new conditions that occurred in the interim, even if the models are updated regularly as pointed out by Cabrera et al. (2013).

Yang et al. (2013) and Mohamed et al. (2011) analysed the SARIMA models more in depth with additional scope such as mathematical relationships and interpretation.

The application of the SARIMA model used in the auto.arima function was established in four main steps as pointed out by Yang et al. (2013) as follows.

1) Identification structure of the SARIMA (p,d,q) (P,D,Q): the autocorrelation function (ACF) and the partial autocorrelation function (PACF) are used to build the rough function. At this stage, different models are built and appropriate models are chosen. This identification step is principally to determine the adequate AR, MA, or ARMA processes and their respective orders.
2) Estimation of parameters: this phase consists of determining the unknown parameters through ordinary least squares (OLS) or sometimes via other means such as nonlinear estimation methods. The AR and MA processes parameters obtained through ARIMA model should determine whether these processes are stationary and invertible or not, respectively.
3) Goodness-of-fit tests applied on the estimated residual: in this phase, estimated ARIMA models are analysed to determine whether they harmonized or not by diagnostic checking.
4) Data driven forecasting: determining the future outcomes of the estimated ARIMA models that the derived AR and MA observe the unit circle and normality assumptions.

2.2.3 Expert Systems

Expert Systems or Knowledge-Based Expert Systems (KBES) as pointed out by Taylor (2013) are recent heuristic techniques resulting from progress in the artificial intelligence (AI) field. The basic idea consists of trying to emulate the reasoning of an expert operator in power industry, but keeping track of reducing the analogical thinking supporting the intuitive forecasting. This imitation process can then be converted into formal logical steps that can be automated and form an expert system. The system is basically constructed based on the knowledge of the expert, the load, and the relevant weather variables as stated by the aforementioned author who, furthermore, added that KBES is a computer program that is not characterised as being based on any algorithm. In the same line of thought, Moghram and

Rahman (1989) defined the KBES as a program that “can reason, explain, and have its knowledge basis expanded as new information becomes available”. Gupta (2012) contributed to the establishment of expert systems in underlying the fact that once the load and the factors affecting it are known and extracted, a parameter-based rule can be implemented. This rule is of the form “IF THEN”, plus some mathematical expressions. This rule can be used on a daily basis to generate the forecasts.

Expert systems and their heuristic approach to find solutions make them promising; however, the knowledge of the expert might not always be consistent and the reliability of such ideas may be questionable.

2.2.4 Fuzzy Logic

Rouse (2006) defined fuzzy logic as a computing technique based on “degrees of truth” instead of the well-known Boolean logic “true or false” (1 or 0). Fuzzy logic is rather a generalisation of this Boolean logic on which modern computers are based. Ranaweera, Hubele and Karady (1996) described fuzzy logic models as a function that links a set of input variables to a set of output variables; these input variable values do not need to be numerical. They just need to be transcribed in a natural language. For example, a weather parameter such as the temperature may take on the “fuzzy” instances such as “low”, “medium” and “high”. The literature adds that very often fuzzy logic models incorporate a mapping of input and output values via a simple “IF THEN” logic statement. “IF the temperature is very low, THEN the load demand will be very high”, is an example of this logic statement given in the Ranaweera et al. (1996) paper. The authors further reiterated that this is a type of mapping and logic that allows a combination of the expert knowledge with fuzzy logic models. In many instances when precise outputs are needed, such as point estimates for forecast values, a reverse mapping called “defuzzification” process can be undertaken to produce those desirable outputs. Advantages of this method over traditional ones can be found in Gupta (2012). The drawbacks of the fuzzy logic models are that they are time consuming and lack of guarantees to obtain optimal fuzzy rules and membership functions since this process is based on trial and error.

2.3 Neural Networks Literature Survey on STLF

The literature on STLF, especially the one based on NNs, is very extensive proving that NN power systems models have not become a ‘passing fad’ as apprehended Chatfield (1993).

In the early beginning of the ascension of NN, Park, El-Sharkawi and Marks II (1991) built and proposed a merged time series and NN BP model to predict future load based on the Puget Sound Power and Light (PSPL) company data. The built model took into account the temperature and excluded weekends and was trained to recognise particular characteristics of the load, such as hourly load, peak and total load of the day. Obtained results compared favourably with less than 3% of error over the prediction made by PSPL using the same data. However, the authors posited that additional weather components should yield even better results.

Lee, Cha and Park (1992) constructed an NN model based on the backpropagation algorithm to forecast 24 hours ahead without including the temperature. The week was divided into two parts regarding load patterns: weekdays and weekend. In this paper, two different techniques of using NN were presented: the first technique was a static approach that forecasted the 24-hour-load at a time, whereas the second technique was a dynamic approach forecasting the one-day-load hour by hour using the previous hour forecast. The performance of the model was tested through the two techniques using an illustrative example based on the Korea Electric Power Company data. Various structures of NN with constant learning rate and momentum parameters were tested. Results showed that the dynamic technique performed better than the static one with an error less than 2%, which seems to be good according to the literature. The authors concluded that including the weather variables, some additional parameters, such as the sigmoid function would increase the forecasting accuracy.

Peng, Hubele, and Karady (1992) proposed a modified NN approach by implementing a novel strategy to select weather variables and relevant load patterns using the smallest distance measurement during the training phase, so as to improve the network accuracy. The implemented NN structure was very flexible, it could adjust to an hourly, peak load or multiple days ahead forecasting. The authors used two-year utility normalised data to test the proposed search strategy and algorithm; the obtained results were reported in a new measure of performance using a cumulative errors distribution plot in addition to reporting the Absolute Percentage Error (APE), and summary statistics. These results were satisfactory as less than 2.5% of error, compared to those reported previously in the literature.

Papalexopoulos, Hao and Peng (1994) developed and implemented an NN-based LF model with particular attention given to accurately forecasting unusual days like public holidays, very hot or cold successive days and other weather conditions that perturb the electric load common features. Energy control centre, Pacific Gas & Electric Company (PG&E) data from 1986 to 1990 were used to train and test the model. The performance of the model was compared to an existing regression model using the same data. The NN-based model produced more accurate results in terms of forecast errors, and was robust, adaptive to weather changing conditions.

According to Hong and Fan (2016), one of the most successful implementations of NN models for STLF was developed by Khotanzad et al. (1997) and sponsored by the Electric Power Research Institute (EPRI). The authors nevertheless admitted that a couple of conventional techniques were used previously with varying degrees of satisfaction, but not as accurate as would be desired. Besides, most of these models could not be used elsewhere, but at the built site. This paper investigated several types of NN architecture such as recurrent NN and radial basis NN, and came to a conclusion that there was no major advantage of these architectures over the MLP in terms of the load forecasting problem. This NN Short-Term Load Forecaster (ANNSTLF) constructed by the aforementioned authors, was subjected to different comparative studies using various methods as well as other NN-based models. The accuracy of the load forecasts was evaluated and expressed in terms of the MAPE. The ANNSTLF yielded very good results and induced its acceptance across Canada and USA.

Unlike all the previous authors, Yoo and Pimmel (1998) used NN with a self-supervised adaptive algorithm to build an STLF model to forecast a one-hour-ahead and a one-day-ahead power load. The authors defined the self-supervised adaptive algorithm to be a self-organising or topological neighbourhood learning algorithm providing a topological ordering by updating the weights and the nearest neighbour neurons. The paper further showed how the built algorithm could obtain correlation patterns between weather variables, such as temperature and load data by means of one-hour delay function. Next, the authors pre-processed the data, structured the architecture and implemented the model, which was tested on a power plant’s actual data in 1993. The test results showed that day-ahead model with 1.92% errors average performed better than the 3% errors average reported in the literature using the BP algorithm.

Hippert, Pedreira, and Souza (2001) presented the state-of-the-art of the NN method applied to STLF in a review that massively surveyed about 40 papers on the application of the NN to LF published in globally well-known journals in electrical engineering, for almost ten years (1991-1999). The authors explained why there was a strong hesitation among experts to admit the success of ANNs compared to standard load forecasting methods. They also underlined two major shortcomings that probably led to that scepticism. Firstly, many of the ANN proposed architectures seemed too large for the data samples to model resulting in overfitting that may have produced inaccurate out-of-sample results. Secondly, in most of the published works in this field, the authors noticed that models were not consistently evaluated, to an extent that test results were not always adequately presented. Furthermore, the authors criticised the fact that models were not even compared to the benchmark standard models. Finally, the review suggested four different stages: data pre-treatment, design of the network, its realisation and validation to deal with STLF designing issues.

Senjyu, Takara, Uezato, and Funabashi (2002) proposed a one-hour-ahead LF using NN to minimise its learning time and size structure. Indeed, the authors strongly criticised the use of similar day's data to learn the shape of the curve of similarity as this is too complex and not suited for the NN approach. The paper justified the use of its approach on the fact that in most of the literature, they used 24-hour-ahead LF and forecasted temperature information; but if the weather changed abruptly on the forecasting target day, load energy forecast error would dramatically increase. In that case, the paper suggested to retrain the NN to allow the re-learning of the relationship between temperature and load. An illustration of the effectiveness of the proposed approach was carried out through Okinawa Electric Power (Japan) case study.

Taylor and Buizza (2002) investigated the use of weather ensemble predictions to improve accuracy in the NN load forecasts. Indeed, the authors used 51 ensemble members for the weather variables (temperature, wind speed and cloud coverage) in different scenarios to build the load density function and obtained its mean that they used to forecast the load. Next, the weather ensembles were used to estimate the load forecasts error variance through a naïve method, an exponential smoothing method, and a rescaled variance of NN load scenarios to take control of the uncertainty generated by the residual error and parameters estimation error. Based on the same weather ensembles reasoning, load prediction intervals were constructed. Finally, the paper compared the proposed approach to a traditional method, the Box and Jenkins procedure, for lead times from 1 to 10 days ahead, in terms of the MAPE, and the results were satisfactory allowing the authors to conclude that using weather ensemble predictions in NN load forecasting were promising.

Mandal et al. (2006) proposed an NN-based several-hour-ahead load forecasting model applying similar days approach and used the temperature as a weather variable to detect the trend of similarity. In fact, the authors, in this paper, used the Euclidean norm combined with weighted factors, obtained via the least squares method, to determine the similarity between the forecasting target day and past days, in a specific season. These similar days load were averaged to enhance the forecasting precision. The proposed NN structure could easily deal with non-linearity part of the load and special days and weekend problems. Hourly load and temperature data from 1999 to 2000 from the Okinawa Electric Power Company, in Japan, were used to train and test the network. The authors carried out six case study simulations (one to six-hour-ahead) to assess the predictive capacity of the proposed method. Results gave 0.98% for one-hour-ahead in terms of MAPE. Since there was a growing trend with increasing hour ahead, this figure raised up to 2.43% of MAPE for over six-hour-ahead forecasting. In the light of this behaviour (increasing average error with increasing hour ahead), it turns out that the proposed model could perform better only for few-hour-ahead (less than six).

Amral et al. (2008) developed and evaluated a 24-hours-ahead model. In fact, they implemented three NN different models: the first model was of 24 output nodes forecasting 24 hours at once, the second model forecasted the maximum and minimum power, and lastly a model with 24 individual NNs for 24 hours of the day working together to forecast the energy demand. The three MLP based models were examined and compared to each other, using the South Sulawesi (Indonesia) hourly load and temperature data for 2005-2006. The MAPE was used to evaluate the models. The last model performed much better than the two others.

Osman, Awad, and Mahmoud (2009) proposed a one-hour-ahead NN-based model in STLF to complement NN-based models with a 24-hour-ahead shortcoming in case of abrupt changes in the weather that could lead to erroneous forecasts. The authors underlined the fact that the NN structure is strongly system dependent and thus, thoroughly studied the characteristics of the Egyptian Unified System (EUS) power load profile. The authors used the correlation analysis to select the input variables. They also used a “minimum distance” between inputs and target data to select appropriate training vectors and eliminated the special days such as public holidays and weekends. Next, the authors built four models for each season to test the possibility of application of the proposed approaches based on the national grid in Egypt, using actual 2004 EUS data. In terms of the MAPE, the proposed model realised better results with 2.2% compared to other some complex regression benchmark models.

Qingle and Min (2010) constructed a Very Short-Term (a couple of dozen minutes) Load Forecasting (VSTLF) model that combined rough set theory, a computer science approximation method based on set theory by Pawlak (1982), and NN to improve forecasting accuracy. Basically, in this paper, an MLP model was used to perform the load forecasting, and the input variables consisted of the load of the target time, the load of the previous time, and the load of the difference between the target and the previous time. The authors used 10 neurons in the first hidden layer, 5 neurons in the second hidden layer, and one node at the output layer. The outputs of the MLP model were adjusted by the rough set theory to yield even more precise load forecasts.

In most of the latest work we surveyed, researchers revisited the merits and advantages of NN method in highlighting its ability to capture the non-linear relationships between the load and the weather variables, and over and above its ability to learn from past load patterns and adapt to new data during the training and validation phases.

An ANN forecaster based on the Matlab-R2008b Levenberg-Marquardt BP algorithm was built by Buhari and Adamu (2012) using load data from the Kano Power Utility (Nigeria). Hernandez et al. (2013) applied an STLF model based on NN to microgrids scenarios and showed that on small geographical areas, NN models could yield even very accurate results. The philosophy developed in this paper converges with many ideas or suggestions made by Hedden (2015) on the World Economic Forum website as some of the solutions to the South African energy crisis.

In the work proposed by Reddy and Momoh (2014), a new way to formulate load forecasting models using NN-BP based on different mathematical models for STLF was born. In this paper, the authors emphasised the fact that access to historical load data on the utility makes ANN implementation extremely convenient to the LF field. A marriage of several input parameters, such as load inertia, autocorrelation, number of time lags in the data, and short-term trends were used to construct these mathematical models so as to produce reasonably accurate results. The structure of the BP algorithm was constantly updated so as to find an optimal model able to meet the demands of large utilities for the hourly load forecasting one-day-ahead.

Al-Subhi and Ahmad (2015) used the NN technique applied to STLF on an industrial residential area, in Saudi Arabia, by proposing two different models, the next hour and the next day load forecasting models. The proposed next day LF model was just an extension of the next hour iterated 24 times. These models were based on two-layer and three-layer feedforward networks, respectively. The authors insisted on the importance of the factors affecting the load demand, such as weather conditions, religious behaviours, and official calendar on which basis they built their models. Three year-data (2009-2011) were used to train and forecast the models and the MAPE was calculated to evaluate the performance of the models. The results were very accurate with the errors ranging between 0.35% and 0.49% for the next hour model and 1.48% to 2.58% for the next day model. The authors, finally, compared their proposed models to a published work by Abdel-Aal (2004) to get a good opinion on the effectiveness of the proposed models, and the result was satisfactory.

2.4 Summary

In the first part of this chapter, we outlined the most commonly used NN methods, and surveyed the STLF literature in the second part.

In most of the papers on the STLF literature that we surveyed in this chapter, they used the feedforward MLP for its backpropagation algorithm good performance. Authors could be divided into two groups, namely, those who built their models based on forecasting the whole day or 24 hours at a time, and those whose models were constructed based on the idea of forecasting the load curve hourly recursively up to 24 hours. But in reporting the effectiveness of their models, most of the authors used only one error metric, the MAPE. Some of them did not even compare their models to well established benchmark models, while Hippert et al. (2001) underlined that this fact led to the scepticism reigning among experts as to the advantage of using NN in load forecasting. This is the reason why we undertook to build an hourly NN model that minimise errors hour by hour so as to get a much more accurate model and compared it to a SARIMAX model using the MAPE, the MSE, the MAE, the APE and the Daily Peak Error to assert its validity.

CHAPTER 3

NEURAL NETWORKS

3.1 Introduction

Historically, the neurologist Warren McCulloch and the logician Walter Pits are considered as fathers of the NN method as they designed and produced the first neural network in 1943. However, they could not go further as the technology available at that time was not able to facilitate practical research work. Wasserman (1989) underlined that researchers lost all interest in theory and applications of NN in the 1970's. Ten years later, in the early 1980's, NN started to grow massively.

Zhang, Patuwo and Hu (1998) defined NN as a biologically inspired mathematical means of computation. NNs are structured in a way that their components perform similarly to the most basic functions of the biological neuron in a human brain. NNs have many characteristics of the brain; they can be taught and can learn from previous experience, generalise to new ones.

3.2 Why use Neural Networks?

In a conservative way, Buhari and Adamu (2012) highlighted what is essentially emphasised in the literature by many authors. They claimed that statistical and expert system techniques failed to solve the nonlinearity problems related to the factors affecting the load demand, such as the weather variables, human and industrial activities.

In a more conciliatory way, Kumar (2014) admitted that some of the conventional approaches to solve the above mentioned problems yielded satisfactory results in some well constrained domains but still none of them was flexible enough as the NN techniques.

Kumar (2009) pointed out that NNs have an extraordinary capacity to obtain and make sense of very complex or imprecise data. He added that NNs are useful in detecting trends and abstracting patterns that are too complicated to be identified by either humans or other computer means. The author referred to a trained neural network as an “expert” who has been given information to analyse. In some new cases of interest, this expert should be able to provide reasonable projections and answer the “what if” questions.

In the literature, NNs are said to be very good at performing human-like tasks in the fields such as pattern recognition, speech processing, image recognition, machine vision, classification, system identification, control system, etc.

The so-called universal approximation theorem replicated by Kalogirou (2001), Zhang, Patuwo and Hu (1998) states that: “ANNs are able to numerically approximate any continuous function to the desired accuracy”. The authors added that NNs can be seen as nonlinear and nonparametric multivariate methods. For NN models it is not required to formulate any tentative model and then estimate its parameters. Provided with a set of input vectors, an NN can be taught, can learn and map the relationship between inputs and outputs of a network. NNs are model free estimators and data-driven. Rewagad and Soanawane (1998) stated that NNs are mostly used because of the following properties seen as big advantages compared to other techniques:

- Auto-coordination: NNs are able to generate their own structure of the information extracted during the training phase.
- Robustness: NNs have the ability to recover from a major damage of their components and be still usable
- Parallelism: different real time operations can be performed at the same time in NNs

3.3 Neural Networks and Statistics

Sandoval (2002) claimed that statistics and NNs do not compete but complement each other. A table of similarities is given in Table 3.1 below.

Table 3.1 Similarity between NNs and statistics

Neural Networks	**Statistics**
Learning	Model estimation
Supervised Learning	Nonlinear Regression
Weights	Parameters
Inputs	Independent variables
Outputs	Dependent variables
Unsupervised Learning	Cluster Analysis

3.4 Neural Networks Architecture

Haykin (1999) defined a neuron as the fundamental building piece of any NN architecture. He added that a neuron is taken as a data treatment unit that is critical to the NN operation. Inputs come from some other different neurons, in some cases from an external source. According to the author the three fundamental components of a neuron-based model are:

1. A collection of weights, each of which is described by its own ability.
2. An adder for adding information, balanced by the individual weights of the unit.
3. An activation function to restrict the size of the output of a neuron. This activation function is also called a transfer function. There are many distinctive sorts of transfer functions, but the most widely used are the logistic sigmoid and tangent hyperbolic functions. Haykin (1999) represented both functions as given below in Figures 3.1 (a) and 3.1 (b).

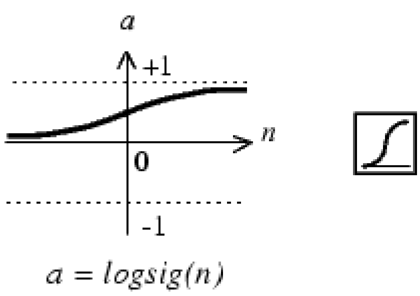

Figure 3.1 (a) Log-Sigmoid Transfer Function

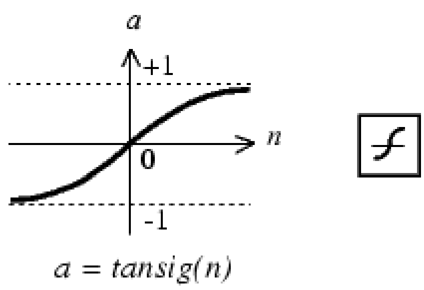

Figure 3.1(b) Tan-Sigmoid Transfer Function

The author further stated that, in general, NN architecture can be categorised in different essential classes given below.

i. Single Layer Feed forward Network

In a basic form of an NN with different layers, there is an input layer of source nodes that casts forward directly onto computational nodes. This is referred to as a “single layer” network because only the computational layer counts as can be seen in Figure 3.2 below:

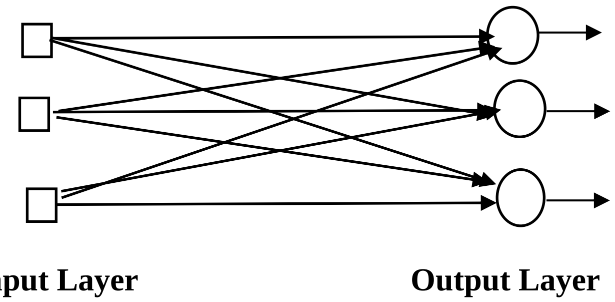

Figure 3.2 Single layer network

ii. Multilayer Feedforward Networks

The multilayer feedforward NN has one or more hidden layers. The hidden neurons facilitate the NN to learn complex tasks. Neurons in a layer are projected forward onto the contiguous layer, but not in the opposite direction. The source nodes provide information to the neurons in the next layer; the outputs of this layer are the inputs to the adjacent layer, and so on for the remaining network. The final signal from the output layer constitutes the general response of the network.

A drawn example of a multilayer feedforward network is given in Figure 3.3 below.

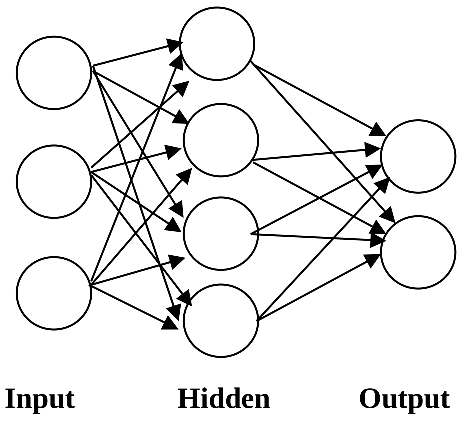

Figure 3.3 Multilayer Feedforward network

iii. Hopfield Network

The Hopfield network is a model comprising a number of neurons and a related set of unit-time delays, constituting an ensemble of multiple-loop self-input. There are as many neurons as the signal loops. The individual neuron signal is essentially sustained back to each of the alternate units in the network through a unit-time delay component. There is no self-input in the model, in a sense.

iv. Recurrent Networks

There is at least one feedback loop in a Recurrent Neural Network (RNN). This model may comprise a unique layer of neurons, each of which sustaining its signal back to the remaining units in the network. In some cases, the network does not have any self-feedback loops. When the signal of a neuron is sustained back to itself, this is what is called self-feedback. There are two categories of RNN; one with and another without any hidden neuron.

