DEVELOPMENT OF AN IMPROVED SHORT-TERM PEAK LOAD FORECASTING MODEL BASED ON SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE AND NONLINEAR AUTOREGRESSIVE NEURAL NETWORK FOR NIGERIA POWER SYSTEM GRID

BY

KEHINDE RASAQ EKUNDAYO

P15EGEE8040

deerockeng2012@gmail.com

A THESIS SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING, AHMADU BELLO UNIVERSITY, ZARIA, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE (M.SC.) DEGREE IN ELECTRICAL POWER SYSTEMS ENGINEERING

DECEMBER, 2017

DECLARATION

I EKUNDAYO, Kehinde Rasaq hereby declare that the work in this dissertation entitled "Development of An Improved Short-term Peak Load Forecasting Model Based on Seasonal Autoregressive Integrated Moving Average (SARIMA) and Nonlinear Autoregressive Neural Network (NARX) for Nigeria Power System Grid" has been carried out by me in the Department of Electrical and Computer Engineering, Ahmadu Bello University, Zaria. The information derived from the literatures has been duly acknowledged in the text and a list of references is provided. No part of this dissertation was previously presented for another degree or diploma at this or any other institution.

EKUNDAYO, Kehinde Rasaq ______________________________

Signature & Date

CERTIFICATION

This Dissertationentitled “Development of An Improved Short-term Peak Load Forecasting Model Based on Seasonal Autoregressive Integrated Moving Average (SARIMA) and Nonlinear Autoregressive Neural Network (NARX) for Nigeria Power System Grid” by EKUNDAYO, Kehinde Rasaq meets the regulations governing the award of the degree of Master of Science (M.Sc.) in Electrical Engineering (Power Systems) of the Ahmadu Bello University, and is approved for its contribution to the knowledge and literary presentation.

Dr. Jibril Yusuf	____________________	_______________
(Chairman, Supervisory Committee)	Signature	Date
Professor Boyi Jimoh	____________________	_______________
(Member, Supervisory Committee)	Signature	Date
Dr. Jibril Yusuf	____________________	_______________
(Head of Department)	Signature	Date
Professor S.Z. Abubakar	____________________	_______________
(Dean, School of Postgraduate Studies)	Signature	Date

DEDICATION

This dissertation is dedicated to God Almighty, my Family and also to the memory of my Late Father Alhaji Suleimon Ishola, EKUNDAYO, may his soul R.I.P (Amen)

ACKNOWLEDGEMENT

The successful completion of the M.Sc. dissertation could hardly be possible without the helps and supports from a lot of individuals. I will take this opportunity to thank all of them who helped me either directly or indirectly during this important research work.

First of all,I wish to express my profound gratitude to Almighty God, the most gracious and merciful Lord for His unfailing love and mercies over me throughout my schooling and most especially during this research journey. May His name alone be glorified forever (amen).

My deepest gratitude goes to supervisor Dr. Y. Jibril and co-supervisor Prof. B. Jimoh for their moral support, understanding, valuableguidance and positive contributions toward the success of this research work. Sir, your continuous encouragements has been very beneficial throughout my schooling in Ahmadu Bello University. Most importantly, the completion of this research wouldn't have been possible without your participations. I am immensely grateful to you sir and may Almighty Allah continue to bless you (amen).

I would like to thank Prof. M. B. Mua'zu, Dr. S. M. Sani, and Dr. E. A. Adedokun, for sacrificing their time, energy and for always showing keen interest in myqueries and providing important suggestions to the success of this work despite their busy schedules, may God almighty reward you and your family abundantly.

I also acknowledge and appreciate the contributions of all the present and past lecturers of Department of Electrical and Computer Engineering, Ahmadu Bello University, Zaria for their support most especially; Dr. G. A. Olarinoye, Dr. P. U. Okorie, Dr. I. J. Umoh, Dr. H. T. Sikiru, Dr. A. Mati, Dr. A. M.S. Tekanyi, Dr. K. A. K. Abubilai, Prof. U.O Aliyu, Prof. A. Adegboye and my mentor Dr. Olaitan Akinsanmi, and many others who's their name could not be mentioned. My sincere appreciation also goes to Mallam A. Tukur for his administrative support towards the success of this work.