3.4.1 Neural Networks Topology

There are different ways to construct an NN, but deciding on its structure and the number of neurons in its layer(s), specifically the hidden one(s), is the most important aspect, thus in the construction of an NN we need to determine:

a) Number of neurons in the output layer

The question of how many neurons to use in the final layer dependents on one case to another; one should first think of the intended use of the NN. If the NN is used in classification, then one output neuron for each class of input items is sufficient. In some other cases, like the error reduction on a signal, the number of input and output neurons is exactly the same.

b) Number of hidden layers and hidden neurons

At the current state of the science, there is not any exclusive rule to determine the exact number of hidden layers. However, an NN with two hidden layers is able to describe functions with any shape.

These hidden layers are very important, they have an extremely strong influence on the final output. Deciding on how many hidden neurons should be used so that to obtain the best results is very crucial but system dependent. Charytoniuk and Chen (2000) underlined the point that not enough neurons in the hidden layer will produce an underfitting network, and too many neurons can have consequences such as very long training time or an overfitting of the network. Besides, the network may perform poorly on unforeseen input patterns. In most cases, the selection is performed by trial and error.

3.5 Learning Processes

Learning in NNs means that weights are able to adjust their values according to the modifications undergoing in the network. Of all the intriguing characteristics of NNs, their ability to learn is the most attractive. The NNs' self-organisation also plays an important role in their good reputation. Given a set of inputs, they can self-adjust and yield accordant outputs.

A lot of learning algorithms have been created; however, every learning algorithm involves the learning process as described above. Neurons that constitute the network are interconnected through their synaptic weights, allowing communication between themselves as the data are proceeded. Weights, in the network, are not evenly assigned to neuron connections. If there is not any communication between two neurons, then the weight is zero.

Training is the phase where these weights are assigned to neuron connections. Most of the training algorithms initialise weight matrices with random small numbers, generally between [-1, 1]. Next, the weights are adjusted based on how well the network performs. The structure of the NN is directly related to the learning algorithm used to train it. In a broad sense, there are two forms of the learning process, supervised learning (learning with a teacher) and unsupervised learning (learning without a teacher).

3.5.1 Supervised Learning

The supervised learning or learning with a teacher necessitates the input dataset together with the desired output or target values for its training. During the training stage, the outputs from the NN are compared to the target and the difference or error is reduced by using a training algorithm. As the process of learning carries on under supervision, the NN is taken through a number of iterations, or epochs, until its outputs match the target or reach some reasonable small error rate. Supervised learning is the learning technique used in this project because we have been provided with the desired outputs (targets) in the data, so that we can ready the pairs consisting of input object and target value needed for the training phase.

3.5.2 Unsupervised Learning

In unsupervised learning or learning without a teacher or simply self-organized learning, there is no target provided. The quality metrics for the task that the network has to learn is provided, and the neuron weights are optimized according to that measure. Unsupervised learning is mostly used to train NNs in classification problems. A set of input patterns is presented to the input nodes, then they are processed in the hidden layer generating a firing neuron on the output layer. This firing neuron gives a classification of the input patterns and indicates to which class they are to be allocated.

3.5.3 Learning Rules

Haykin (1999) defined a learning rule as a mathematical formal system or just a technique that iteratively enhances the NN performance over the training phase. Numerous learning rules are commonly applied, but many of them are just an approximate modifications of the best known one, the Hebb's Rule. Some of the major rules are:

a) Hebb's Rule

This rule is probably one of the well-known learning rules developed by Donald Hebb in 1949 and mostly used in unsupervised learning, claimed Heaton (2008). Its basic principle states that if two unit neurons are connected and both have similar activations, then the weight between them should be increased. This is sometimes summarised by "Neurons that fire together, wire together" (Heaton, 2008). Symbolically the rule is given below.

$$w_{ij} = \frac{1}{n}\sum_{k=1}^{n} x_i^k y_j^k, \tag{3.1}$$

where w_{ij} is the weight from neuron *j* to neuron *i*, *n* is the size of the sample of training input data and x_i^k the k^{th} input for neuron *i*, y_j^k the k^{th} input for neuron *j*.

b) The Delta Rule

Heaton (2008) stated that the Delta rule is just a transformation of the Hebb's Rule, also referred to as the Least Mean Square (LMS) learning rule. The author described this rule as built on a basic idea of regularly updating the synaptic weights of the input patterns so as to minimise the error (called delta), which is the difference between the target value and the output signal of the network. The resultant delta is back propagated into prior layers one by one until the first layer is reached. The Delta rule can be written as follows.

$$\Delta w_{ji} = \alpha(t_j - y_j)x_i \, , \tag{3.2}$$

where α is the learning rate, t_j is the target output, y_j is the j^{th} output and x_i the i^{th} input.

c) The Gradient Descent Rule

In this rule, the derivative of the activation function is used to update the delta, as described in the previous section, before using it on the connection weights.

Algebraically we have: $$\Delta w_{ij} = \eta \frac{\partial E}{w_{ij}} \, , \tag{3.3}$$

where η is the learning rate and $\frac{\partial E}{w_{ij}}$ is the derivative of the error gradient w.r.t. the weight w_{ij} from neuron j to neuron i.

3.5.4 Learning Rates and Momentum

The learning rate is a parameter that specifies the speed at which the NN will learn. Whereas the momentum keeps track and quantifies the effect of previous training iteration on the current one.

The learning rate depends on many factors affecting the network. Choosing an appropriate learning rate is not an easy task. A very small rate implies learning at slower pace and the smoother will be the curve trajectory but the process will take a long time to accomplish and produce a suitable trained NN. With a big learning rate to speed up the learning pace, the learning algorithm can easily exceed the limit in updating the weights and the network will swing back and forth. Usually the learning rate is a positive constant between zero and one, and the momentum is usually a positive value close to one.

3.6. Training Algorithms

During the training phase, neuron weights are updated to reach desirable outputs as the error is minimised. The function that seeks for weights that will reduce the error rate can be the gradient descent, among others, and the function that evaluates the NN error rate is the learning rule or training algorithm. Many common training algorithms are used, such as the genetic algorithm, the simulated annealing, the evolutionary methods, gene expression programming, etc. However, one of the most popular algorithms used in the literature, and the one we used in this research report, is the BP algorithm.

3.6.1 The Backpropagation Algorithm

Wasserman (1989) pointed out that the advent of the BP algorithm has catalysed the NN resurgence interest. He added that BP is a very powerful technique used to train MLP-networks, and it is mathematically very strong and highly practical. Although it is not a panacea; the BP algorithm has also massively expanded the NN domain and demonstrated its success and due power, says the author. The BP algorithm proceeds in two phases in a supervised manner as described below.

The Forward Phase

In the forward phase, the input patterns are introduced into the network and after processing the outputs are produced. The synaptic weights of the network do not undergo any change; the input patterns are fed forward through the network, layer by layer until they reach the output layer. During this phase, changes are restricted to the activation functions and potential neuron outputs.

The Backward Phase

In the backward, the error signal resulting from comparing the network output against the desired response is propagated through the network, again, layer by layer, but this time in the backward direction. In this back course, successive adjustments are made to the synaptic weights to adapt the network and produce desirable outputs.

The BP algorithm can be summarised in these few steps by Wasserman (1989):

1. Selection of the paired input-target vectors from the training dataset; and application to the NN input nodes
2. Process the network output
3. Compute the errors between the network output and the target
4. Minimise the errors by adjusting the neuron weights connection
5. Iterate 1 to 4 for all the paired input-target vectors in the training set until a reasonable error is reached for the entire set.

Momoh et al. (1997) described the BP algorithm as follows: the weights w_{ij} of the network are adapted so as to minimise the error of the output; the i^{th} output o_i from neuron i is linked to the j^{th} input neuron by the interconnection weight w_{ij}. If the neuron k is not an input neuron then its state is given as follows:

$$O_k = f\left(\sum_i w_{ij} O_i\right), \tag{3.4}$$

where $f(x) = 1/(1 + e^{-x})$ is the sigmoid activation function; the summation is done in the entire contiguous layer over all the neurons.

If t is the target then the output neuron may be specified as follows.

$$E_k = \frac{1}{2(t_k - O_k)^2}, \tag{3.5}$$

where E is the error and k is the output neuron. The gradient descent algorithm adjusts the synaptic weights depending on the gradient error, that is:

$$\Delta w_{ij} = -\left(\frac{\partial E}{\partial O_j}\right) \times \left(\frac{\partial O_j}{\partial w_{ij}}\right) = -\left(\frac{\partial E}{\partial w_{ij}}\right), \tag{3.6}$$

and $\delta_j = -\left(\frac{\partial E}{\partial O_j}\right)$ is the signal of the error so that

$$\Delta w_{ij} = \varepsilon \delta_j O_i, \tag{3.7}$$

where ε is the learning rate parameter and δ_j is calculated according to the state of the neuron j being or not in the output layer. If it is in the output layer then we have

$$\delta_j = O_j(1 - O_j) \sum_k \delta_k w_{jk}. \tag{3.8}$$

To ameliorate convergence properties, the momentum rate α is included in the process so that

$$\Delta w_{ij}(n+1) = \varepsilon \delta_j O_i + \alpha \Delta w_{ij}(n), \tag{3.9}$$

where n is the number of epochs or iterations.

A BP flow chart proposed by Moghadassi, Parvizain and Hosseini (2009) is given in Figure 3.4 below.

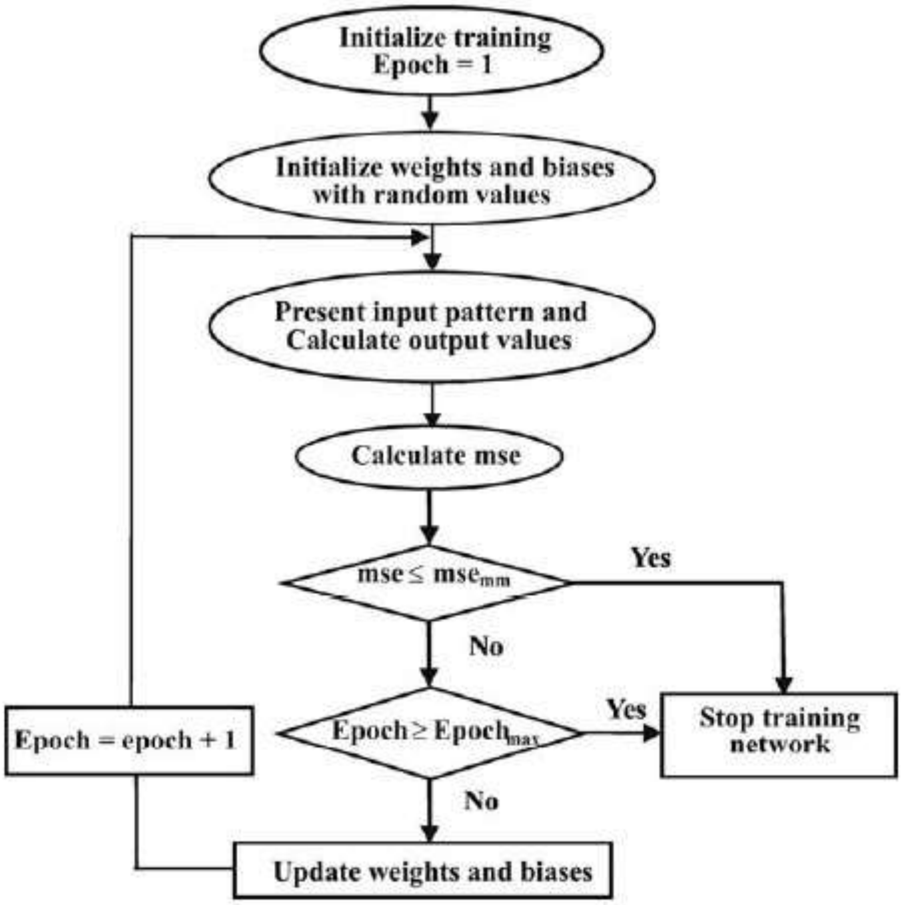

Figure 3.4 Backpropagation flow chart

3.6.2 Generalisation

The network can learn over the training phase, but the most important thing is that it should be able to generalise. Generalisation implies that the network can produce an output as close enough to the target as possible for a set of input patterns that have not been used during the training phase. The aim of generalisation is to reduce the error of the network output as much as possible with regard to out-of-sample input data.

3.7 Neural Networks Models in STLF

According to Kumar (2014), there are three categories of STLF NN models based on the forecasting target. These models are generally intended to forecast the load of the next hour, the daily maximum, the average load of the day, or just the complete daily load at one time.

Amral et al. (2008) added another classification according to the number of nodes in the input and output layers. According to this classification, NN models are categorised as either having many inputs and only one output or multiple inputs and multiple outputs as described below.

3.7.1 Multi-Input Single-Output Models (MISO)

In the work of Momoh et al. (1997), a MISO model is used, characterised by a simple feedforward MLP. The MISO model was the first NN model to be experimented and used in STLF. The network in this MISO model had a single output node providing the forecast for the peak (maximum) for the next 24 hours, the next day's total or average load, or the next hour's load. For a forecasting lead-time greater than one, Park et al. (1991) and Chen et al. (1992) used the forecasted output to feed back the same network together with the original input variables, in a dynamic recurrent manner. In fact, by doing so, the forecast of any arbitrary number of lead-time can be obtained.

3.7.2 Multi-Input Multi-Output Models (MIMO)

In this category Amral et al. (2008) proposed an NN model with 24 output nodes to predict a series of 24-hour-electric power at once in a 24-dimensional vector of output representing each hourly profile.

Murto (1998) built a variety of models based on MISO and MIMO topology and compared them one to another. He even constructed a Single-Input Single-Output (SISO) model, one network for each hour of the day. Results from this work, and as corroborated by Reddy and Momoh (2014), showed that the best NN model for STLF is the hourly model for forecasting an arbitrary lead-time one up to 24 hours.

3.8 Summary

We firstly introduced and defined the NN technique. Secondly, a parallelism was established between NN and statistical techniques. Thirdly, we explored NN architecture and topology. Next, we went through all the major learning processes and different learning rules. We lingered a little on the BP algorithm as this is the core of the MLP built model, and finally, the STLF NN-based models classification was elaborated on.

CHAPTER 4

METHODOLOGY

4.1 Introduction

In this chapter, we go through different steps necessary for building an STLF model using NN so as to make accurate predictions of hourly load forecasts up to 24 hours ahead. We started by the input data and input variables selection in which we looked at the correlation analysis and time lags. Next we went through the proposed model and looked at the design of the NN, the implementation of the model, the evaluation of the prediction performance of the model through different error metrics, the validation of NN, and finally we investigated the MLP built model in four different cases and compared it to a SARIMAX model.

4.2 Input Data

Eleven-year-load data, spanning from January 1rst 2000 to August 30th 2010, and temperature data were imported from Eskom databases, loaded in a Microsoft Excel spreadsheet and exported to MATLAB for the construction of the MLP model.

In the Excel spreadsheet, the data were organised in nine columns and 93480 rows or observations. The columns are: the day in format dd-m-yy, the hour coded as 0 to 23, the load in Megawatt, the year in format yyyy, the numerical months of the year, days of week coded as 1 to 7, South African public holidays, and the date in format ddmyy:hh:mm:ss.

The data were pre-processed in Excel prior to their exportation into Matlab. We checked for missing values, irregular values such as negative numbers and zeros, which do not make sense in this situation, and checked for some potential outliers.

4.3 Input Variables Selection

One of the most challenging tasks in constructing an NN is the selection of suitable network inputs. Since the dynamic behaviour of the network is highly dependent on the chosen input variables, the load must be highly correlated with these variables. It is also very important that the set of input patterns adequately represents all the factors influencing the system load. Thus, the process of selecting the relevant network inputs has to be guided by an intuitive knowledge of all the different factors affecting the load, along with a careful numerical validation of these assumptions.

In constructing the MLP model the input variables were selected based on the correlation analysis as suggested by Sinha (2000) and Taylor (2013). The temperature was added and used as a factor affecting the load. Some variables such as "day", "date" and "year" were simply discarded since they added little or no information. Hence, we retained the following variables from the original data: the temperature, the lagged load, the hour, days of week, the months of year, and public holidays.

4.3.1 Correlation Analysis

Based on the correlation analysis, we could identify the relevant input data variables. These are the variables that are highly correlated with the load data of the target hour and used in our model. This correlation analysis also allowed us to select the number of time-lags needed for the target hour load data since the previous hours load data have a strong influence on it (Lee et al., 1992)

4.3.2 Time Lags

It is demonstrated in the literature that loads of previous hours for a particular target have a strong impact on the load forecasts. But, "how far back in time are those loads" is determined by the correlation analysis, which we ran on the data, and noticed that the load inputs are highly correlated (more than 0.8) up to three lag times to the target hour load. The time lagged load data and temperature are given in Table 4.1 below.

Table 4.1 Time lagged input load and temperature

Target day		One – Day Lag		One – Week Lag	
L(t)	*NA*	*L(d-1, t)*	*T(d-1, t)*	*L(d-7, t)*	*T(d-7, t)*
L(d, t-1)	*T(d, t-1)*	*L(d-1, t-1)*	*T(d-1, t-1)*	*L(d-7, t-1)*	*T(d-7, t-1)*
L(d, t-2)	*T(d, t-2)*	*L(d-1, t-2)*	*T(d-1, t-2)*	*L(d-7, t-2)*	*T(d-7, t-2)*
L(d, t-3)	*T(d, t-3)*	*L(d-1, t-3)*	*T(d-1, t-3)*	*L(d-7, t-3)*	*T(d-7, t-3)*

In Table 4.1 above *L(t)* is the target hour load to be forecasted.

L(d, t) corresponds to the load on day d and target hour t

L(d, t – p), where p = 1, 2, 3 (for 1, 2, 3 hours before the target hour)

L(d-1, t), previous day same hour as the target hour

L(d-1, t – p) p =1, 2, 3 (for 1, 2, 3 hours before the target hour)

L(d-7, t), previous week same hour as the target hour

L(d-7, t – p) p =1, 2, 3 (for 1, 2, 3 hours before the target hour)

4.3.3 Model Input Variables

The input variables of the model are one to three hours before the target load and temperature data, same hour as the target and one to three hours before the target hour load and temperature data of the day before, same hour as the target and one to three hours before the target load and temperature data of the week before. In short the input variables consist of historical load and temperature data, the type of the day, the months of the year, and public holidays (South African public holidays) were taken into account to improve the accuracy of the load forecasting. Most of the input variables were selected based on the autocorrelation apart from the calendar variables. The structure of all input variables used in our model is given in Table 4.2 below.

Table 4.2 List of input variables

Load	**Temperature**	**Date**
L(d, t-1)	*T(d, t-1)*	Hour of day
L(d, t-2)	*T(d, t-2)*	Day of Week
L(d, t-3)	*T(d, t-3)*	Month of year
L(d-1, t)	*T(d-1, t)*	Pub. Holiday
L(d-1, t-1)	*T(d-1, t-1)*	

L(d-1, t-2)	*T(d-1, t-2)*	
L(d-1, t-3)	*T(d-1, t-3)*	
L(d-7, t)	*T(d-7, t)*	
L(d-7, t-1)	*T(d-7, t-1)*	
L(d-7, t-2)	*T(d-7, t-2)*	
L(d-7, t-3)	*T(d-7, t-3)*	
Avg[L(d – 1)]		

Avg [L(d – 1)] is the average of the load of the day before the target day

T(d, t) corresponds to the temperature on day d at hour t, and the same applies here as in load data.

The days of the week were coded as 1 to 7, with 1 for Sunday and 7 for Saturday. The public holidays were coded as 1 and working days as 0. Table 4.3 below shows only the days of week and their coding values.

Table 4.3 Days of Week Coding Values

Days of Week	Sunday	Monday	Tuesday	Wed	Thursday	Friday	Sat.
Coding Value	1	2	3	4	5	6	7

4.4 Proposed Model

4.4.1 Model Design

The model we built, in this research report, consisted of a feedforward MLP network using a BP algorithm with the gradient delta learning rule, a nonlinear sigmoid function as a transfer function in the hidden layer and the Purelin function at the output layer to allow the network to produce a wide range of output. The MATLAB R2015b NN package with its built-in learning function (Levenberg-Marquardt) was used because it has the best learning rate. The steepest gradient function and a momentum were also used and the learning rate set to the default value with the possibility to adjust automatically along the training process. One hidden layer was used, but different numbers of hidden neurons were carried out based on trial and error, before retaining 25 hidden neurons as a structure that produced the minimum error.

4.4.2 Cross-Validation

During the training phase, we used the cross-validation technique to prevent the fall into one of the drawbacks of NN models, such as overparameterization that results in overfitting because of the model complexity. Overfitting occurs when a model fits the data so well to an extent it includes the noise and ends up by yielding inaccurate forecasts. At this stage of knowledge, the adequate training sample size proportional to the number of network weights has not been formally established so that it is difficult to tell how many parameters are too many for a given number of data points in the sample. However, to avoid overfitting, we used the cross-validation method, which consists of splitting the data into a training, validation and test sets.

4.4.3 Evaluation of Prediction Performance

After the designing procedure and running the MLP model, the forecasting performance of the trained network could be assessed by calculating the prediction error on samples other than those used during the training phase. Various error metrics between the actual and forecasted loads are presented and defined in the literature, but the most commonly adopted by load forecasters are the Mean Absolute Percentage Errors (MAPE), the Absolute Percentage Errors (APE), the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) or Root Mean Squared Error (RMSE).

$$\boldsymbol{MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|actualLoad(i) - forecastedLoad(i)|}{(actualLoad)(i)} \times 100}, \qquad \boldsymbol{(4.1)}$$

$$\boldsymbol{APE = \frac{|actualLoad - forecastedLoad|}{(actualLoad)} \times 100}\,, \qquad \boldsymbol{(4.2)}$$

$$\boldsymbol{MAE = \frac{1}{n}\sum_{i=1}^{n}\frac{|actualLoad(i) - forecastedLoad(i)|}{(actualLoad)(i)}} \quad , \qquad \boldsymbol{(4.3)}$$

$$\boldsymbol{MSE = \frac{1}{n}\sum_{i=1}^{n}(t_i - O_i)^2 \;\; or \sqrt{MSE}} \quad , \qquad \boldsymbol{(4.4)}$$

where *n* is the number of the data points and *i* is the period at which the load is produced or forecasted, *t* is the target and *O* the NN output.

To make sure that the system is accurate, the relative error is retained on the hourly basis. In the case of positive error, it means the forecasted load is greater than the actual consumption load, and the opposite is true when the forecasted load was less than the actual load.

Besides the error metric to evaluate the performance of the NN models, a particular attention was given to different plots generated by the training process such as the regression plots, the performance function versus epochs (number of times training vectors were used to update the weights), the training state plot, and the forecast and actual data comparison plot as suggested by Buhari and Adamu (2012).

4.4.4 Neural Networks Validation

Sandoval (2002) suggested a couple of methods to validate an NN model. He stated that the performance of an NN model must be compared to that of some well accepted techniques such as in the following manner:

a) Compare the performance of the NN model with some 'naïve' method considered as a benchmark, or good standard method such as fuzzy engines, regression, ARIMAX, or other NN, etc.

b) Comparison must be based on test samples performance.

c) Test samples must be representative enough to allow inferences to be drawn.

d) Evaluate the error by using standard metrics such as the MAPE, MSE, APE, and the MAE among others.

In this research work we compared our MLP model to a SARIMAX model in terms of the MAPE, MAE, MSE, APE and Daily Peak Error.

4.5 Model Investigation

We first started by analysing the load profiles so as to get a good sense of the shape of the load curve and draw some characteristics of the load profile. To do so, we took one year-data (data for the year 2000), at the beginning of our data collection, and plotted it against the time. Next we investigated four different cases in different seasons of the year to test the proposed model and report the results. Finally, to validate the NN technique we compared the built MLP model to a SARIMAX model constructed using the entire and same training and testing datasets and reported the results.

4.6 Summary

In this methodology chapter, we presented the techniques and material used to build the MLP model and how we used them

CHAPTER 5

LOAD PROFILE ANALYSIS - RESULTS AND DISCUSSION

5.1 Characteristics of the Load Profile

Before we ran our MLP model on the data, we looked at the characteristics of the load profile since this constitutes the uniqueness of every power utility.

First, we displayed all the data as a time series from 2000 to 2010 in Figure 5.1 so as to get a sense of the general shape of the electricity consumption in South Africa during this time. This data represents the aggregate quantity of electricity that was consumed hour by hour in South Africa.

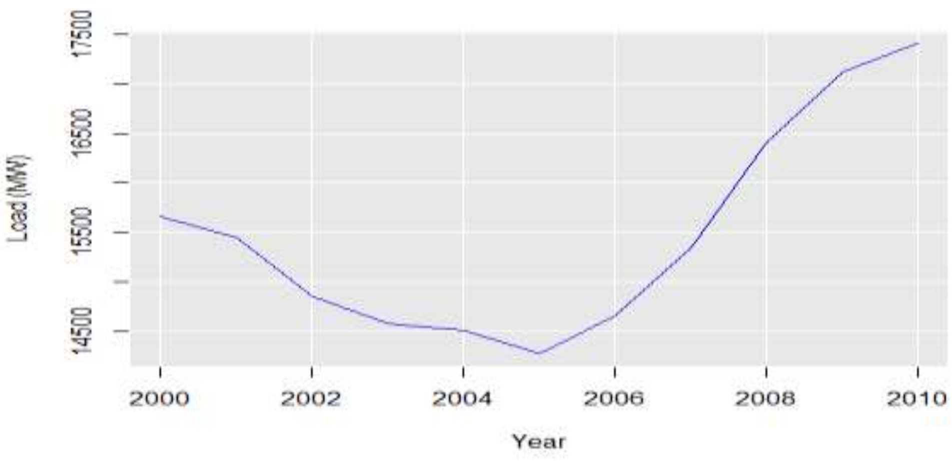

Figure 5.1 Electric load in Megawatts from 2000 to 2010

It can be seen that the load was very low in 2005 comparatively to other years reported in Figure 5.1 above. Next, in Figure 5.2 below, we focused on one-year data for a more detailed description of the characteristics of the load variations against the time.

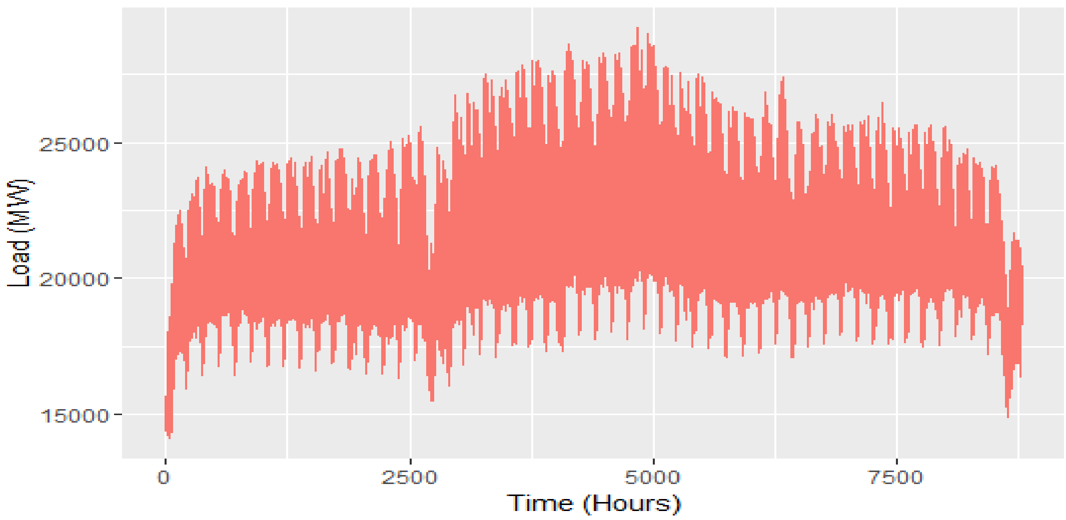

Figure 5.2 Electric load in Megawatts from 1st Jan – 31st Dec 2000

It can be seen in Figure 5.2 that the load profile curve starts at a low point because the first day in 2000 was a weekend and a public holiday. On weekends and public holidays, generally, industrial and social activities are at low levels. Then the curve goes up and reaches a sort of steady cycle. Then comes a break in the patterns (trough) around 2500 hours, which corresponds to a transition season (March-April) in South Africa. The curve drops significantly down during this transition season and then resumes its shape going up steadily again with small breaks in the patterns here and there, between 4000-5000 hours, during winter, and starting to drop down slowly until the end of the year, and the same patterns repeat. That is, the load profile exhibits seasonality and cycles.

We also displayed the load autocorrelation charts in Figure 5.3 to get the sense of the dependency at different time lags.

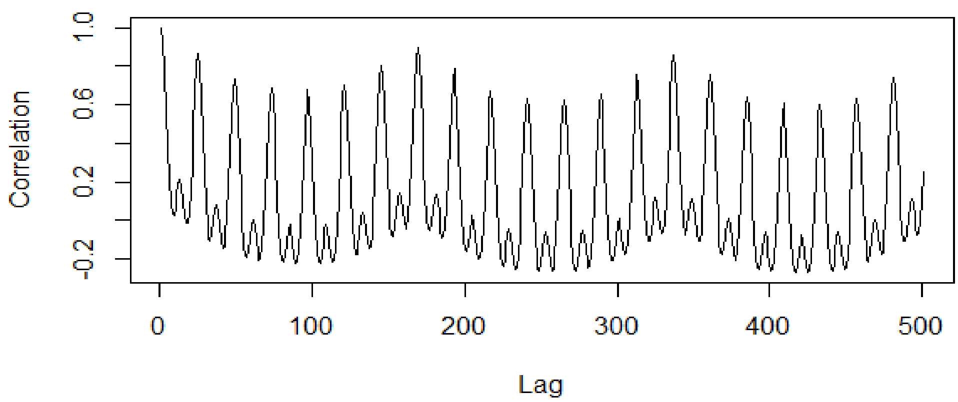

Figure 5.3 Electric load sample autocorrelation for the first 500 lags

We can see from the graph in Figure 5.3 that there are seasonal effects regularly shaped with peaks at 24, 48, 72, … corresponding to the daily activities, and similar patterns at the multiples of 168 for weekly activities. These weekly seasonal effects come from working days as industrial activities are at high level. If we could display more lags, then we would also observe the monthly seasonal effects.

The temperature as a weather variable and the load profile for a period of two weeks are respectively displayed in Figure 5.4 and Figure 5.5 below for a good parallelism.

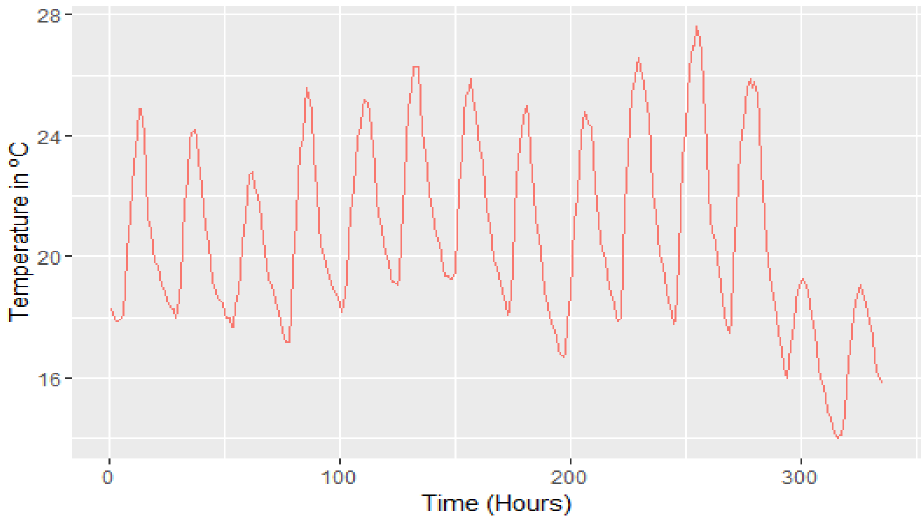

Figure 5.4 Temperature during 15th – 30th Jan 2000

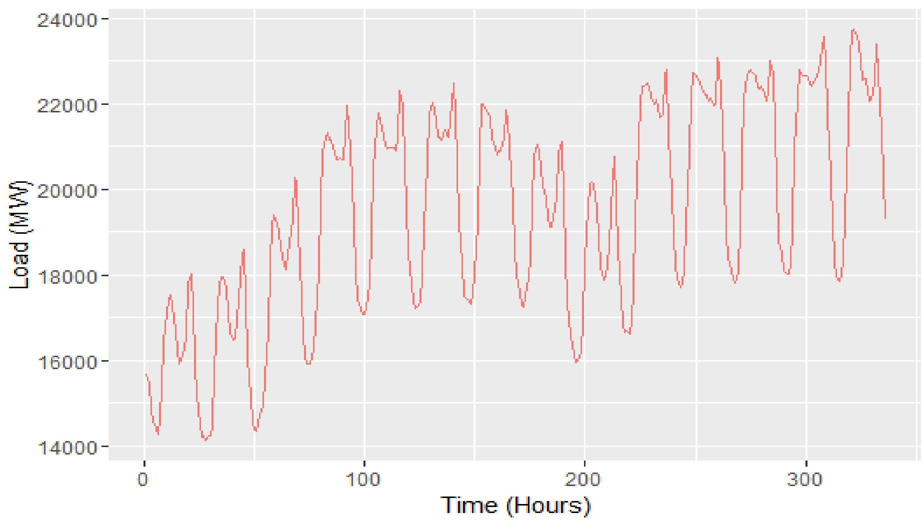

Figure 5.5 Load profile during 15th to 30th Jan – 2000

Indeed, a comparison of Figure 5.4 and Figure 5.5 gives an interesting picture regarding the behaviour of these two curves (load and temperature). Observation can be made that for the first two days of this period (around 50 hours), when the temperature is relatively high, around

24ºC, the load curve corresponding to the same time is low. After a day drop in temperature, it can be seen that from then onward the two curves vary in the same direction, meaning that when the temperature goes up, the load also goes up, when down, they both go down. But it should be borne in mind that the load curve does not always behave in this way. The load curve can go down when the temperature curve is up, as it can be seen at the beginning of the period 15th – 30th Jan 2000, but this same curve will raise up if the temperature keeps going down (because more heating will be needed). The load curve will go up when the temperature is up because of VAC (Ventilation and Air-Conditioning) needed.

The first day in Figure 5.4 is a Saturday. It can be clearly seen that the curve starts at a low point since it is a weekend during which activities are very low, that explains why the two first portions patterns of the curve are similar. Then comes Monday when activities tend to get back to normal, i.e. normal working days, normal behaviours meaning high-level of activities. For this particular week after the summer holiday, the first Monday seems to start a bit slowly compared to the Monday of the week after where the five working (Monday to Friday) days exhibit quite high similar patterns. These weekly similar patterns are then repeated through the succeeding weeks.

As to the daily working pace, things are almost the same because different activities take place synchronically. Working hours, lunch time, leisure time are the same for the majority of people, and they all sleep at night. This is the reason why the load will be high during the day and significantly low during the night. However, it should be noticed that this pace of life varies throughout the year and impacts the load profile as shown in Figures 5.6 (a-d) below.

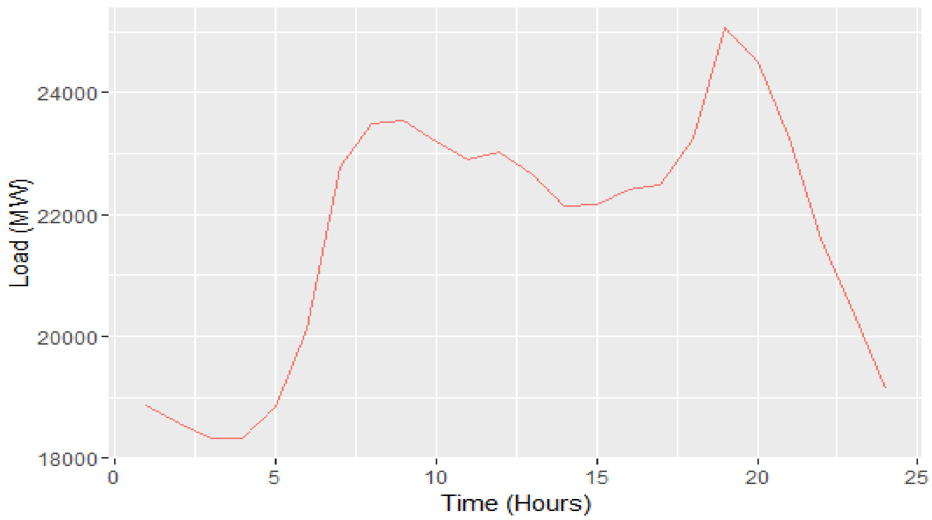

Figure 5.6 (a) Load profile on Wednesday, 19th April 2000

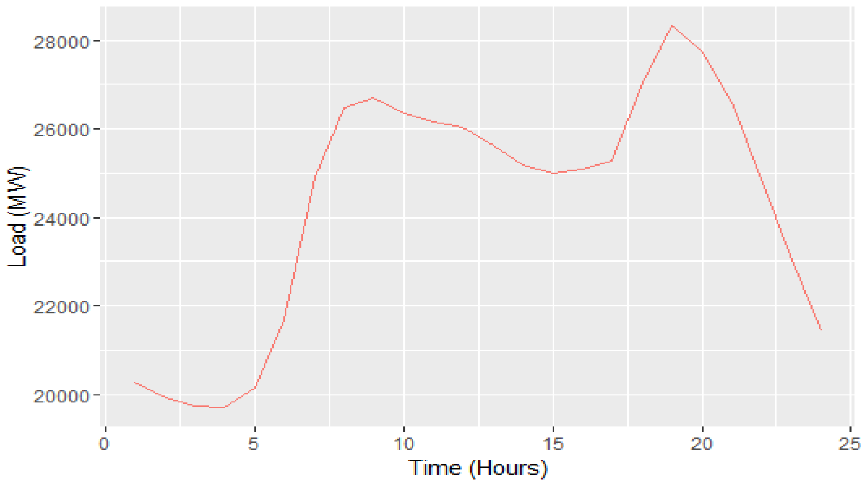

Figure 5.6 (b) Load profile on Wednesday, 21st June 2000

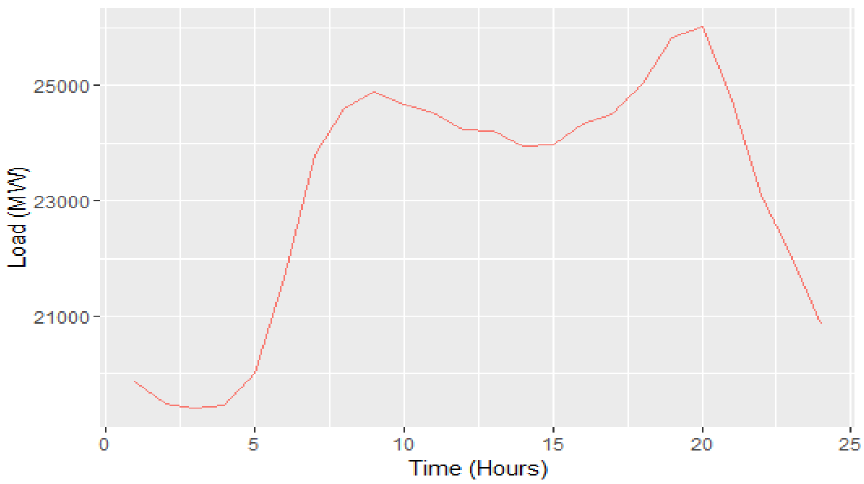

Figure 5.6 (c) Load profile on Wednesday 11th October 2000

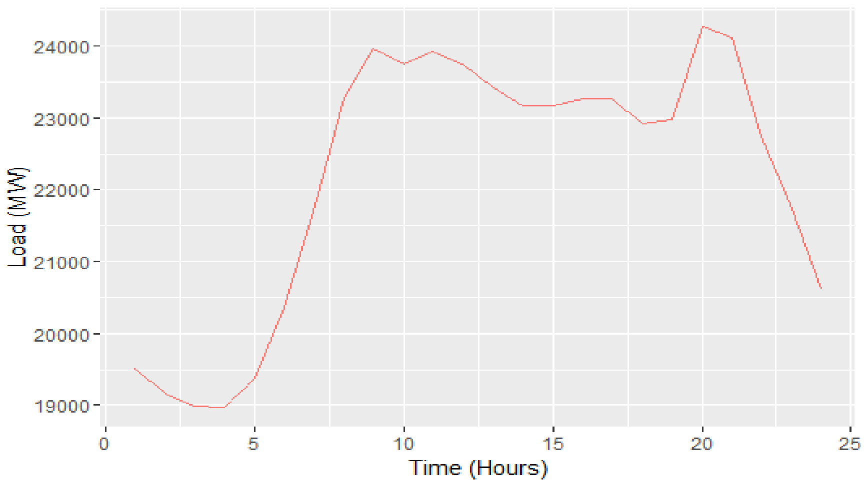

Figure 5.6 (d) Load profile on Wednesday, 13th December 2000

In Figures 5.6 (a-d) we chose the load profile on a Wednesday because Chikobvu and Sigauke (2012) showed that Wednesdays have the highest index of the daily demand seasonal indices, plus they are not influenced by the weekend. We kept the same day of the week in all different seasons so as to make consistent comparisons. We also made sure that the chosen day was not a public holiday as this gives a particular load profile similar to a weekend one.

These four graphs displayed in Figure 5.6 are clearly different, despite the fact that a conciliatory opinion may classify the first two load profiles in one category and the other two in another, based on their shapes. This fact demonstrates sufficiently that there are differences in the same day load profile in different seasons, and there are of course differences between days of the week in the same season. This is why it is advised, in the literature, to classify days of the week into different types in the load forecasting area, since each day has its own characteristic load patterns. This is especially true on Saturdays and Sundays and public holidays, which tend to have their own particular load profile.

If we go back to Figure 5.5 and take a close look at the five working days load profiles, we can see that although they all tend to be similar, Mondays and Fridays have a slightly different profile than the other working days. These two days of the week are closed to the weekend and may undergo its effects.

In short, we can see that our load profile exhibits some seasonality behaviours and cycles patterns. It can also be noticed that the load is strongly positively correlated with the temperature during summer hot days and the opposite, i.e. negatively correlated with temperature, in winter cold days.

5.2 Load Forecasting Results and Discussion

In this section we focused on the results of hourly load forecasts obtained by using a trained NN and designed as follows: ten input nodes, one hidden layer with 25 neurons obtained by trial and error, and one output layer. Real time data that includes historical hourly load consumption and temperature data collected from Eskom over eleven years from 2000 to 2010 were used to train and test an hourly load forecasting MLP model. The data was randomly partitioned with 70% used to train the model and 30% of the data used in the validation and testing phases of the model. The NN model was trained using an MLP using Matlab R2015b. The Matlab code used in this research report was inspired by Ameya (2010) and can be found in appendix A.

In the next few lines, we summarised the training process and presented its results. Basically, after the division of the data, the training phase took place during which the bias and weights were produced. During the training phase, Matlab displayed the NN toolbox (Figure 5.7) to inform us of what was going on behind the scenes.

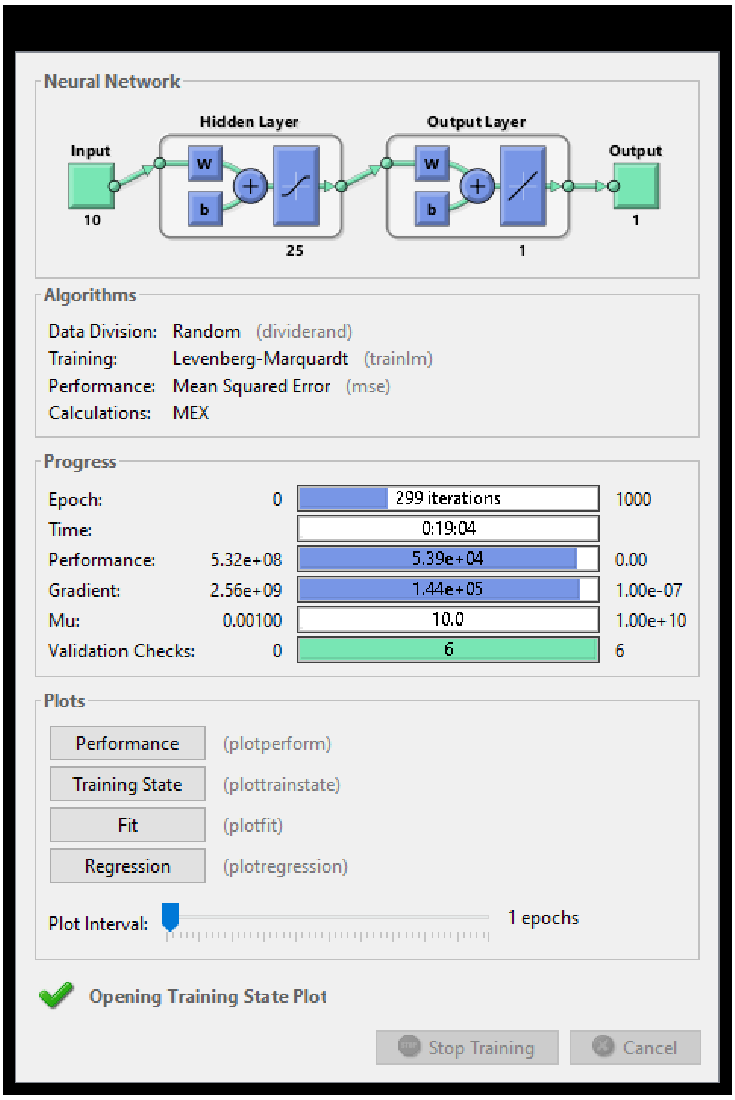

Figure 5.7 Matlab NN toolbox during the training phase of our MLP model

Figure 5.7 shows that the NN toolbox is divided into four parts:

a. The first part shows a schematic structure of NN model as it is designed, the inputs variables as given in Table 4.2, 25 hidden neurons and one output node forecasting one hour at a time;
b. The second part, algorithms, shows all the algorithms that are involved in the data division, network training, training performance evaluation and derivative computation;
c. The third part is where we can see the progress of the training mechanisms, i.e. the number of epochs, the elapsed time, the performance (how well is the training going), the gradient function value and validation checks;
d. The last part offers a possibility to monitor the network performance graphically.