My special thanks go to my friends Dr. E. V. Tikyaa, Engr. A.S. Abubakar, Engr. O. A. Olaniyan, Engr. Yusuf, Engr. A.T. Salawudeen, Engr. S. Taiwo, Engr. Kuyan, Engr. B.O. Sadiq, Engr. Elvis, Engr. Ore-Ofe Ajayi and those who's their name could not be mentioned for their encouragement, moral supports and valuable contributions toward the success of this work. Oh, you people are wonderful and I am indeed grateful to you guys; your handsome reward is with God. The moments, I share with you all will always remain as happy memory throughout my life time.

Above all, I am sincerely grateful to my family especially my mother and my siblings; Mr. Adeleke and family, my twins' sister Taiwo (Mama Grace), my little sisters' Taiwo Abiola and Kehinde Abolanle and my brother Ibrahim and his family who shoulder most of my responsibilities in trials times. I thank you all for your prayers, patience, understanding, unconditional and endless love you showered me, I owe you a lot. Oh! Ibrahim, you are really a brother indeed this journey became a success because of your endless sacrifices, thank you very much. In the loving memory of my late Father Alhaji S. I. Ekundayo whom this journey began in his presence five years ago while he was on a sick bed, you told me you will be strong and healed when I returned not knowing that was your goodbye. Though you have gone but your prayers and words of advice still keep me going, may Almighty Allah forgive you and grant your soul eternal rest (R.I.P). To my in-law Godwin Idowu and family, I owe you big thanks. To all my nephews and niece thank you.

A big thanks to my family friends and fiancée most especially Engr. S. O. Obi, and his family, Folakemi Akinsehinwa, Rabecca Adeola, Mrs. Abe Olwakemi, Mrs. Uloma Olivia, Gloria Abel for your love, prayers and endless supports. Engr. you always encouraged me to be positive and trust God. I am overly indebted to you all, and I share this moment of happiness with you all.

I would also like to thank the management of the Transmission Company of Nigeria (TCN) and Nigeria Metrological Agency (NiMet) for providing me with necessary data and information's used for this research. To all those that have contribute directly and indirectly towards the success of this research, and my study in Ahmadu Bello University, Zaria. May God bless you all.

EKUNDAYO, Kehinde Rasaq
December, 2017

ABSTRACT

Electricity demand forecasting is a central and integral process for planning periodical operations and facility expansion in the electricity sector. Demand pattern is very complex due to the highly unpredictable behavior of consumers load consumption. Therefore, finding an appropriate forecasting model for a specific electricity network at peak demand is not an easy task for the utilities and policymakers. Many load forecasting methods developed in the past decades were characterized by poor precision, and large forecast error because of their inability to adapt to changes in dynamics of load demand. To fill this gap, this research has developed an improved short-term daily peak load forecasting model based on Seasonal Autoregressive Integrated Moving Average (SARIMA) and Nonlinear Autoregressive Neural Network (NARX). The developed model used SARIMA to captures the linear pattern (trend) and seasonality of the load time series but due to seasonal and cyclical nature of the load behavior which cannot accurately describe by linear regression model, NARX neural network was combined with SARIMA in order to improve and captures the non-linear patterns of the data series to minimize it forecast error. The structures of NARX was optimized by the tenets of chaos theory to avoid trial by error approach during training. A daily peak load data of Nigeria power system grid and daily average weather data for ten years, from January 1st, 2006 to December 31st, 2015 were used in this study to complete the short-term load forecasting using MATLAB 2015a environment for simulation and mean absolute percentage error (MAPE) as a measure of accuracy. The model forecast result was validated and compared with real peak load demand data of Nigeria grid in 2015 to measure the performance of the method. The evaluation results showed that the developed model trained with Levenberg-Marquardt training algorithm (*LM*) is more effective and performs better than classical SARIMA model with MAPE of 2.41%, correlation coefficient of 96.59% which is equivalent to an improvement of 63.70% in error reduction. Performance of different training methods also compare on the developed method and results shows that developed model training with *LM*shows more superiority and high precision over Bayesian regularization training algorithm (*Br*) with 1.6318% in error reduction equivalent to an improvement of 40.37%. Finally, the proposed model was further used to forecasts the daily peak load demand of year 2017 and 2018 successfully for planning and operations of the grid.

.

TABLE OF CONTENT

CHAPTER THREE: MATERIALS AND METHOD

CHAPTER FOUR: RESULTS AND DISCUSSION

CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

LIST OF FIGURES

LIST OF PLATES

Page number

LIST OF TABLES

Page number

LIST OF ABBRIVIATIONS

Acronyms	Definition
ACF	Autocorrelation Function
AIC	Akaike Information Criterion,
ANN	Artificial neural network (ANN)
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
ARIMAX	Autoregressive Integrated Moving Average with exogenous Input
BP	Backpropagation Neural Network Model.
DISCOs	Distribution Companies
DNWT	Denoising wavelet transform
DWT	Discrete Wavelet Transform
EMS	Energy Management System
FFN	False nearest neighbors
FFNN	Feed-forward neural networks
GA	Generic Algorithm
GENCOs	Generation Companies
ISOs	Independent Systems Operators
KF	Kalman Filter model
KPSS	Kwiatkowski, Phillips, Shmidt, Shin test
kV	Kilo Volt
LTLF	Long-term load forecasting
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MARSpline	Multivariate Adaptive Regression Splines
MATLAB	Matrix Laboratory
M. Sc.	Master of Science
MOD	Method of Delay
MTLF	Medium-Term Load Forecasting
MW	Mega Watt
MWH	Mega Watt-Hour
NARX	Nonlinear Autoregressive neural network with exogenous input

NBET	Nigerian Bulk Electricity Trader
NCC	National Control Centre
NERC	Nigerian Electricity Regulation Commission
NiMET	Nigeria Metrological Agency
NIPP	Nigerian Independent Power Producer
NRMSE	Normalized Root Mean Square Error
PACF	Partial Autocorrelation Function
PSO	Particle Swarm Optimization
PPAs	Power Purchase Agreements
QP	Quadratic programming optimization
RBFNN	Radial Bias Function Neural Networks
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
SARIMA	Sessional Autoregressive Integrated Moving Average
SARIMAX	Seasonal Autoregressive Integrated Moving Average with exogenous Input
SAWBS	South Africa Weather Services
SMA	Simple Moving Average
STLF	Short-Term Load Forecasting
SVR	Support Vector Regression (SVR)
TCN	Transmission Company of Nigeria
TSO	Transmission System's Operators

INTRODUCTION

1.1 Background of the Study

The growing concern of rapid urbanization globally has presented the world with tasks of meeting the daily increase in electricity demand and transaction in the past few decades (Al-Kandari &Solima 2005). But meeting this demand is still a major concern to many of the electric power utility companies globally because large electric energy produced cannot be stored;therefore,it must be consumed at the same time as it is being produced (Musa, 2017; Luiz *et al.*, 2015). To tackle this challenge of inadequate power generation and supply in many nations lead to deregulation of electric power industry. In Nigeria, power sector was deregulatedinto three entities: generation, transmission and distribution companies, which further unbundle into 18 successor companies (Nwohu, 2009).

Nonetheless the operation of these utilities company in the last decade has not really improved the situation of unavailability of electricity in the country. This may be attributed to poor planning and inadequate infrastructure facilities for generation, transmission and distribution of power to end users. On the other hand, in planning process pre-information about the future energy needs is the key strategies to guarantee efficiency which is achieve through forecasting. However, many utilities nowadays are still face with challenges of how to accurately forecast their loads requirement at various time. The foremost reason behind this task is that power demand in any locality or regions and nationwide vary with growth in population and economic activities (Popoola & Ibrahim 2014). This has made accurate load forecasting very critical for a day – to - day planning and operation of the power grid system (Musa *et al*., 2014).