After a successful accomplishment of the training stage, three plots were produced by the Matlab NN toolbox namely, the regression plots, the performance function versus epochs plot, and the training state plot.

i. The regression plot

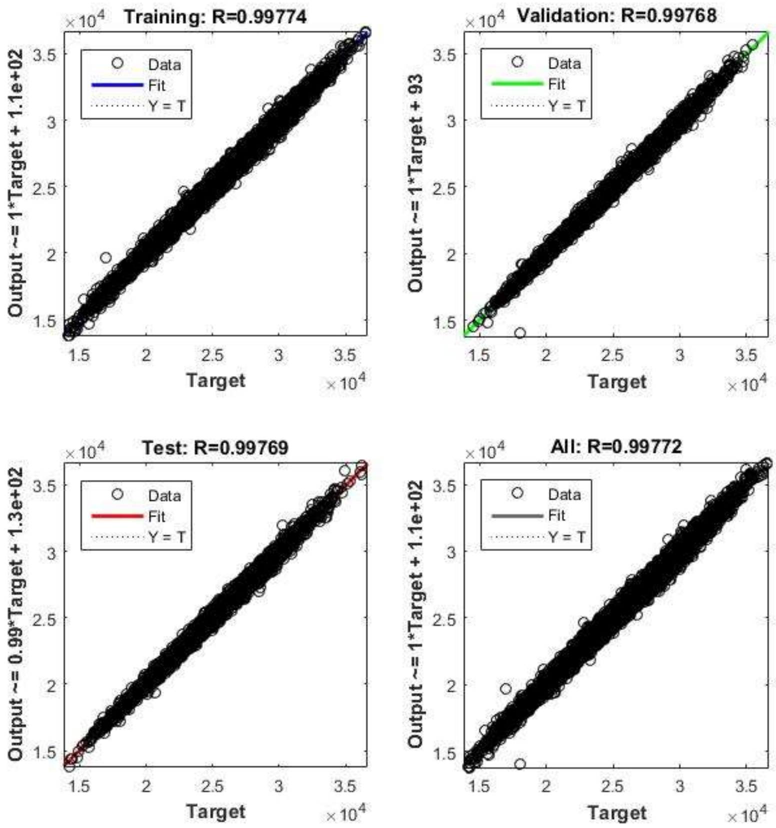

Figure 5.8 NN Toolbox Regression plots of the MLP model

The four plots in Figure 5.8 showing the output of the training data set against the target, the output of the validation data set versus the target, the test data output versus the target, and the overall network output data against the target. These plots from Figure 5.8 show how strong the data output and the target are correlated and how accurate the trained network model were able to forecast after learning some complex relationships between the input variables and the target.

ii. The performance function (MSE) versus the number of epochs plot

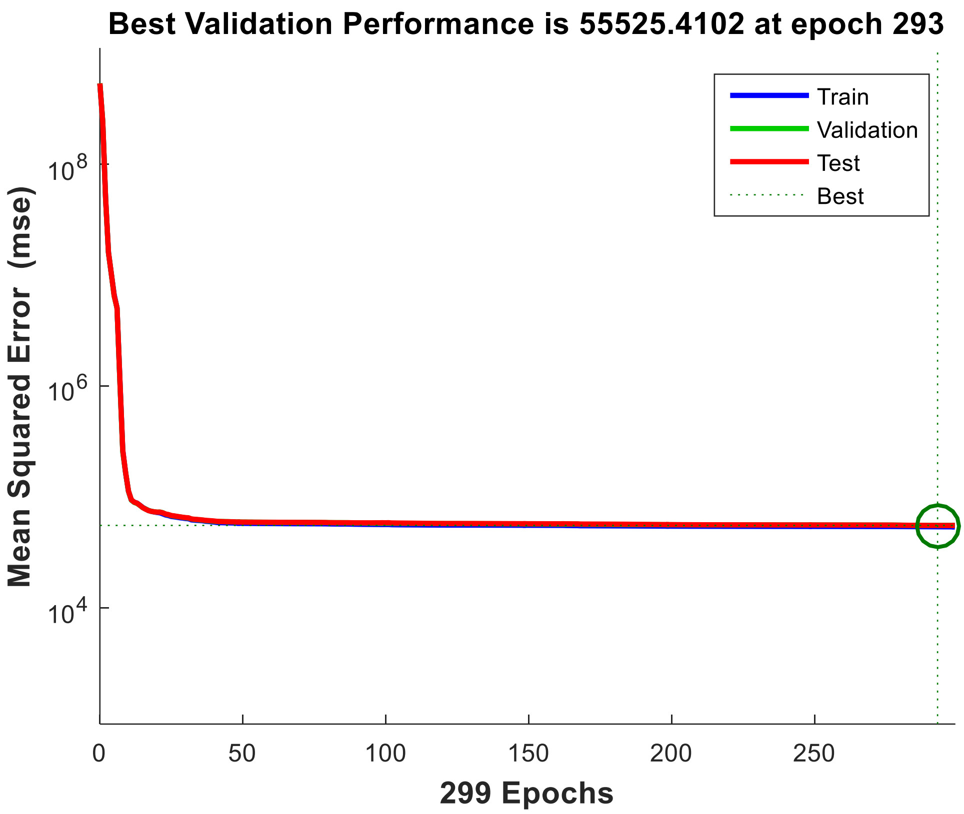

Figure 5.9 NN Toolbox Performance function of our MLP model

Figure 5.9 shows a plot of the performance function, which is the Mean Squared Error (MSE) by default in Matlab NN toolbox, versus the number of epochs or iterations. A stopping criterion that can spot a change in the course of the learning algorithm is used by using the data from a test-set after every ten iterations (epochs). Whenever the fit error minima for the test-samples is detected, the learning algorithm is stopped to avoid overfitting the model. This error minima may signal the transition between under-fitting and overfitting of the model.

iii. The training state plot

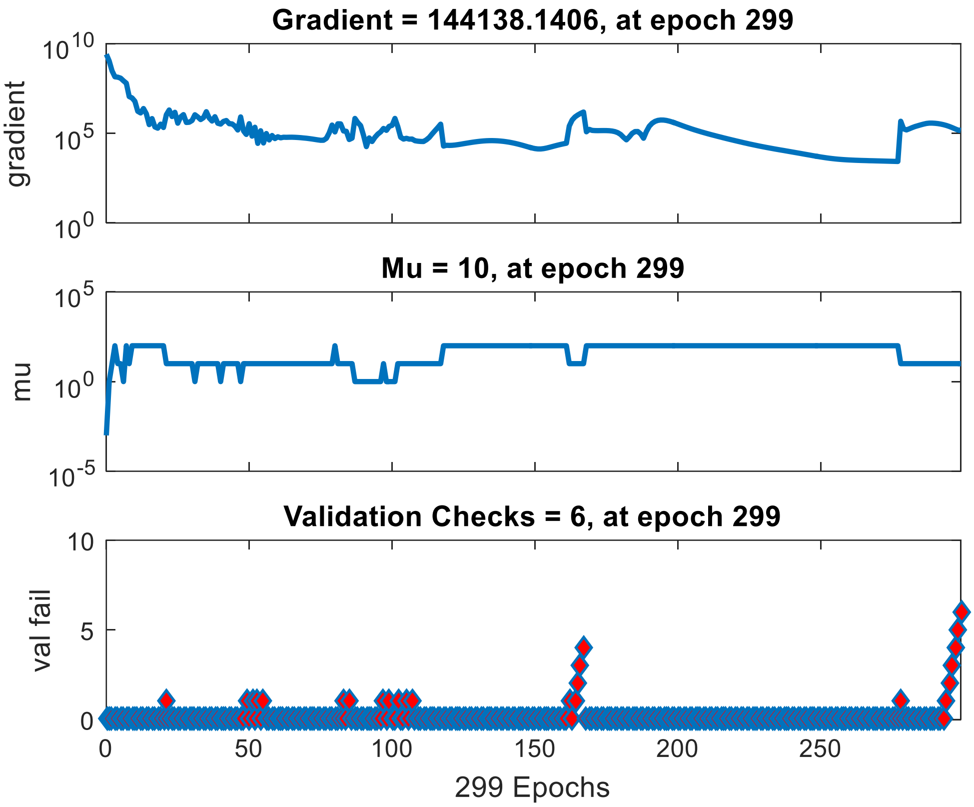

Figure 5.10 NN Toolbox Training state plot of our MLP model

Figure 5.10 has three different plots consisting of a first plot of the learning function against the number of epochs for essentially displaying the development of the gradient function values as the number of iterations increases. The next plot consists of the learning rate (mu) versus the number of iterations (epochs) to keep under observation the learning rate trend so as to see when the network error decreases along the training process. Finally the plot of validation checks, which is implemented automatically for whenever an abrupt modification occurs in the gradient function computation.

5.2.1 Case Studies

Four arbitrary cases in different four seasons of the year were investigated to test and validate the proposed MLP model. The forecasts of one hour up to 24 hours were performed based on the daily, hourly data without using the target hour load data. Then, we ran the MLP built model on the data to forecast one hour at a time and 24 times recursively for a day ahead. The obtained forecasts were compared to the real load data and the relative errors were calculated.

Case I: Hourly Forecasting in August 2009

Figures 5.11 through 5.14 show the actual and forecasted load (FL) profiles of the days in Table B1 in Appendix B.

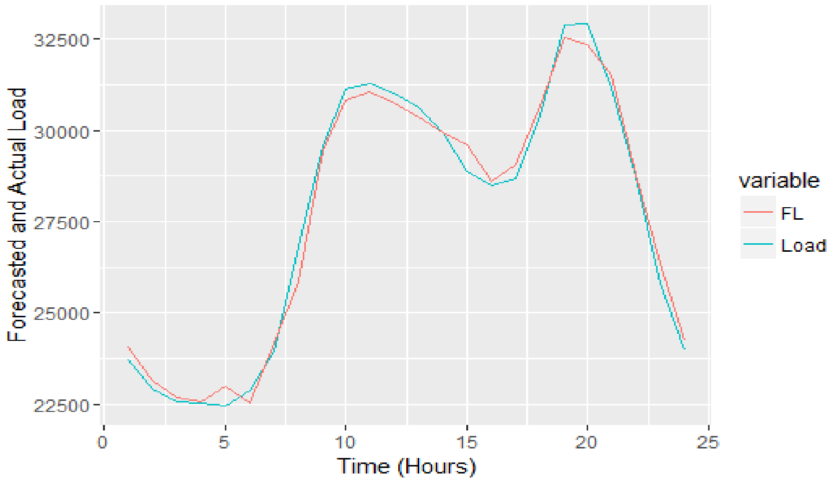

Figure 5.11 Actual Load and FL on Sunday 2nd August 2009

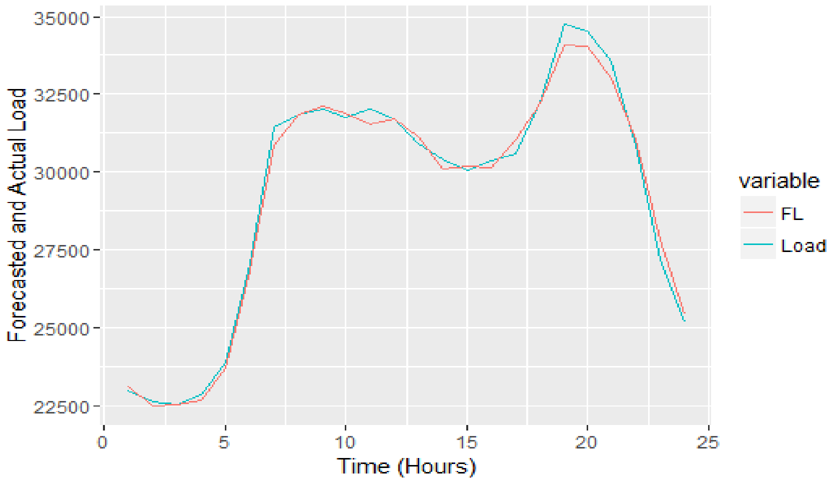

Figure 5.12 Comparison of actual load and FL on Monday 3rd August 2009

Figures 5.11 and 5.12 above establish a comparison between the actual load and the FL curves in terms of their shapes. It can be seen, in the first graph that the FL failed to map the actual load curve shape, in the early hours of the day, at the peak hours of the day and at the valley of the curve between 13h and 16h. In the second graph, the FL was a little bit poor from 7h up to the peak time around 19h, with a big gap, meaning large errors, occurred around this time of the day.

Figures 5.13 and 5.14 below display a comparison of the actual load and FL curves during same month of August 2009 but this time for a Wednesday in the middle of the week and on Saturday, a weekend.

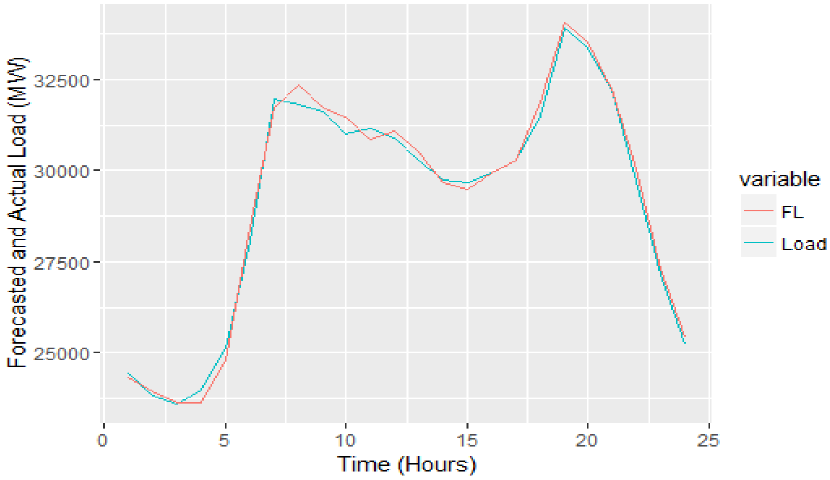

Figure 5.13 Actual Load and FL on Wednesday 5th August 2009

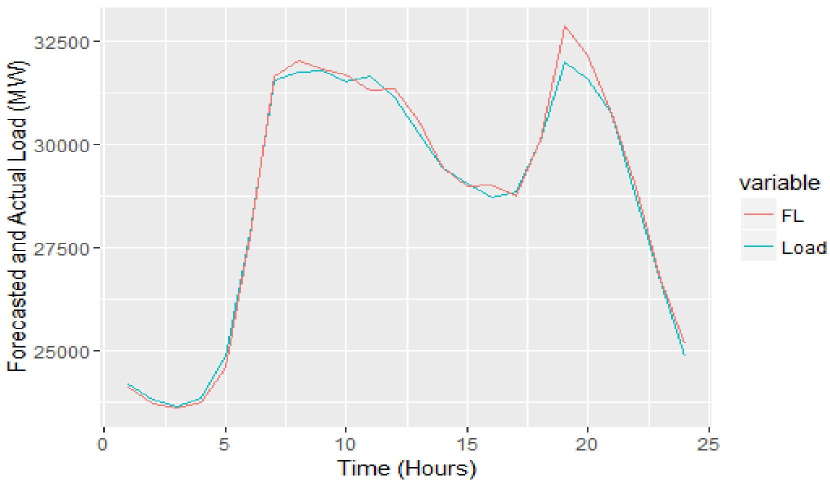

Figure 5.14 Actual Load and FL, Saturday, August the 7th 2009

In this first case study, from Figures 5.11 to 5.14, we can see that the MLP curve performed poorly following the shape of the actual load curve around the peak hours, i.e. when the load demand is high (7h – 10h, 19h – 20h), especially at night when the difference between the two curves is relatively noticeable except for Wednesday (5th August 2009), Figure 5.13.

Figure 5.15 below gives a superimposed view of the load behaviour during the first week of the month of August 2009.

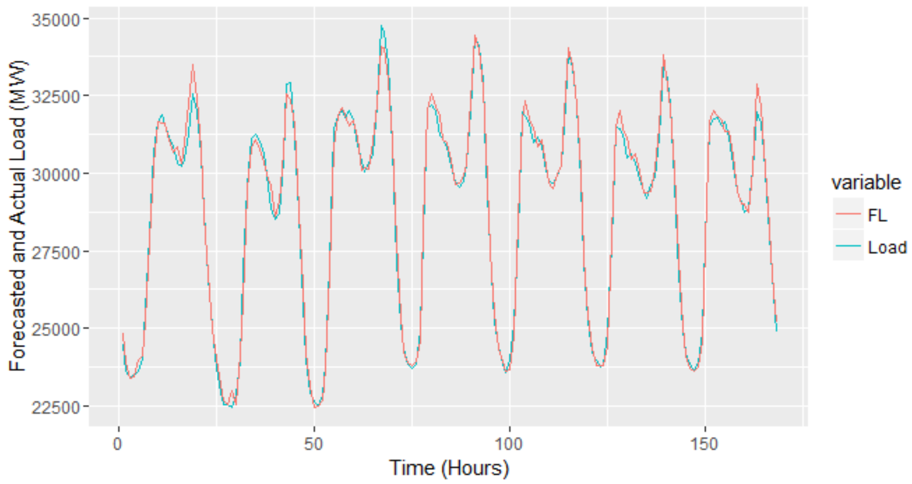

Figure 5.15 Actual Load and FL for a week: 1st – 7th August 2009

It can be seen in Figure 5.15 above that in general, the MLP model performed well as the gap between the actual and forecasted load curves is relatively small, except for some cases during the peak hours at the beginning of the week until Tuesday, but could easily recover from Wednesday onward up to Saturday.

The recorded MAPE (0.77%) in this first case study is less than the one obtained in (Park et al., 1991; Lee et al., 1992; Yoo and Pimmel, 1998), and the other error metrics are as follow: the RMSE is 292.29 MW and the Daily Peak Error varied between 0.08% and 1.85%.

Case II: Hourly Forecasting in October 2009

Table B2 in Appendix B gives one way of looking at the MLP model hour by hour, through its outputs (FL), the actual load compared to the FL, and the resulting APE. Whereas the corresponding Figures 5.16 through 5.19 give yet another way of looking at the shape of the forecasted and actual load curves so as to see their differences.

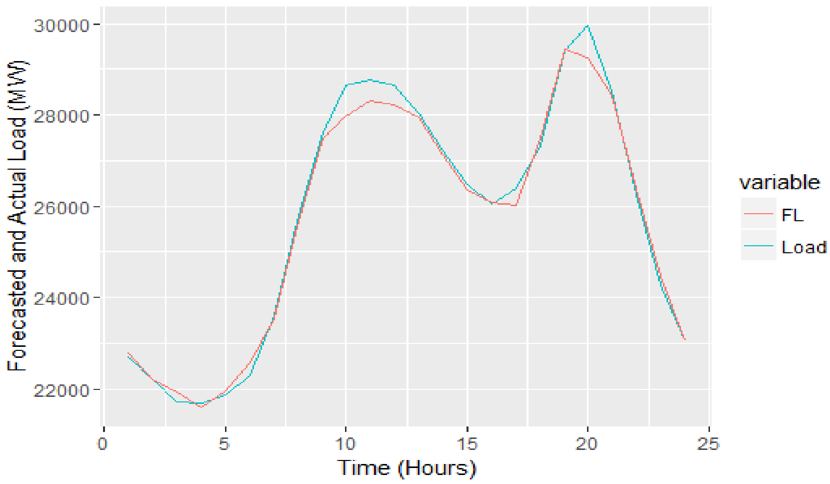

Figure 5.16 Actual Load and FL, Sunday, October 4th 2009

In Figure 5.16 above, it can be noticed that the separation between the actual load and the FL curves is very big around the peaks and the valley of the two curves, and almost reasonably confound elsewhere, providing evidence that the FL is performing well in forecasting at these intervals in time. The subsequent graphs are telling another story in terms of the performance of the FL. There is a lot of fluctuations as can be seen in Figures 5.17 through 5.19 below.

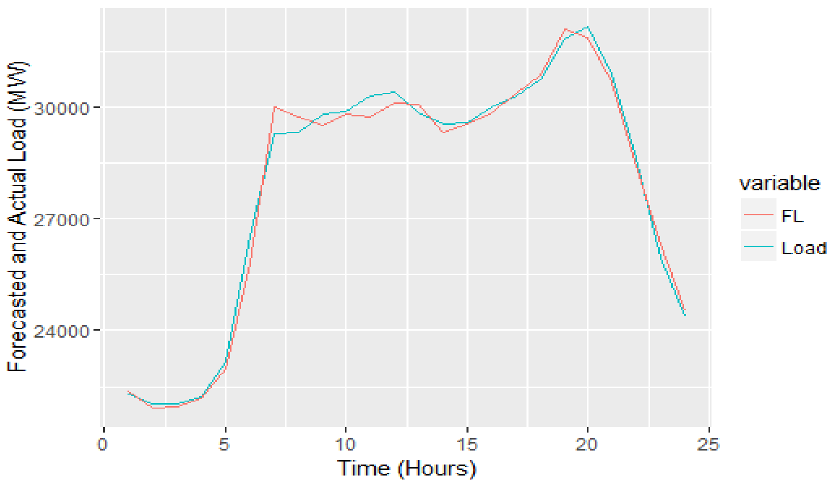

Figure 5.17 Actual Load and FL on Monday the 5th October 2009

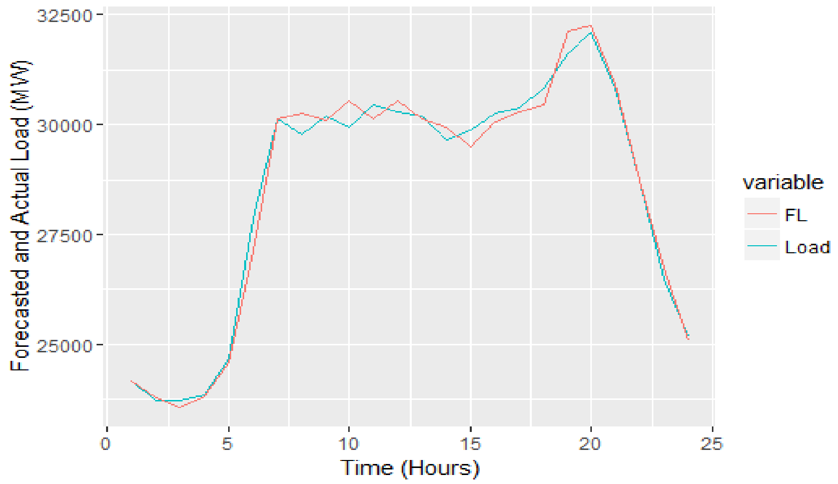

Figure 5.18 Actual Load and FL on Wednesday, October 7th 2009

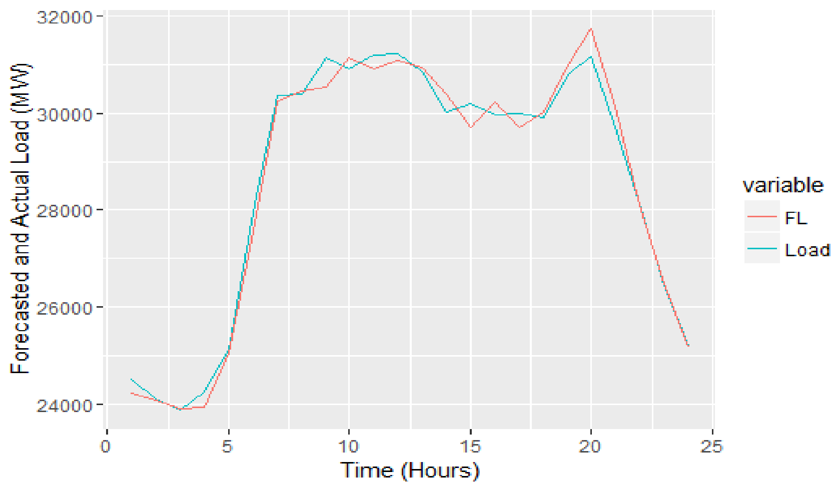

Figure 5.19 Actual Load and FL, Friday 9th October 2009

In the second case study, Figures 5.16 through 5.19 depict the behaviour of the MLP model during four randomly chosen days in October 2009. Here, it can be noticed that the MLP model has some shortcomings in forecasting during the peak hours, the profile curves are not as smooth as the previous ones in the month of August for the same corresponding days of the week. There is a lot of fluctuations of the load curve that the MLP model could not track, especially in Figures 5.17 through 5.19 (5th, 7th and 9th of Oct. 2009). The huge fluctuations of the load profile might have been due to the fact that October is during a transition season in South Africa as stated Banda and Folly (2007). During this period of time the mornings and evenings are very cold and the afternoons are very hot on some days.