Forecasting refers to the prediction of the load behavior for the future planning and decision making (Seifi & Sepasian, 2011). The outcomes obtained from load forecasting (LF) plays an important role in many decisions making by power system operators, both in the engineering side and financial perspective as they feed into their downstream components most especiallyin resources management (Banda & Folly, 2015; As'ad, 2012). Economically, the accurate load forecasting allows utilities, generators and other stakeholder in the to operate at the least cost (Mansour & Najmeh 2011; Afshin & Sadeghian, 2007) i.e., for every small decrease in forecast error, amount to significant savings in operation cost. It is however estimated that for every 1% decrease in loads forecast error committed, a 10 GW electric utility can save up to 1.6% million annually (Dwijayanti, 2013).Contrariwise, for any increase in load forecast errorswill make electric utility to over schedule (increase operating costs) or under schedule which risky (Bozkurt & Biricik, 2017; Popoola & Ibrahim, 2014; Mansour *et al.,* 2011). Bunn & Famer in 1985 pointed out that in the UK, a 1% increase in forecasting error implied a £10 million increasein operating costs. If the predicted electric load is higher than the actual demand, the operating cost will increase significantly, and it wastes scarce resources (Dwijayanti, 2013; Soliman & Al-Kandari 2010). On the other hand, if the predicted electric load is less than the actual demand, it can cause brownouts and blackouts, which can be costly, especially to large industrial customers. In addition, reliable load forecasting can reduce energy consumption and decrease environmental pollution.

Load forecasting (LF) is categorized into four (Luiz *et al.,* 2015) which includes: long term load forecasting (LTLF), mid-term load forecasting (MTLF), short term load forecasting (STLF) and very short-term load forecasting (VSTLF). Each category is done at different time horizons according to requirements but the thresholds for this time interval varies (Rafal, 2006).Long term load forecasting covers one to several years (1 to 50 years) for plant and infrastructure investment decisions; such as economical location, type and size of future power plants and transmission system expansion etc., (Luiz *et al.,* 2015; Sanjib, 2008). Mid-term load forecasting is prediction that covers a period of one week to one year, and is used for spinning reserve capacity, fuel allocation, maintenance scheduling and negotiation of forward contracts (Bozkurt & Biricik, 2017;Musa *et al.,* 2014). Short term load forecasting (STLF) covers one hour to a week ahead i.e. 1 – 168 hours), and is mainly used for generation scheduling purposes such as unit commitment, hydro-thermal co-ordination, power interchange and electricity sport price market evaluation, security assessment analysis (Banda & Folly, 2015; Luiz *et al.,* 2015). Very short-term load forecasting (VSTLF) is for few minutes to an hour ahead and is used for automatic generation monitoring and control among other functions (Mansour *et al.,* 2011; Harun *et al.,* 2009; Mohammed *et al.,* 2002).

The scope and the application of this research work is focus on STLF, this is done for nationwide, regional and can also be used for micro-grids operation for daily real-time generation control, security analysis and energy transaction planning in power utility so as to reduce cost. Demand is shaped hourly and it is impossible to start or stop production instantaneously in a huge power plant; therefore, production planning is mostly done in daily basis. Thus, STLF plays a crucial role for managing operations in electricity grid (Bozkurt & Biricik, 2017;Musa *et al.,* 2014). STLF is an old area of research in power system planning and operation. Despite the long history of active research in this area, a universal accurate load forecasting model is still a difficult task. With the deregulation of electricity markets, a variety of STLF models are developed. These models include econometric/statistical (pragmatic) models such as trend extrapolation (Chaoming *et al.,* 2005), linear and multi-linear regression (Kandananond, 2011), Box-Jenkins methodology which includes autoregressive integrated moving average models (ARIMA), seasonal ARIMA, ARIMA with exogeneous input (ARIMAX), etc. However, these methods always fail to avoid the influence of observation noise in the forecasting which amounts to greater losses in real terms (As’ad, 2012).