Figure 5.20 below gives a global picture of the forecasting behaviour during a week in October 2009.

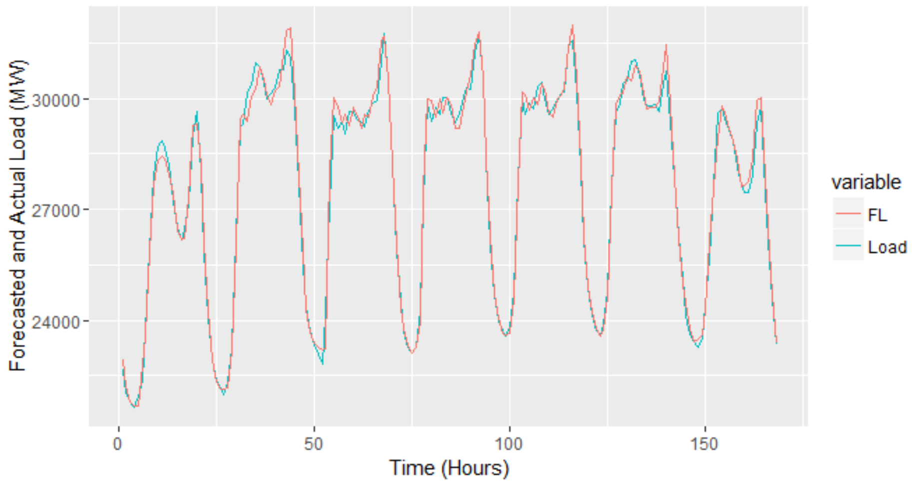

Figure 5.20 Actual Load and FL, week 11th – 17th October 2009

The FL profile has a good shape as it closely traces the load profile during weekdays but some irregularities of the curve at the beginning of the week during peak hours can be observed in Figure 5.20 above.

Once again we compared the forecasted and actual load curves on different days of the week during this month of October throughout Figures 5.16 to 5.20, and results showed that the MAPE is 0.80%, the RMSE is 295.006 MW and the Daily Peak Error ranges between 0.15% and 3%. These results are more accurate than the 2.2% of MAPE achieved in the work of Osman, Awad and Mahmoud (2009).

Case III: Hourly Forecasting in December 2009

Four different days in December 2009 were arbitrarily selected to compare the actual and forecasted load based on their day of the week profile. The corresponding hourly load, FL and APE are given in Table B3 in Appendix B.

Results in the aforementioned table show that the APE is very small on average for the Sunday 6th December profile as can be corroborated in the corresponding Figure 5.21 below with a satisfactory forecasting shape following the actual load curve.

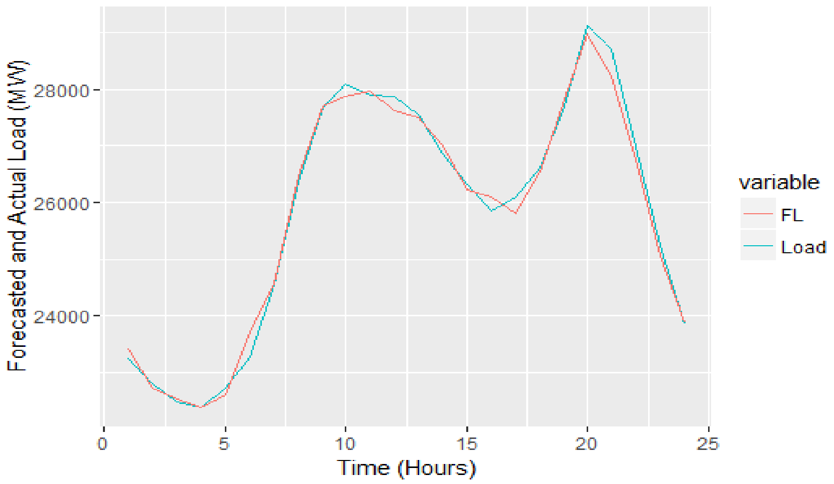

Figure 5.21 Actual Load and FL, Sunday, Dec 6th 2009

Figure 5.22 to Figure 5.24 below depict the forecasting model behaviour on some particular days given in Table B3 in Appendix B.

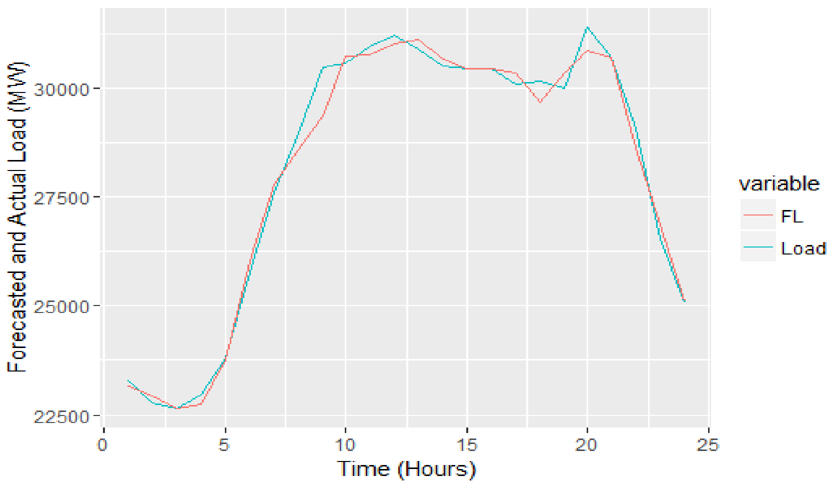

Figure 5.22 Actual Load and FL on Monday 7th Dec 2009

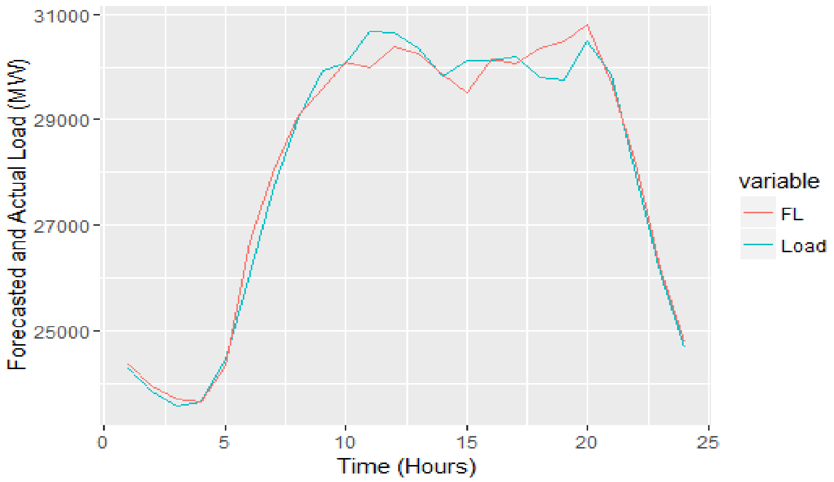

Figure 5.23 Actual Load and FL on Wednesday 9th Dec 2009

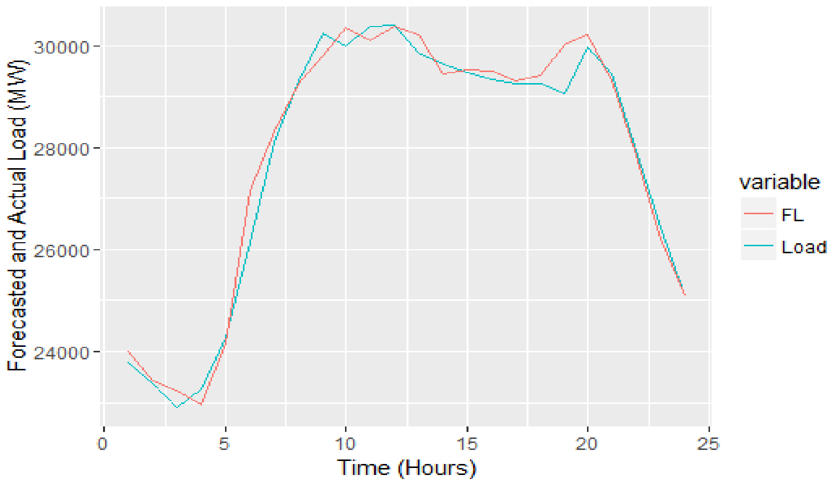

Figure 5.24 Actual Load and FL on Friday 11th Dec 2009

The third case study, in the month of December 2009, depicts the forecasting model behaviour on some randomly chosen days given in Table B3. It can be seen in Figures 5.22 and 5.24 that the forecast MLP model on Monday and Friday, respectively, performed poorly in forecasting the actual load profile at the peak hours. Table B3 gives two higher APE values, on average, for these days. Figure 5.23 corresponding to Wednesday shows that the model did not perform well during the peak hours and Table B3 gives a couple of higher APE values of this profile for this particular day.

Figure 5.25 below gives a sense of how well the MLP model performed on the global point of view for seven days period of time in December 2009.

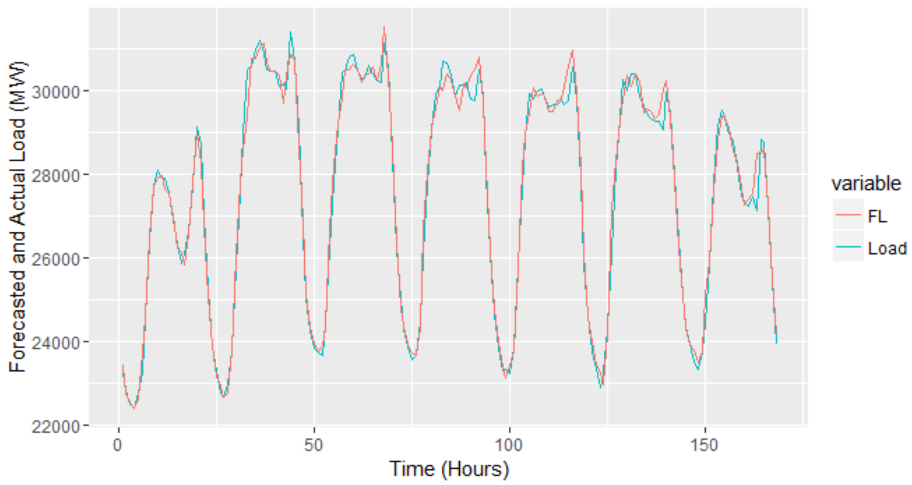

Figure 5.25 Actual Load and FL for 6th – 12th Dec 2009

From a superimposed view, Figure 5.25 shows that the MLP model has some shortcomings during the working days from Tuesday to Thursday but did better from Friday and all the weekend long through Monday.

The forecasted and actual load curves were compared in this case study, and results were 0.90% of the MAPE, the RMSE is 315.209 MW, the Daily Peak Error between 0.18% and 2.66% for December 2009. The performance errors of this case study is the poorest compared to other cases, but still less than the results reported in most of the literature on STLF we surveyed.

Case IV: Hourly Forecasting in March 2010

Hourly load, FL and APE of four arbitrary selected days in March 2010 are given in Table B4 in Appendix B, which gives various APE values and Figures 5.26 through 5.29 below give graphical views from which we can trace the performance of the MLP model.

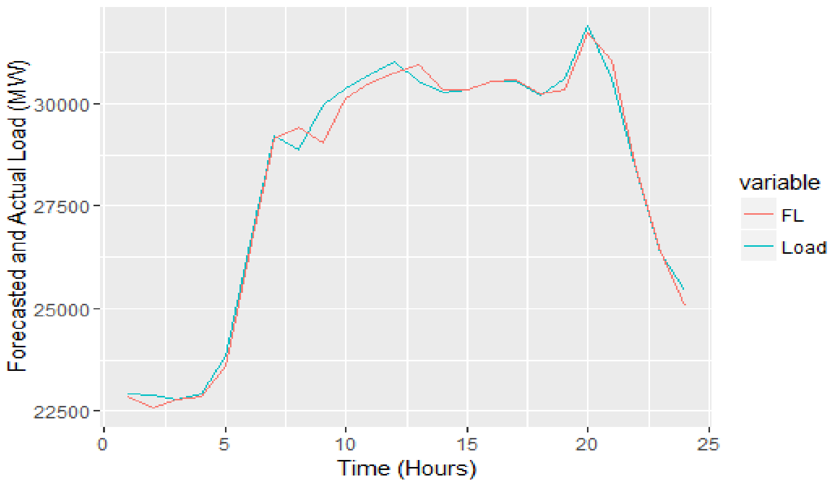

Figure 5.26 Actual Load and FL on Monday 8th March 2010

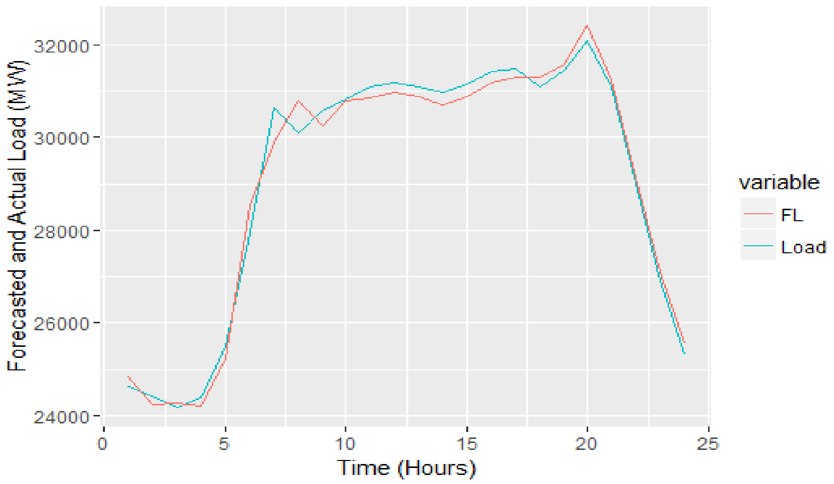

Figure 5.27 Actual Load and FL on Wednesday 10th March 2010

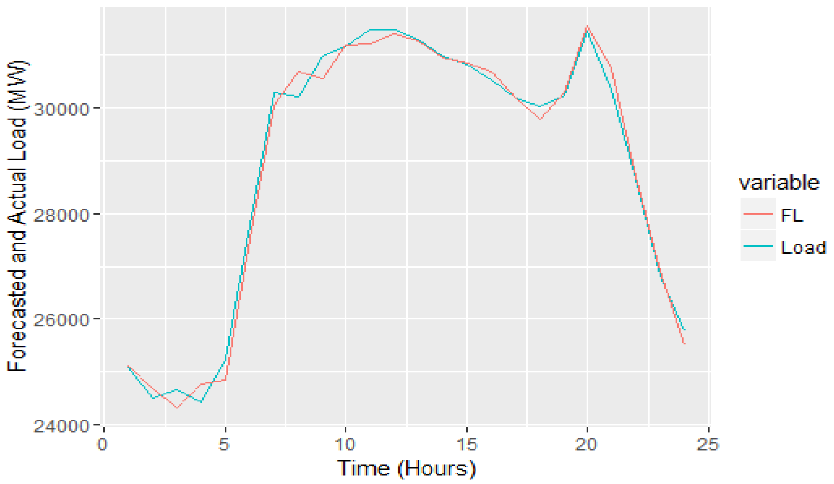

Figure 5.28 Actual Load and FL on Friday 12th March 2010

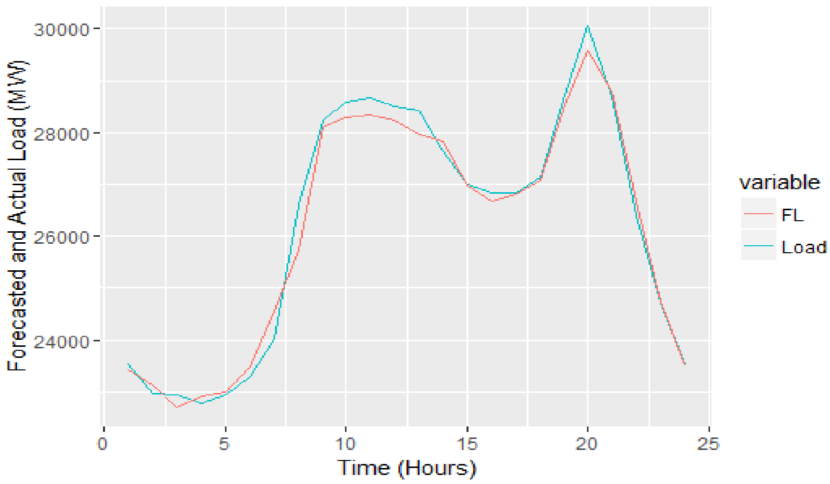

Figure 5.29 Actual Load and FL on Sunday 14th March 2010

In this last case study in March 2010, Table B4 shows that there are a very few higher APE values on average and Figures 5.26 through 5.29 show that the forecast profile curve follows smoothly the actual load profile shape testifying that the forecasting model is performing very well.

Figure 5.30 below gives another way to compare the shape of these two curves, the forecasted and actual load, on a period of seven days in March 2010.

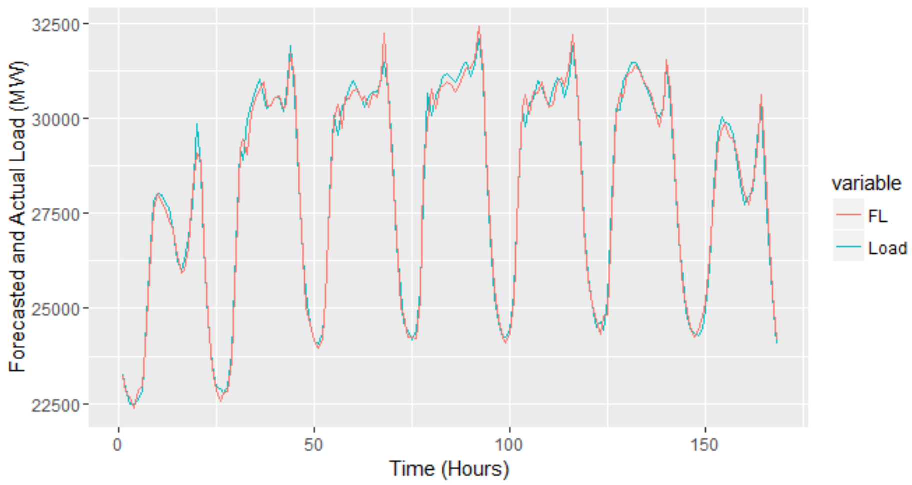

Figure 5.30 Actual Load and FL for 7th – 13th March 2010

On the global point of view Figure 5.30 above gives also a satisfactory picture of a forecasting model that performed well as the gap between the two curves is relatively small.

The forecasted and actual load curves were compared once again in this last case study, and results showed that the MAPE is 0.77%, the RMSE is 295.867 MW, the Daily Peak Error ranged between 0.07% and 1.34% for March 2010. The recorded MAPE in this case is equal to the one in the first case, and the Daily Peak Error is even more accurate than what was recorded in case I. It is a very good performance regarding to the results reported in the literature.

Table B5 Appendix B gives the average error on a daily basis in terms of the MAPE, MAE and the Daily Peak Error of randomly chosen days in August 2009 and October 2010.

Besides the four analysed scenarios above, we then took a look at the MLP model performance during the FIFA World cup (11th June – 11th July 2010) so as to obtain more insight of its robustness.

The set of graphs below, from Figures 5.31 to 5.41, and Table B6 are dedicated to a period of time when took place the FIFA World Cup in South Africa, from June 11th to July 11th 2010.

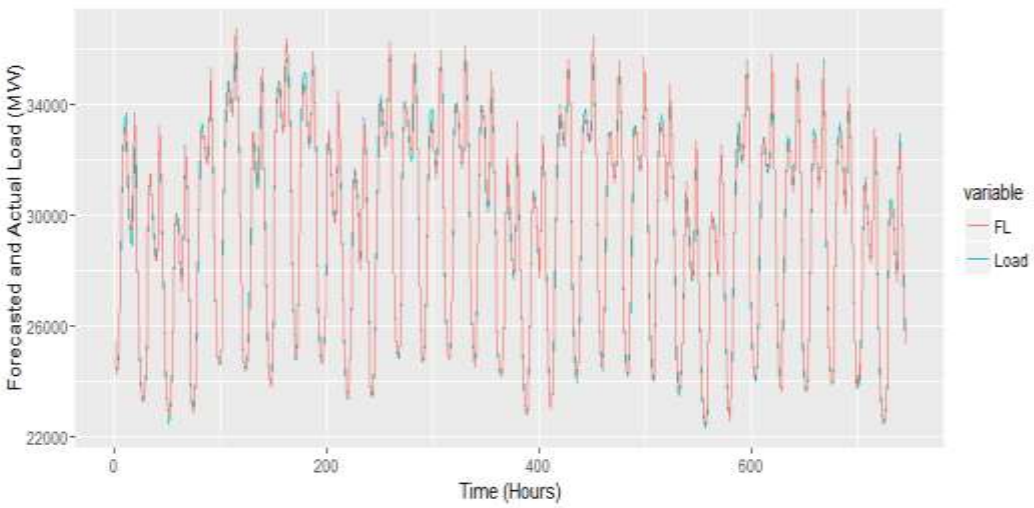

Figure 5.31 Actual Load and FL from 11th June to 11th July 2010

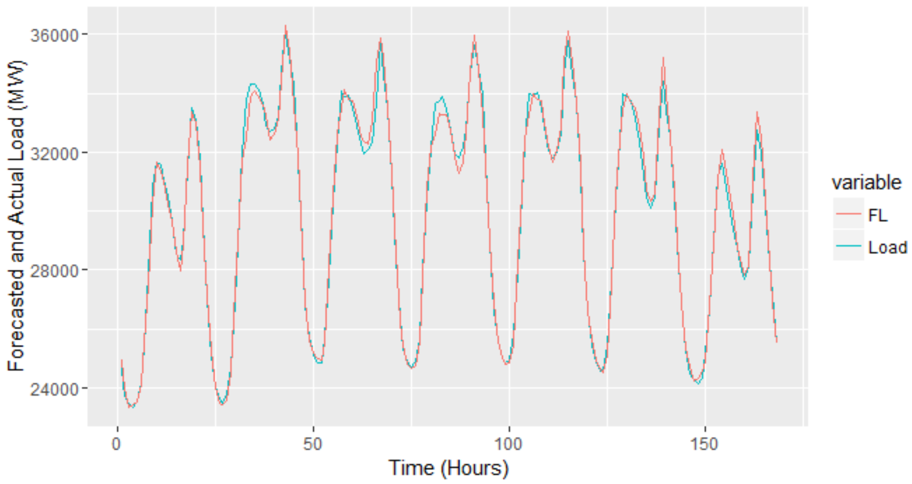

Figure 5.32 Actual Load and FL for 20th – 26th June 2010

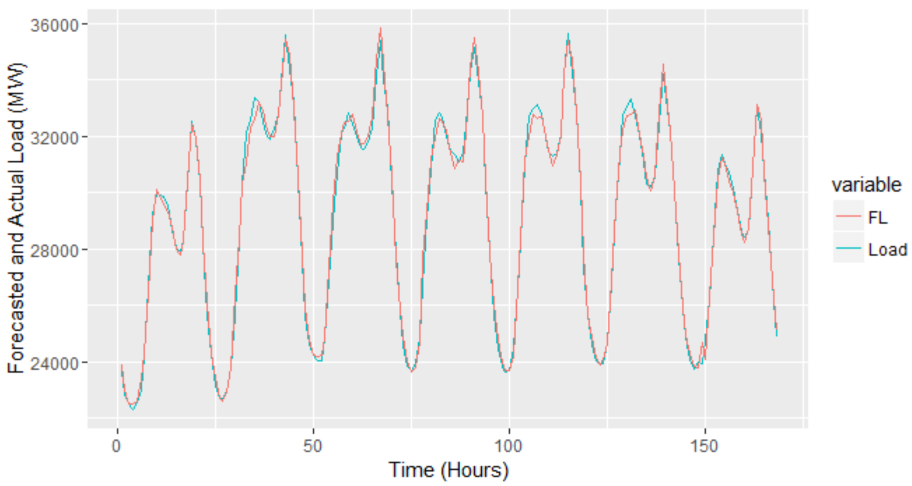

Figure 5.33 Actual Load and FL for 4th – 10th July 2010

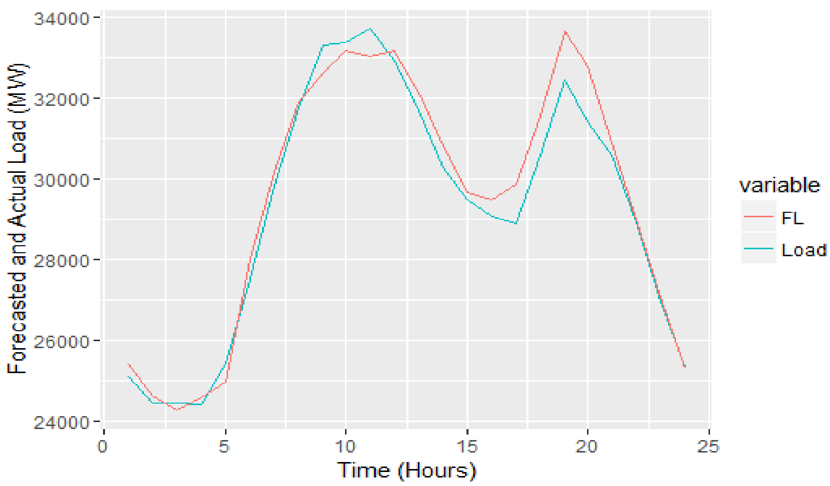

Figure 5.34 Actual Load and FL on Friday 11th June 2010

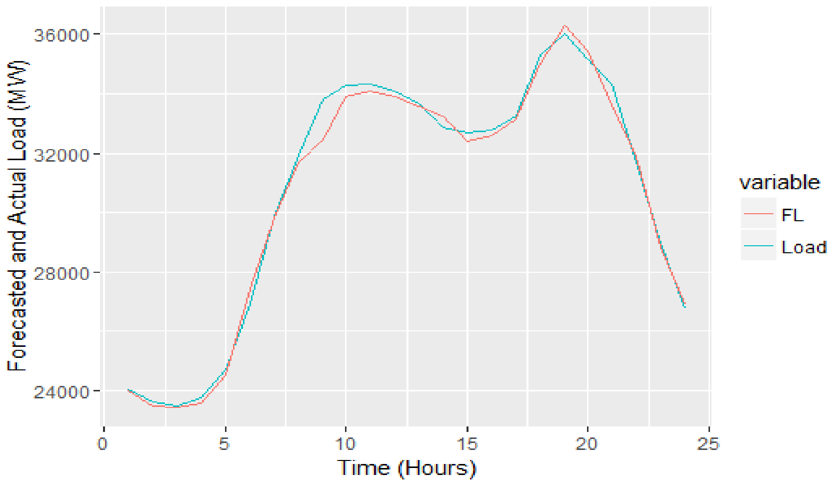

Figure 5.35 Actual Load and FL on Monday 21st June 2010

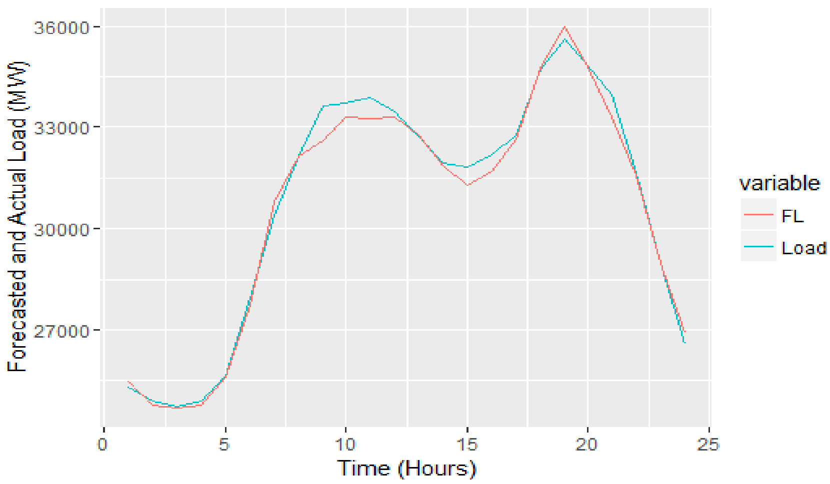

Figure 5.36 Actual Load and FL on Wednesday 23rd June 2010

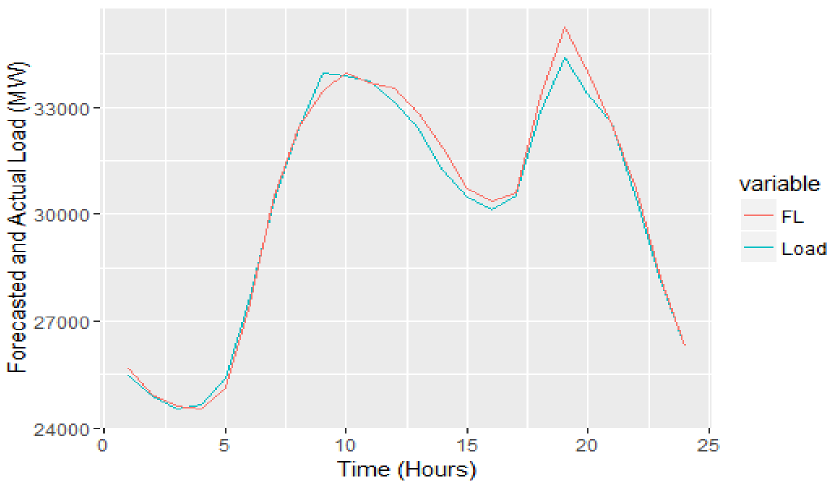

Figure 5.37 Actual Load and FL on Friday 25th June 2010

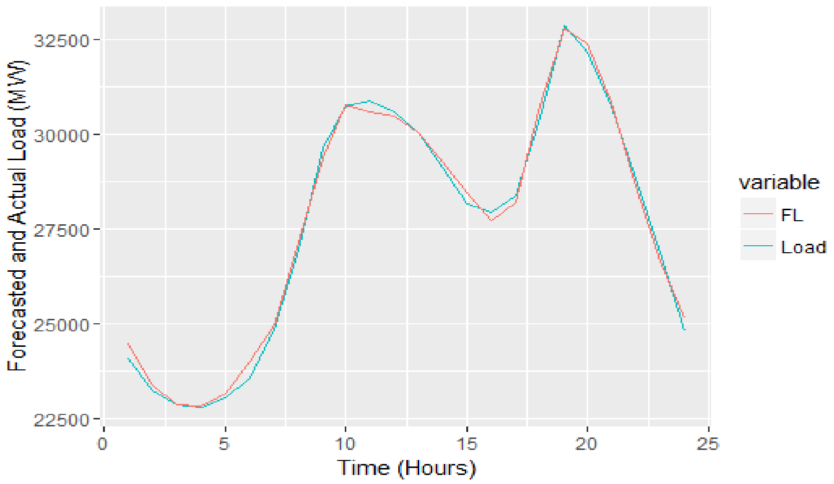

Figure 5.38 Actual Load and FL on Sunday 27th June 2010

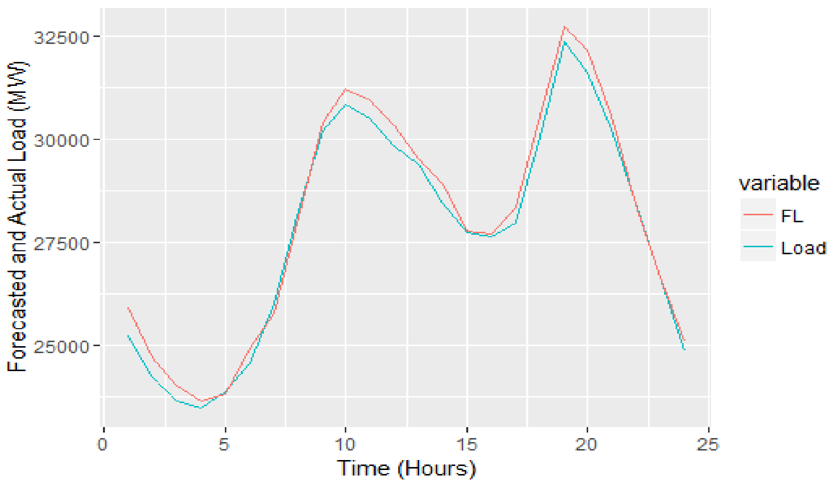

Figure 5.39 Actual Load and FL on Saturday 3rd July 2010

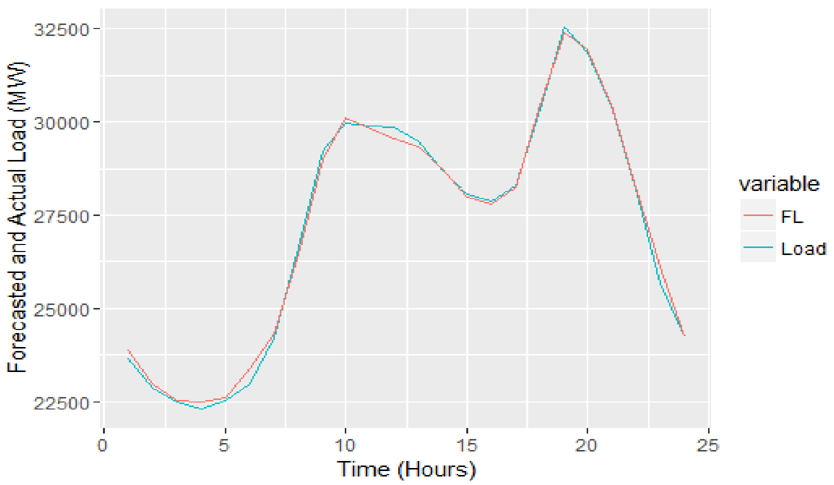

Figure 5.40 Actual Load and FL on Sunday 4th July 2010

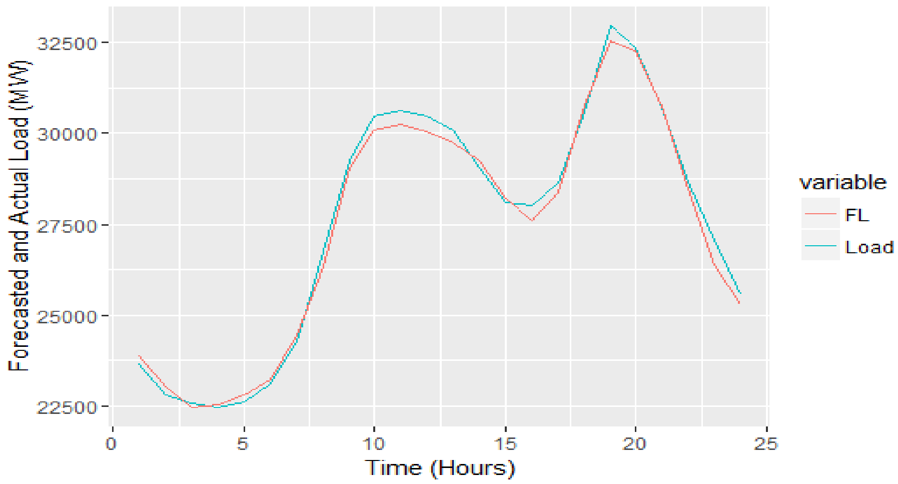

Figure 5.41 Actual Load and FL on Sunday 11th July 2010

Daily average errors for four weeks during the FIFA World Cup 2010 are given in Table B6.

From Figure 5.31, we expected to see some uptrend in the load consumption due to the World Cup, but nothing more than those daily and weekly seasonal patterns that the MLP forecasting model could easily track. Figures 5.32 and 5.33 zoom in the picture in depicting two different weeks during this football festive time. Figure 5.32 for a week in June and Figure 5.33 a week in July. From these graphs, we can deduce that the MLP model, in general, could easily trace the shape of the actual load profile curve despite a few higher Daily Peak error values here and there. In Figure 5.34 we can see that the gap between the two curves is noticeable, especially around the peak hours. The results in Table B6 for this particular day can confirm this fact. Not surprising that the forecasted loads on this day recorded a higher MAPE and a higher MAE values because this day was the game opening day. There was a high level of activities and high energy demand that the forecasting model could not track.

Figures 5.35 through 5.41 show that the load forecasts curve can smoothly follow the path trajectory of the consumption load profile and the error metrics are all small on average, except some few Daily Peak errors, which are high values. It can also be noticed that the forecasting is more accurate (with a MAPE less than 1% on average) during this period of time because it is winter in South Africa and the load consumption is less unpredictable in general.

In a nutshell, the implemented forecasting model performed reliably and with a satisfactory accuracy. Table B5 gives some daily error metrics corroborating this performance with a daily MAPE ranging from 0.50% to 0.90%, which is less than the 3% recommended in the literature by Khotanzad et al., (1997), the Daily Peak Error is between 0.01% and 3%, and the MAE is between 133.12 and 314.35 MW.

5.2.2 Comparison between MLP Model and SARIMAX Model

Hippert et al. (2001) discussed some guidelines to evaluate the "effectiveness of validation" of NN models. They strongly discouraged to evaluate a model by only looking at the goodness of fit statistics and examining in-sample errors instead of the out-of-sample errors, i.e., samples other than those used to fit the model during the training phase. These authors strongly supported the idea that the proposed technique should be compared to some benchmark models, such as ARIMAX or regression models but not to another NN model or to some fuzzy engine because they believed that these are not yet considered as standard or well accepted methods.

So, to get a good idea of our model, a seasonal ARIMAX (SARIMAX) model was trained and run on the same training and testing datasets as in the MLP case. This SARIMAX model was briefly discussed in chapter 3. We included the temperature variable as an exogenous variable in the ARIMA model, hence we have a seasonal ARIMAX that can be written as follows.

$$(1-\phi_1 L-\phi_2 L^2)Z_t \;=\; (1-\theta_1 L-\theta_2 L^2)(1-\theta_{24}L^{24})(1-\theta_{168}L^{168})a_t + \; (1-\theta_1 L-\theta_2 L^2)v_t. \tag{5.1}$$

Equation (5.1) can be rewritten as follows.

$$Y_t = (1-L)(1-L^{24})(1-L^{168})z_t = y_t - y_{t-1} - y_{t-24} + y_{t-25} - y_{t-168} + y_{t-169} - y_{t-192} + y_{t-193}\,, \tag{5.2}$$

$$v_t = (1-L)(1-L^{24})(1-L^{168})x_t = x_t - x_{t-1} - x_{t-24} + x_{t-25} - x_{t-168} + x_{t-169} - x_{t-192} + x_{t-193}\,, \tag{5.3}$$

where y_t is the load at hour *t*, v_t is the temperature at the corresponding hour *t* and L is the lag operator. The free parameters $\phi_1, \phi_2, \theta_1, \theta_2, \theta_{24}, \theta_{168}$ were estimated from the model.

We used the "auto.arima" function included in the 'forecast' R-package via step wise algorithm suggested by Hyndman and Khandakar (2008) to construct automatically the seasonal ARIMA model and forecast the load 24 hours at a time for exactly the same days used in testing the MLP model as to obtain comparable results. The pre-processing of the data was carried out as suggested by Yang et al. (2013).

The MAPE, MAE, MSE and Daily Peak error for both SARIMAX and MLP model on some randomly chosen days are presented in Table 5.1 below.

Table 5.1 SARIMAX and NN Model Average Errors

Date	Day of the Week	MAPE (%)		Daily Peak Error (%)		MAE (MW)		MSE (MW^2)	
		NN	SARIM	NN	SARIM	NN	SARIM	NN	SARIM
01/08/2009	7	0.95	1.82	3.00	5.08	276.08	532.06	140832.173	465990.173
02/08/2009	1	1.15	2.55	2.82	7.66	314.36	732.56	144436.50	866825.391
03/08/2009	2	0.88	3.16	1.07	9.82	263.84	939.73	109514.401	1551814.833
04/08/2009	3	0.64	2.81	1.87	9.86	188.47	850.99	60242.006	1368772.003
05/08/2009	4	0.72	2.82	0.08	10.28	208.03	845.20	64915.675	1360569.797
06/08/2009	5	0.85	2.84	0.42	9.63	245.61	839.55	87455.709	1252119.395
07/08/2009	6	0.67	2.50	0.96	8.61	197.96	733.43	76037.02	1018168.534
11/10/2009	1	0.78	1.99	0.85	7.65	204.04	529.90	56662.554	523753.067
12/10/2009	2	0.98	2.32	1.10	8.59	282.44	646.04	127902.083	729734.430
13/10/2009	3	1.05	2.21	1.85	7.55	289.75	621.53	146581.887	810203.271
14/10/2009	4	0.67	2.33	0.15	8.69	192.82	657.70	62457.346	821348.247
15/10/2009	5	0.59	2.17	0.25	7.89	170.33	620.95	51144.378	745283.478
16/10/2009	6	0.69	1.76	1.24	7.19	200.14	500.87	68617.493	484193.128
17/10/2009	7	0.81	1.72	1.37	6.92	220.49	473.35	81948.284	447276.119
06/12/2009	1	0.57	1.65	0.68	5.34	148.63	441.70	36702.882	350357.747
07/12/2009	2	0.86	1.98	0.43	5.87	248.27	557.36	116679.053	555109.068
08/12/2009	3	0.71	1.70	0.87	5.92	201.50	478.45	65451.678	425979.794
09/12/2009	4	0.83	1.58	1.24	5.05	238.51	445.15	108362.873	324780.045
10/12/2009	5	0.75	1.65	0.36	5.86	211.61	451.19	78145.906	382109.605
11/12/2009	6	0.93	1.61	1.34	4.17	256.40	445.75	127082.888	352222.468
12/12/2009	7	0.80	1.93	0.07	6.78	213.81	525.44	124277.167	499913.070
07/03/2010	1	0.68	2.09	0.42	10.10	179.02	558.57	57888.614	691767.600
08/03/2010	2	0.74	2.41	2.66*	8.41	211.07	683.72	87603.624	948044.574
09/03/2010	3	0.76	1.98	0.49	7.60	221.32	565.60	96874.632	666602.500
10/03/2010	4	0.89	2.00	2.45*	9.05	259.45	581.76	97867.046	735472.646
11/03/2010	5	0.64	2.12	1.02	7.40	187.36	614.83	65954.955	707314.314
12/03/2010	6	0.67	2.08	0.96	6.93	188.11	599.93	57544.278	723023.773

13/03/2010	7	0.84	1.58	0.18	7.67	231.25	442.44	66777.556	448911.616
20/06/2010	1	0.64	2.41	0.01	6.31	179.22	709.88	51347.561	823418.186
21/06/2010	2	0.85	2.11	0.46	6.12	270.99	648.74	146121.946	697030.514
22/06/2010	3	0.92	2.11	0.68	8.42	290.51	655.34	163716.149	883247.716
23/06/2010	4	0.81	2.01	0.17	6.20	257.26	618.93	132997.467	627686.571
24/06/2010	5	0.63	2.15	0.93	6.01	200.55	664.18	68937.790	737511.662
25/06/2010	6	0.77	2.11	0.89	6.67	241.93	641.75	107103.355	808137.416
26/06/2010	7	0.86	2.12	2.42*	7.46	251.05	619.27	97036.056	761234.503
04/07/2010	1	0.50	2.39	1.07	7.56	133.12	672.85	31074.404	773798.453
05/07/2010	2	0.84	2.09	0.36	6.10	254.50	628.41	139255.083	681638.675
06/07/2010	3	0.87	1.90	0.45	7.06	266.17	571.82	107929.055	702226.793
07/07/2010	4	0.93	2.02	1.18	5.54	272.94	605.78	11668.996	560770.396
08/07/2010	5	0.64	2.29	1.02	7.44	190.72	683.32	70841.625	798688.717
09/07/2010	6	0.60	2.03	0.78	7.12	184.07	607.62	58304.925	690398.443
10/07/2010	7	0.71	2.28	1.01	8.59	190.87	659.86	71370.948	799242.195

Table 5.1 above displays the outputs of a comparison between the MLP and SARIMAX models in terms of different error metrics.

Figures 5.42 and 5.43 below give the outputs of the performance of the SARIMAX model run on the same training and testing datasets used to build the MLP model.

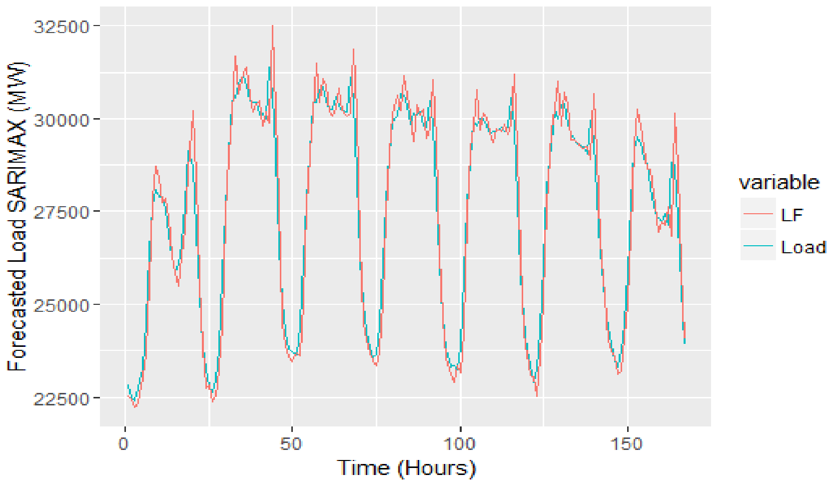

Figure 5.42 Actual Load and FL with SARIMAX, 6th – 12th Dec 2009

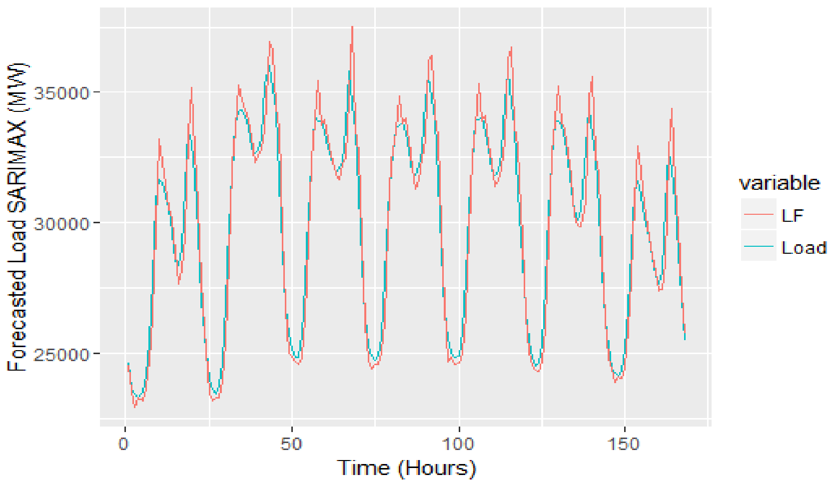

Figure 5.43 Actual Load and FL with SARIMAX, 20th – 26th June 2010

Comparison of the performance of the MLP and the SARIMAX models in terms of the APE is displayed on Figures 5.44 and 5.45 below, on arbitrarily chosen days in August 2009 and June 2010, respectively.

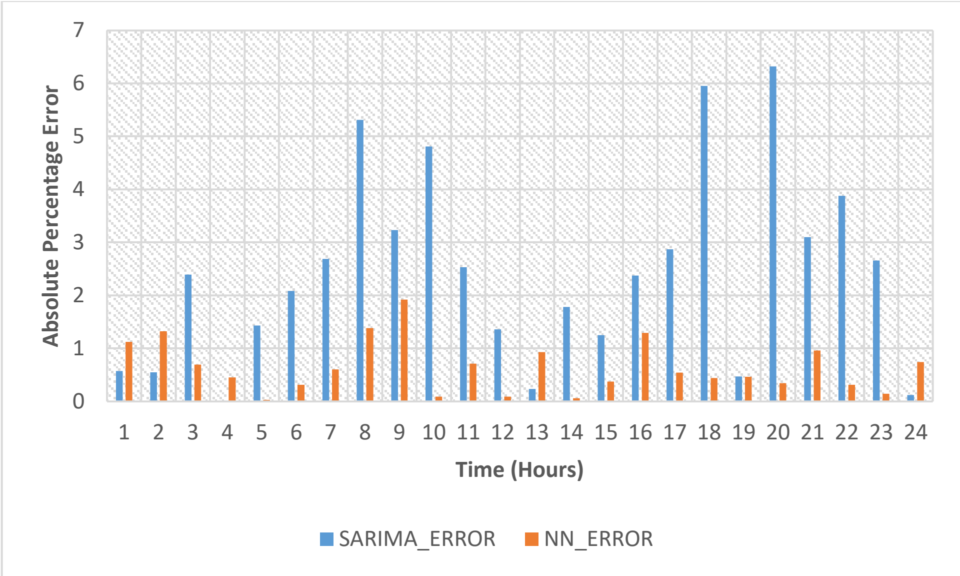

Figure 5.44 APE of SARIMAX and MLP model on 1st August 2009

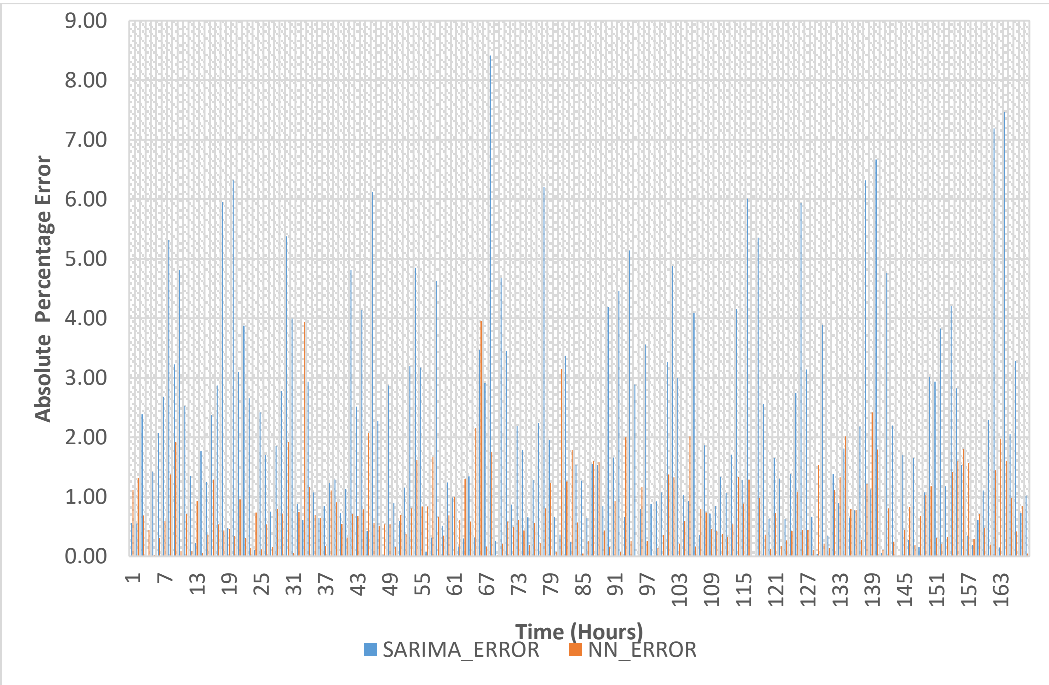

Figure 5.45 APE of SARIMAX and NN model during 20th – 26th June 2010

It can be seen in Table 5.1 (comparison table) and in Figures 5.42 through 5.45 that the SARIMAX model presented much larger errors than the MLP model. Figures 5.42 and 5.43 showed two different weeks of forecasts with SARIMAX, which was not too bad as to judge on these graphs, but not as good as the MLP as show the respective MAPEs below. In Figure 5.44 it can be clearly seen that the MLP model is superior to the SARIMAX with a MAPE of 0.50% and 1.90% respectively

The MLP model presented in this research report performed better at forecasting recursively hourly load 24 hours ahead as shown in the results for all the different error metrics used in this work to evaluate the performance of LF models. Through the four case studies we tested and validated the proposed model, but to get its overall performance error, we ran the MLP model through all the entire training and testing datasets at once and obtained its overall MAPE of 0.50% and MSE of 5.32e+08 as can be seen in Figure 5.1. It should not be a surprise that the built NN model improved its results because it is well known in the literature and in particular as emphasised Park et al. (1991) that the NN technique performs very well when the training data is widely spread in the feature space. These results compared favourably to most of those reported in the STLF literature we consulted and the 3% of the MAPE recommended by Khotanzard et al. (1997). Besides the built NN model accuracy, it has also proven to be robust and adjustable to changing conditions as proved by its performance during the 2010 FIFA World Cup period. In fact, this performance is a welcome surprise as the LF curve could follow and track properly the shape of the real load curve without any major discrepancy as can be seen in Figures 5.31 through 5.41.

However, the built MLP model was inferior to the NN-based model constructed by Reddy and Momoh (2014), which achieved an impressive accuracy of 0.004% in terms of MAPE, and Al-Subhi and Ahmad (2015) with the MAPE ranging between 0.35% and 0.49%. Nevertheless, we strongly believe that introducing other weather variables such as wind speed, cloud coverage, humidity, and rainfall will produce even better results in terms of accuracy for the MLP model built in this research report.

5.3 Summary

In the first part of this chapter, we produced the distribution of the data, analysed the Eskom load profile and highlighted its characteristics. Next, we displayed the Matlab NN toolbox to give an idea of what was going on during the training phase of the network according to how we structured and designed it. We also presented a couple of different plots giving some measures of performance, goodness of fit of the data to the MLP built model, and a validation check. More importantly, in the second part of the chapter, we presented and discussed the results of four different case studies to investigate our MLP built model. In the last part of the chapter, we tested a SARIMAX model and compared it to our MLP model so as to have an opinion on the validity of the latter built model.

CHAPTER 6

SUMMARY - CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary

We used the NN technique in the STLF area and built an MLP model using eleven years (2000 - 2010) load data and corresponding temperature from Eskom. We investigated the MLP built model through four different case studies and presented the results that showed a satisfactory performance and achieved a sensible prediction accuracy. The forecasting accuracy was evaluated by calculating different error metrics such as the MAPE, the MAE, the APE, the Daily Peak error, and the MSE or the derived RMSE. The range of some error metrics lies between 11 668.996 MW^2 and 163 716.149 MW^2 for the MSE, 0.01% and 3.58% of the Daily Peak error, and 0.50% and 0.90% for the MAPE. These ranges of error are very consistent with the results obtained in the literature on STLF. To have an opinion of our MLP model, we compared it to a benchmark SARIMAX model and the results showed that the MLP model performed better. In addition to accuracy, the model has proven to be robust and adjustable to changing conditions as proved its performance during the 2010 FIFA World Cup. Moreover, the model proved to be also reliable as it could forecast reasonably the first day of this big event with a lot of unpredictable activities. As an hourly model, the built MLP model can be up to date as it performs dynamically by using new available data as they come in.

6.2 Conclusions

STLF can assist electric power management decision makers to operate and secure their power system efficiently and economically 24 hours ahead. The electric power management is very challenging in South Africa. There is a need for accurate techniques to support power system managers in the performance of their daily duties. The LF field has plenty of methods to build accurate models for such purposes but the NN technique has a couple of advantages that we outlined in this research work that justified their preference.

We applied the NN technique to STLF and built an MLP model that produced an overall MAPE of 0.50% error and MSE performance of 5.32e+08 that compared favourably to most of the error rates reported in the STLF literature we consulted or the utmost 3% of the MAPE expected in general. Indeed, these results are highly accurate and further reinforces the capability of NN models in forecasting the electric load. Therefore, the model can assist decision makers in the EMS with needed information to perform their workday more efficiently and economically by minimising the operating costs, planning routine maintenance, preventing the overloading and reducing the occurrence of equipment failures.

However, we have to admit that the NN technique is not a panacea in STLF area, since it has some limitations or drawbacks when it comes to interpretation of the models and a high risk of overfitting of models if not carefully designed. Still, the MLP model we built in this research report demonstrated a lot of interesting properties reported in the literature. Therefore, it is a very suitable MLP model that can assist decision makers in the EMS with the necessary information in their daily operations.

6.3 Recommendations

Since the MLP model was built based on only one site (Eskom), more evidence on other electric utilities is required so as to ensure its portability. Otherwise, the model must be re-trained every time it is run on a new site. We also believe that introducing other weather variables such as wind speed, cloud coverage, and humidity would yield better results for our MLP model.

Given that the built MLP model performance was inferior to some of the models in the literature, a hybrid model, SARIMA-MLP, with SARIMA handling the linearity of the load series and the MLP dealing with nonlinearity of the load, will yield better results.

Another point to be taken into account is the reliability of the MLP models. The way NNs perform the forecasts is rather complicated and very difficult to understand, that is why they call them “black box”, and therefore some abnormal behaviour may unexpectedly occur in unusual conditions. This is one of the drawbacks of how the NN models operate. Hence, a detailed online testing is recommended so as to insure NN models reliability in different situations.

In building the MLP model, we took into account the load of special days such as public holidays that we treated as Sundays. This is a simplistic solution to handle public holidays, but in order to obtain more accurate results, some more appropriate techniques should be considered for future refinements of the built MLP model.

Appendix A Matlab® Code

Import Weather & Load Data

The data set used is a table of historical hourly loads and temperature observations from Eskom for the years 2000 to 2010. The weather information includes the dry bulb temperature only.

The dataset is imported from an excel file using an auto-generated function *fetchDBLoadData*.

```
data = fetchDBLoadData('2000-01-01', '2010-08-31');
```

Import list of holidays

A list of South African public holidays that span the historical date range is imported from an Excel spreadsheet.

```
[num, text] = xlsread('..\Data\Holidays.xls');
holidays = text(2:end,1);
```

Generate Predictor Matrix

The function **genPredictors** generates the predictor variables used as inputs for the model.

```
% Select forecast horizon
term = 'short';

[X, dates, labels] = genPredictors(data, term, holidays);
```

Split the dataset (cross-validation)

```
% Create training set
net.divideParam.trainRatio = 70/100
trainInd = net.divideParam.trainRatio;
trainX = X(trainInd,:);
trainY = data.SYSLoad(trainInd);

% Create validation set
net.divideParam.valRatio = 15/100;
valInd = net.divideParam.valRatio
valX = X(valInd, :);
valY = data.SYSLoad(valInd);

% Create test set and save for later
net.divideParam.testRatio = 15/100;
testInd = net.divideParam.testRatio;
```

```
testX = X(testInd,:);
testY = data.SYSLoad(testInd);
testDates = dates(testInd);

save Data\testSet testDates testX testY
clear X data trainInd testInd term holidays dates ans num text
```

Build the Load Forecasting Model

```
reTrain = false;
if reTrain || ~exist('Models\NNModel.mat', 'file')
    net = newfit(trainX', trainY', 25);
    net.performFcn = 'mae';
    net = train(net, trainX', trainY');
    save Models\NNModel.mat net
else
    load Models\NNModel.mat
end
```

```
load Data\testSet
forecastLoad = sim(net, testX')';
```

```
err = testY-forecastLoad;
fitPlot(testDates, [testY forecastLoad], err);

errpct = abs(err)./testY*100;

fL = reshape(forecastLoad, 24, length(forecastLoad)/24)';
tY = reshape(testY, 24, length(testY)/24)';
peakerrpct = abs(max(tY,[],2) - max(fL,[],2))./max(tY,[],2) * 100;

MAE = mean(abs(err));
MAPE = mean(errpct(~isinf(errpct)));

fprintf('Mean Absolute Percent Error (MAPE): %0.2f%% \nMean Absolute Error
(MAE): %0.2f MWh\nDaily Peak MAPE: %0.2f%%\n',...
    MAPE, MAE, mean(peakerrpct))
```

Appendix B Tables of Error Different Metrics

Table B1 Hourly Actual Load, FL and APE

	02/08/2009			03/08/2009			05/08/2009			07/08/2009		
Hour	Load	FL	APE	Load	FL	APE	Load	FL	APE	Load	FL	APE
1	23704	24046.50	1.44	22960	23105.78	0.63	24437	24336.00	0.41	24215	24135.45	0.33
2	22928	23150.84	0.97	22632	22476.73	0.69	23852	23933.78	0.34	23835	23717.39	0.49
3	22565	22666.26	0.45	22513	22512.89	0.00	23599	23631.81	0.14	23671	23623.43	0.20
4	22527	22555.72	0.13	22842	22675.30	0.73	23978	23657.65	1.34	23873	23761.27	0.47
5	22447	22975.92	2.36*	23868	23689.92	0.75	25132	24777.60	1.41	24869	24590.40	1.12
6	22880	22539.91	1.49	27041	26812.67	0.84	28068	28455.04	1.38	27857	27767.84	0.32
7	23929	24171.91	1.02	31436	30852.81	1.86	31969	31737.02	0.73	31537	31665.05	0.41
8	26824	25867.07	3.57*	31825	31846.20	0.07	31816	32349.32	1.68	31746	32026.74	0.88
9	29573	29452.55	0.41	32010	32104.24	0.29	31609	31751.09	0.45	31784	31827.81	0.14
10	31118	30837.71	0.90	31776	31897.79	0.38	30992	31462.22	1.52	31511	31692.13	0.57
11	31274	31059.88	0.68	32034	31536.06	1.55	31155	30843.77	1.00	31646	31324.55	1.02
12	31037	30751.79	0.92	31710	31695.42	0.05	30894	31083.33	0.61	31159	31362.60	0.65
13	30639	30357.92	0.92	30891	31128.73	0.77	30301	30526.10	0.74	30256	30588.69	1.10
14	29957	29939.38	0.06	30397	30081.48	1.04	29730	29687.73	0.14	29414	29434.54	0.07
15	28863	29588.37	2.51*	30042	30174.81	0.44	29689	29487.35	0.68	29045	28990.07	0.19
16	28495	28623.97	0.45	30372	30128.97	0.80	29923	29932.45	0.03	28720	29025.67	1.06
17	28684	29083.37	1.39	30571	31026.81	1.49	30287	30267.04	0.07	28853	28752.50	0.35
18	30426	30665.44	0.79	32248	32222.04	0.08	31518	31879.70	1.15	30112	30088.08	0.08
19	32890	32560.46	1.00	34733	34085.18	1.87	33898	34041.31	0.42	31986	32850.76	2.70*
20	32912	32372.67	1.64	34517	34044.61	1.37	33389	33512.75	0.37	31597	32146.11	1.74
21	31181	31522.21	1.09	33529	33002.73	1.57	32217	32216.20	0.00	30730	30754.13	0.08
22	28706	28805.17	0.35	30858	31025.04	0.54	29718	30071.08	1.19	28679	28959.20	0.98
23	25834	26345.58	1.98	27183	27812.87	2.32*	27157	27308.74	0.56	26704	26728.49	0.09
24	24002	24269.67	1.12	25169	25439.72	1.08	25232	25436.35	0.81	24892	25179.12	1.15

*higher error value

Table B2 Actual Load, FL and APE for October 2009

	04/10/2009			05/10/2009			07/10/2009			09/10/2009		
Hour	Load	FL	APE	Load	FL	APE	Load	FL	APE	Load	FL	APE
1	22689	22790.48	0.45	22292	22364.98	0.33	24172	24178.15	0.03	24525	24222.53	1.23
2	22197	22210.58	0.06	22041	21916.02	0.57	23735	23788.28	0.22	24119	24075.51	0.18
3	21716	21931.00	0.99	22022	21969.35	0.24	23734	23560.07	0.73	23882	23920.94	0.16
4	21676	21589.04	0.40	22238	22201.28	0.17	23855	23831.08	0.10	24246	23931.95	1.30
5	21882	21957.28	0.34	23152	22984.83	0.72	24681	24571.28	0.44	25129	25063.65	0.26
6	22290	22573.24	1.27	26531	25805.22	2.74*	27864	27133.63	2.62*	27963	27558.72	1.45
7	23601	23532.04	0.29	29269	29979.55	2.43*	30131	30134.16	0.01	30364	30245.83	0.39
8	25801	25693.98	0.41	29308	29712.60	1.38	29791	30261.37	1.58	30404	30455.72	0.17
9	27617	27490.53	0.46	29805	29488.96	1.06	30202	30111.93	0.30	31136	30526.27	1.96
10	28628	27966.82	2.31*	29864	29795.01	0.23	29938	30557.96	2.07*	30896	31129.94	0.76
11	28759	28301.18	1.59	30280	29738.44	1.79	30451	30131.48	1.05	31183	30902.42	0.90
12	28629	28224.55	1.41	30388	30084.53	1.00	30306	30540.04	0.77	31235	31067.62	0.54

13	28034	27937.66	0.34	29825	30064.89	0.80	30196	30130.51	0.22	30858	30945.36	0.28
14	27218	27122.18	0.35	29535	29330.78	0.69	29647	29939.01	0.98	30034	30394.56	1.20
15	26486	26364.10	0.46	29595	29559.52	0.12	29875	29491.15	1.28	30181	29691.95	1.62
16	26037	26079.63	0.16	29995	29828.15	0.56	30261	30055.22	0.68	29950	30214.93	0.88
17	26383	26019.45	1.38	30306	30354.09	0.16	30401	30306.20	0.31	30006	29693.86	1.04
18	27319	27470.94	0.56	30734	30857.25	0.40	30832	30445.85	1.25	29893	30025.00	0.44
19	29416	29424.73	0.03	31817	32088.82	0.85	31616	32117.83	1.59	30789	30990.91	0.66
20	29940	29242.44	2.33*	32138	31844.56	0.91	32113	32243.55	0.41	31165	31726.13	1.80
21	28530	28438.68	0.32	30889	30677.56	0.68	30822	30917.95	0.31	29704	30140.53	1.47
22	26263	26383.45	0.46	28610	28424.24	0.65	28751	28724.10	0.09	28135	28091.51	0.15
23	24282	24470.08	0.77	25967	26319.49	1.36	26485	26723.75	0.90	26478	26492.88	0.06
24	23059	23068.51	0.04	24399	24527.51	0.53	25185	25095.99	0.35	25205	25183.06	0.09

*higher error value

Table B3 Hourly Actual Load, FL and APE of December 2009

	06/12/2009			07/12/2009			09/12/2009			11/12/2009		
Hour	Load	FL	APE	Load	FL	APE	Load	FL	APE	Load	FL	APE
1	23252	23414.40	0.70	23278	23146.95	0.56	24279	24369.19	0.37	23768	23992.16	0.94
2	22812	22741.53	0.31	22773	22933.44	0.70	23850	23933.80	0.35	23367	23430.46	0.27
3	22479	22538.12	0.26	22641	22643.55	0.01	23570	23708.56	0.59	22900	23243.71	1.50
4	22389	22385.30	0.02	22972	22746.41	0.98	23655	23643.22	0.05	23269	22945.78	1.39
5	22744	22621.56	0.54	23783	23750.82	0.14	24450	24331.84	0.48	24274	24163.08	0.46
6	23265	23709.72	1.91	25765	26049.73	1.11	26074	26683.75	2.34*	26138	27140.35	3.83*
7	24543	24585.39	0.17	27585	27779.29	0.70	27745	28051.46	1.10	28104	28327.71	0.80
8	26383	26473.66	0.34	28968	28576.27	1.35	29048	29078.73	0.11	29308	29240.32	0.23
9	27677	27703.58	0.10	30479	29364.36	3.66*	29926	29598.02	1.10	30245	29792.90	1.49
10	28085	27880.18	0.73	30575	30734.60	0.52	30080	30093.05	0.04	29992	30353.74	1.21
11	27895	27958.57	0.23	30948	30763.45	0.60	30684	29998.32	2.23*	30383	30098.64	0.94
12	27873	27634.90	0.85	31199	31005.83	0.62	30648	30383.10	0.86	30390	30369.77	0.07
13	27563	27508.22	0.20	30888	31117.19	0.74	30366	30248.95	0.39	29859	30207.12	1.17
14	26873	27001.12	0.48	30494	30667.33	0.57	29836	29850.06	0.05	29624	29439.05	0.62
15	26318	26236.08	0.31	30446	30438.86	0.02	30133	29517.47	2.04*	29467	29537.10	0.24
16	25858	26108.64	0.97	30459	30441.74	0.06	30115	30141.84	0.09	29335	29500.84	0.57
17	26102	25806.65	1.13	30090	30342.70	0.84	30192	30064.16	0.42	29267	29321.86	0.19
18	26615	26555.00	0.23	30148	29688.69	1.52	29794	30365.60	1.92	29256	29422.48	0.57
19	27709	27825.76	0.42	30006	30358.85	1.18	29759	30482.52	2.43*	29053	30013.53	3.31*
20	29114	28988.69	0.43	31389	30852.38	1.71	30484	30793.75	1.02	29958	30218.10	0.87
21	28703	28236.72	1.62	30690	30712.14	0.07	29866	29706.28	0.53	29450	29304.07	0.50
22	26994	26756.61	0.88	29085	28630.54	1.56	27960	28157.46	0.71	27944	27848.38	0.34
23	25248	25058.43	0.75	26540	26879.27	1.28	26115	26183.90	0.26	26429	26215.44	0.81
24	23866	23898.01	0.13	25067	25106.67	0.16	24672	24782.84	0.45	25094	25084.08	0.04

*higher error value

Table B4 Hourly Actual Load, FL and APE for March 2010

	08/03/2010			10/03/2010			12/03/2010			14/03/2010		
Hour	Load	FL	APE	Load	FL	APE	Load	FL	APE	Load	FL	APE
1	22931	22855.53	0.33	24625	24835.78	0.86	25077	25116.51	0.16	23540	23431.70	0.46
2	22890	22569.56	1.40	24422	24250.96	0.70	24495	24684.26	0.77	22974	23138.62	0.72
3	22779	22766.81	0.05	24171	24263.84	0.38	24660	24318.26	1.39	22945	22691.55	1.10
4	22930	22836.89	0.41	24384	24204.75	0.74	24428	24781.99	1.45	22776	22913.61	0.60
5	23821	23610.94	0.88	25484	25226.39	1.01	25240	24851.61	1.54	22938	22983.04	0.20
6	26606	26409.32	0.74	27941	28521.61	2.08*	27791	27514.73	0.99	23280	23462.86	0.79
7	29224	29153.89	0.24	30647	29888.35	2.48*	30273	30059.22	0.71	24016	24540.58	2.18*
8	28888	29435.31	1.89	30087	30780.26	2.30*	30206	30668.73	1.53	26602	25724.31	3.30*
9	29969	29059.05	3.04*	30589	30252.67	1.10	30982	30546.83	1.40	28225	28112.16	0.40
10	30363	30129.64	0.77	30810	30798.71	0.04	31156	31188.25	0.10	28597	28302.67	1.03

11	30718	30507.05	0.69	31104	30845.32	0.83	31463	31206.00	0.82	28671	28340.31	1.15
12	31029	30762.16	0.86	31173	30962.48	0.68	31483	31382.28	0.32	28511	28234.57	0.97
13	30554	30944.52	1.28	31079	30886.59	0.62	31276	31262.52	0.04	28431	27974.24	1.61
14	30263	30347.62	0.28	30958	30696.60	0.84	30959	30943.11	0.05	27650	27834.88	0.67
15	30351	30328.14	0.08	31152	30885.59	0.86	30804	30837.97	0.11	26995	26969.03	0.10
16	30550	30552.86	0.01	31414	31179.73	0.75	30527	30668.87	0.46	26830	26681.30	0.55
17	30531	30596.63	0.21	31477	31309.29	0.53	30180	30190.07	0.03	26839	26813.40	0.10
18	30192	30256.06	0.21	31102	31302.38	0.64	30022	29788.01	0.78	27122	27092.70	0.11
19	30620	30348.23	0.89	31444	31583.41	0.44	30243	30286.28	0.14	28718	28544.22	0.61
20	31893	31737.59	0.49	32071	32399.55	1.02	31459	31539.24	0.26	30051	29585.96	1.55
21	30617	31047.15	1.40	31081	31232.99	0.49	30327	30746.72	1.38	28685	28811.39	0.44
22	28360	28394.98	0.12	29030	29127.63	0.34	28638	28709.19	0.25	26414	26675.56	0.99
23	26372	26385.20	0.05	26901	27105.78	0.76	26830	26917.94	0.33	24718	24750.24	0.13
24	25430	25046.85	1.51	25327	25547.98	0.87	25784	25511.71	1.06	23529	23504.89	0.10

*higher error value

Table B5 Daily FL errors

Date	Day of the Week	MAPE (%)	Daily Peak Error (%)	MAE (MW)
01/08/2009	7	0.95	3.00*	276.08
02/08/2009	1	1.15	2.82*	314.36
03/08/2009	2	0.88	1.07	263.84
04/08/2009	3	0.64	1.87	188.47
05/08/2009	4	0.72	0.08	208.03
06/08/2009	5	0.85	0.42	245.612
07/08/2009	6	0.67	0.96	197.96
11/10/2009	1	0.78	0.85	204.04
12/10/2009	2	0.98	1.10	282.44
13/10/2009	3	1.05	1.85	289.75
14/10/2009	4	0.67	0.15	192.82
15/10/2009	5	0.59	0.25	170.33
16/10/2009	6	0.69	1.24	200.14
17/10/2009	7	0.81	1.37	220.49
06/12/2009	1	0.57	0.68	148.63
07/12/2009	2	0.86	0.43	248.27
08/12/2009	3	0.71	0.87	201.50
09/12/2009	4	0.83	1.24	238.51
10/12/2009	5	0.75	0.36	211.61
11/12/2009	6	0.93	1.34	256.40
12/12/2009	7	0.80	0.07	213.81
07/03/2010	1	0.68	0.42	179.02
08/03/2010	2	0.74	2.66*	211.07
09/03/2010	3	0.76	0.49	221.32
10/03/2010	4	0.89	2.45*	259.45
11/03/2010	5	0.64	1.02	187.36
12/03/2010	6	0.67	0.96	188.11
13/03/2010	7	0.84	0.18	231.25
20/06/2010	1	0.64	0.01	179.22
21/06/2010	2	0.85	0.46	270.99
22/06/2010	3	0.92	0.68	290.51
23/06/2010	4	0.81	0.17	257.26

24/06/2010	5	0.63	0.93	200.55
25/06/2010	6	0.77	0.89	241.93
26/06/2010	7	0.86	2.42*	251.05
04/07/2010	1	0.50	1.07	133.12
05/07/2010	2	0.84	0.36	254.50
06/07/2010	3	0.87	0.45	266.17
07/07/2010	4	0.93	1.18	272.94
08/07/2010	5	0.64	1.02	190.72
09/07/2010	6	0.60	0.78	184.07
10/07/2010	7	0.71	1.01	190.87

*higher error value

Table B6 Daily errors from June 11th to July 11th 2010

Date	Day of the Week	MAPE (%)	Daily Peak Error (%)	MAE (MW)
11/06/2010	6	1.47	1.50	441.19
12/06/2010	7	0.67	0.11	187.68
13/06/2010	1	0.77	0.32	206.15
14/06/2010	2	0.97	0.58	295.53
15/06/2010	3	1.08	0.34	349.39
16/06/2010	4	1.13	2.49*	339.35
17/06/2010	5	1.00	0.91	310.64
18/06/2010	6	0.71	2.00*	226.02
19/06/2010	7	0.71	0.79	207.70
20/06/2010	1	0.64	0.01	179.22
21/06/2010	2	0.85	0.46	270.99
22/06/2010	3	0.92	0.68	290.51
23/06/2010	4	0.81	0.17	257.26
24/06/2010	5	0.63	0.93	200.55
25/06/2010	6	0.77	0.89	241.93
26/06/2010	7	0.86	2.42*	251.05
27/06/2010	1	0.69	1.98	186.11
28/06/2010	2	0.96	0.13	291.59
29/06/2010	3	0.90	0.36	281.18
30/06/2010	4	0.72	2.88*	217.31
01/07/2010	5	0.89	0.74	264.39
02/07/2010	6	0.83	0.75	250.13
03/07/2010	7	1.08	1.61	299.51
04/07/2010	1	0.50	1.07	133.12
05/07/2010	2	0.84	0.36	254.50
06/07/2010	3	0.87	0.45	266.17
07/07/2010	4	0.93	1.18	272.94
08/07/2010	5	0.64	1.02	190.72
09/07/2010	6	0.60	0.78	184.07
10/07/2010	7	0.71	1.01	190.87
11/07/2010	1	0.90	0.14	249.47

*higher error value

References

Abdel-Aal, R. E. (2004) Short-Term Hourly Load Forecasting Using Abductive Networks, *IEEE Transaction on Power Systems*, Vol. 19, no 1, pp. 164 – 173.

Alfares, H. K. and Nazeeruddin, M. (2002) Electric Load Forecasting: Literature survey and classification of methods, *International Journal of Systems Science*, Vol. 33, pp. 23-34.

Al-Subhi, A. and Ahmad, C. B. (2015) Short Term Load Forecasting using Artificial Neural Networks for a Residential Area in an Industrial City, *International Journal of Engineering Research and Technology (IJERT)*; Vol. 4, pp. 307-314.

Ameya, D. (2010) *Electricity Load and Price Forecasting Webinar Case Study*, [online] available from: http://www.mathworks.com/matlabcentral/fileexchange/28684-electricity-load-and-price-forecasting-webinar-case-study/content/Electricity%20Load%20&%20Price%20Forecasting/Load/html/LoadScriptNN.html [Accessed: 3rd March 2015].

Amral, N., King, D., and Ozveren C. S. (2008) Application of Artificial Neural Network for Short-Term Load Forecasting, *IEEE 43rd International Universities Power Engineering Conference, UPEC - 2008.* pp. 1-5.

Banda, E. and Folly, K. A. (2007) Short-Term Load Forecasting Using Artificial Neural Network, *IEEE Power Tech. Lausanne*, pp. 108 – 112.

Bagnasco, A., Saviozzi, M., silverstro, F., Vinci, A., Grillo, S., and Zennaro, E. (2014) Artificial Neural Network Application to Load Forecasting in a Large Hospital Facility, *IEEE International Conference on Probabilistic Method Applied to Power System*, pp. 1-6.

Buhari, M. and Adamu, S.S. (2012) Short-Term Load Forecasting Using Artificial Neural Network, *Proceedings of the MultiConference of Engineers and Computer Scientists (IMECS) Vol 1. Hong-Kong*: 14-16 March, pp. 1-4.

Cabrera, N. G. , Guiterrez-Alcaraz, G. and Gil, E. (2013) Load Forecasting Assessment Using SARIMA Model and Fuzzy Inductive Reasoning, *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 561-565.

Charytoniuk, W. and Chen, M.S. (2000) Very Short-Term Load Forecasting Using Artificial Neural Networks, *IEEE Transactions on power Systems*, Vol. 15, no 1, pp. 263-268.

Chatfield, C. (1993) Neural Networks: Forecasting Breakthrough or Passing Fad? *International Journal of Forecasting*, Vol. 9, no 1, pp. 1-3.

Chen, S. –T., Yu, D. C. and Moghaddamjo, A. R. (1992) Weather Sensitive Short-Term Load Forecasting Using Non-fully Connected Artificial Neural Network, *IEEE Transactions on Power Systems*, Vol. 7, no 3, pp. 1098-1105.

Chikobvu, D. and Sigauke, C. (2012) Regression-SARIMA modelling of daily peak electricity demand in South Africa, *Journal of Energy in Southern Africa*, Vol. 23, no. 3, pp. 23-30

da Silva, P. A. and Moulin, L. S. (2000) Confidence Intervals for Neural Network Based Short-Term Load Forecasting, *IEEE Transactions on Power Systems*, Vol. 15, no 4, pp. 1191-1196.

Eskom Holdings SOC Limited Integrated Report (2013) GX 0001 revision 14 May 2014. [Online] Available from http://www.eskom.co.za [Accessed: 22nd February 2015].

Gross, G. and Galiana F.D. (1987) Short-Term Load Forecasting, *Proceedings of IEEE*, vol. 75, no. 12 pp. 1558-1573.

Gupta, M. (2012) *Weather Sensitive Short-Term Load Forecasting Using Non-fully* connected Feedforward Neural Network, Master's Thesis in Engineering, Thapar University, Patiala (India).

Hamid, M.B.A and Rohman, T.K.A (2010) Short – Term Load Forecasting Using an Artificial Neural Network Trained by Artificial Immune System Learning Algorithm, *IEEE, 12th International Conference on Computer Modelling and Simulation (UKSim), 24 – 26 March, Cambridge*, pp. 408 – 413.

Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation*. 2nd Ed. Prentice Hall, Upper Saddle River (New Jersey), 842 pages.

Heaton, J. (2008) *Introduction to Neural Networks for Java.* 2nd Ed. St. Louis: Heaton Research Research, 438 pages.

Hedden, S. (2015) How do we solve South Africa's energy crisis, World Economic Forum [Online] Available from: https://www.weforum.org/agenda/2015/09/how-do-we-solve-south-africas-energy-crisis/ [Accessed: 15th December 2015].

Hernandez, L., Carlos, B., Javier, M.A., Carro, B., Sanchez-Esguivillas, A.J, and Lloret, J. (2013) Short-Term Load Forecasting for Microgrids Based on Artificial Neural Networks, *Energies*, Vol. 6, pp. 1385 – 1408.

Hippert, S. H., Pedreira, C. E., and Souza, R. C. (2001) Neural Networks for Short-Term Load Forecasting: A Review and Evaluation, *IEEE Transaction on Power Systems*, Vol. 16, no. 1, pp. 44-55.

Hong, T. and Fan, S. (2016) Probabilistic Electric Load Forecasting: A Tutorial Review, *International Journal of Forecasting*, Vol. 32, no. 3, pp. 914 – 938.

Hyndman, R. J. and Khandakar, Y. (2008) The Forecast Package for R, *Journal of Statistical Software*, Vol. 27, no. 3, pp. 1-22.

Janacek, G. and Swift, L. (1993) *Time Series: Forecasting, Simulation, Applications*, West Sussex: Ellis Horwood Ltd, 331 pages.

Kalogirou, S.A., (2001) Artificial Neural Networks in Renewable Energy Systems Applications: A Review, *Renewable and sustainable energy reviews*, Vol. *5, no.* 4, pp.373-401.

Khotanzad, A., Davis, M. H. and Abaye, A. and Maratukulam, D. J. (1996) An Artificial Neural Network Hourly Temperature Forecaster with application in Load Forecasting, *IEEE Transactions on Power Systems*, Vol. 11, no. 2, pp. 870-876.

Khotanzad, A., Afkhami-Rohani R., Lu, T., Abaye A., Davis, M. and Maratukulam, D. J. (1997) ANNSTLF – A Neural-Network-Based Electric Load Forecasting System, *IEEE Transactions on Neural Networks*, Vol. 8, no. 4, pp. 835-846.

Kumar, B. S. (2014) Short Term Load Forecasting Using Artificial Neural Networks, *International Journal of Research and Communication Technology*, Vol. 3, no. 2, pp. 247-255.

Kumar, M. (2009) *Short-Term Load Forecasting Using Artificial Neural Networks*, B. Tech, National Institute of Technology, Rourkela.

Lee, K. Y., Cha, Y. T., Park, J. H. (1992) Short-Term Load Forecasting Using an Artificial Neural Network, IEEE Transactions on Power Systems, Vol. 7, no. 1, pp. 124-132.

Mandal, P., Senjyu, T, Urasaki, N., and Funabashi, T. (2006) A Neural network based several-hour-ahead electric load forecasting using similar days approach, *Electrical Power and Energy System*, Vol. 28, pp. 367 – 373.

Moghram I. and Rahman S. (1989) Analysis and Evaluation of five Short-Term Load Forecasting Techniques, *IEEE Transactions on Power Systems*, Vol. 4, no. 4, pp. 1484-1491.

Moghadassi, A. R., Parvizian, F., Hosseini, S. M. and Fazlali, A. R. (2009) A New Approach for Estimation of PVT Properties of Pure Gases Based on Artificial Neural Networks, *Brazilian Journal of Chemical Engineering*. [Online], Vol. 26, no. 1 available from http://www.scielo.br/scielo.php?pid=S0104-66322009000100019&script=sci_arttext [Accessed: 2nd February 2015].

Mohamed, N., Ahmad, M. H., Suhartono and Ismail, Z. (2011) Improving Short Term Load Forecasting Using Double Seasonal Arima Model, *World Applied Science Journal*, Vol. 15, no. 2, pp. 223 – 231.

Momoh, J. A., Wang, Y. and Elfayoumy, M. (1997) Artificial Neural Network Based Load Forecasting. *IEEE International Conference on Computational Cybernetics and Simulation*. Vol. 4, pp. 3443-3451.

Murto, P. (1998) *Neural Network Models for Short-Term Load Forecasting*, M.Sc., Helsinki University of Technology, Helsinki.

Osman, H.Z., Awad, L.M., Mahmoud, K.T. (2009) Neural Network Based Approach for Short-Term Load Forecasting, *IEEE/PES Power System Conference and Exposition*, Seattle, WA, 15-18 March, pp. 1-8.

Papalexopoulos, A, Hesterberg, T. C. (1990) A Regression Based-Approach to Short-Term System Load Forecasting, *IEEE Transactions on Power Systems*, Vol. 5, no. 4, pp. 1535-1547.

Papalexopoulos, A. D., Hao, S., and Peng T.-M. (1994) An Implementation of a Neural Network Based Load Forecasting Model for the EMS, *IEEE Transactions on Power Systems*, Vol. 9, no. 4, pp.1956-1962.

Paretkar, P. S., Mili, L., Centeno, V., Jin, K. and Miller, C. (2010) Short-Term Forecasting of Power Flows over Major Transmission Interties: Using Box and Jenkins ARIMA Methodology, *IEEE Power and Energy Society General Meeting*, Minneapolis, 25 – 29 July, pp. 1 – 8.

Park, D.C., El-Sharkawi, M.A. and Marks II, R.J. (1991) Electric Load Forecasting Using an Artificial Neural Network. *IEEE Transactions on Power Systems*, Vol. 6, no. 2, pp. 442-449.

Park, J. H., Park, Y. M. and Lee, K. Y. (1991) Composite Modeling for Adaptive Short-Term Load Forecasting, *IEEE Transactions on Power Systems*, Vol. 6, no. 2, pp. 450-457.

Pawlak, Z. (1982) Rough sets, *International Journal of Parallel Programming*, Vol. 11, no. 5, pp. 341–356.

Peng, T.M.; Hubele, N.F. and Karady, G.G. (1992) Advancement in the Application of Neural Networks for Short-Term Load Forecasting, IEEE Transactions on Power System, Vol. 7, no. 1, pp. 250-257.

Qingle, P. and Min, Z. (2010) Very Short-Term Load Forecasting Based on Neural Network and Rough Set, *IEEE International Conference on Intelligent Computation Technology and Automation, ICICTA – 2010*, Vol. 3, pp. 1132 – 1135.

Ramezani, M., Falaghi, H., Haghifam, M.-R. and Shahryari, G. (2005) Short-Term Electric Load Forecasting Using Neural Networks, *IEEE International Conference on Computer as a Tool*, Belgrade, 21 – 24 November, Vol. 2, pp. 1525 – 1528.

Ranaweera D.K., Hubele, N. F., and Karady, G.G. (1996) Fuzzy logic for Short-Term Load Forecasting, *IEEE Electrical Power and Energy System*, Vol. 18, no. 4, pp. 215-222.

Reddy, S. S. and Momoh, J. A. (2014) Short-Term Electrical Load Forecasting Using Back Propagation Neural Networks. *IEEE North American Power Symposium*, 2014. pp. 1-6.

Rewagad, A. P. and Soanawane, V. L. (1998) Artificial Neural Network Based Short-Term Load Forecasting, *IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*, New Delhi, 17 – 19 November, Vol. 2, pp. 588 – 595.

Rouse, M. (2006) *Fuzzy Logic (programming glossary)* [online] available from: http://whatis.techtarget.com/definition/fuzzy-logic [Accessed on 3rd March 2015]

Sandoval, F. (2002) Short-Term Load Forecasting Using Artificial Neural Networks. Available from: https://samos.univparis1.fr/archives/ftp/preprints/samos160.pdf [Accessed: 2nd February 2015]

Senjyu, T., Takara, H., Uezato, K., and Funabashi, T. (2002) One-Hour-Ahead Load Forecasting Using Neural Network, *IEEE Transactions on Power Systems*, Vol. 17, no. 1, pp. 113-118.

Sinha, A. K. (2000) Short-Term Load Forecasting Using Artificial Neural Networks. *Proceedings of IEEE International Conference on Industrial Technology*, 2000. Vol. 2, pp. 548-553.

Taylor, E. L. (2013) *Short-term Electrical Load Forecasting for an Institutional/Industrial Power System Using an Artificial Neural Network*, M. Sc., University of Tennessee, Knoxville.

Taylor, J.W., Buizza, R. (2002) Neural Network Load Forecasting with Weather Ensemble Predictions, *IEEE Transaction on Power System*, Vol. 17, pp. 626 – 632.

Wasserman, P. D. (1989) *Neural Computing: Theory and Practice*, New York: Van Nostrand Reinhold, 230 pages.

Yang, Y., Wu, J., Chen, Y. and Li, C. (2013) A New Strategy for Short-Term Load Forecasting, *Hindawi Publishing Corporation, Abstract and Applied Analysis*, Vol. 2013, pp. 1 – 9.

Yoo, H. and Pimmel, L. R. (1998) Short-Term Load Forecasting Using a Self-Supervised Adaptive Neural Network, *IEEE Transactions on Power Systems*, Vol. 14, no. 2, pp. 779-784.

Zhang, G., Patuwo, B.E. and Hu, M.Y., (1998) Forecasting with Artificial Neural Networks: The state of the art, *International journal of forecasting*, Vol. 14, no. 1, pp. 35-62.