On the other hand, many research scholars have applied various numbers of state of the art artificial intelligent techniques (AI) to improve the forecasting performance in a short and long-term basis to address the drawback of econometric models. These methods include artificial neural networks(ANNs) and fuzzy logic systems (Amjady, 2006; Santos *et al.,* 2007; Cai, *et al.*, 2011), expert system model (Heiko *et al.*, 2009); Kalman Filter models (Al-Hamadi & Soliman 2010); Data mining method and machine learning; grey theory method (Niu *et al.,* 2010)and hybrid models (Li Li *et al.,* 2011; Banda & Folly, 2015). A support vector machine (SVMs) was also successfully employed to solve nonlinear regression problems (Jingmin *et al.*, 2006; Wei-Chiang, 2009). Support vector regression (SVR) was proposed to correct the error deviation in statistical techniques prediction.

Relationship between external factors and electrical load is not only quite complex but also nonlinear. This nature of load makes it difficult to predict future values with parametric modeling methods such as time series and linear regression analysis. Statistical and parametric methods require making assumptions on the rules of underlying system (Bozkurt & Biricik, 2017; Aqeel *et al.,* 2012; Harun, *et al.,* 2009). On the other hand, AI method (ANNs) require minimum number of assumptions to find out the relation between input and the output. For non-linear multivariate problems with large datasets, ANN is known to exhibit a much higher performance and therefore, seems to be appropriate for STLF. Contrariwise, autoregressive models sometimes outperform ANN based models' due to seasonality effect (Bozkurt & Biricik, 2017). Nevertheless, in spite of all the above techniques, there is yet notable prediction error due to the nonlinear, sophisticated and chaotic behavior of the load data.

1.2 Motivation

The motivation for this research is gotten from the fact that,accurate forecasts can avoid energy wastageas well as preventing system failure. Therefore, it is crucial to produce forecasts with low error in order to relieve the conflictbetween supply and need. The studies of Nigerian deregulated electricity market to improve the systems operational planning for future generation scheduling of resources without compromising on the reliability requirements for capacity expansion using load forecasting information are inadequate. This can be attributed to frequent fluctuation in power generation capacity and limited access of data to date. In addition, many STLF model developed for electricity market in Nigeria are outdated and span the period before the full deregulation which may not fit into the structure of the current market.

Nonetheless as liberalization is yet to be completed in present Nigeria electricity market and it is difficult for industrial, commercial and residential consumers to choose their distributing company to enjoy low prices. However, soon when market becomes fully deregulated, correct moves in the market on energy purchases, bulk power wheeling, fuel purchases, generation scheduling and units' commitment will depend on the accuracy of the expected electricity to minimize cost. The focus of this research work is to fill this gap by building an improved and more accurate STLF model that can run in real-time and cope with the dynamic nature of deregulated market. The initiative is to improve on the operation of the utilities and generator operators,Nigerian Bulk Electricity Trader (NBET) as well as guide Nigerian Electricity Regulatory Commission (NERC) in calculating the changes in electricity tariffs.

1.3 Significance of Research

The importance of liberalization in electricity market is to allow more investment into power sector with the hope of boosting economy drive. The amount of risk borne by electric utilities, power producers and marketers has increased substantially. And every player that signed a short to long-term contract cannot be certain that the future delivery of power at the specified price will earn them profit. Demand frequently differ from expectations at the time the contract was signed and the actual volume traded are not be enough to cover the costs incurred, this is attributed to the forecast error during planning and decision making. Therefore, this has made building more accurate and reliable model that can effectively forecast the load demand for generation scheduling and timely dispatch of power within a short period very necessary for all active market players in Nigeria power sector most especially in the monitoring and control of grid network operation activities by the Transmission Company of Nigeria (TCN). The significance of this research is that if this model is adopted and properly implemented by every stakeholder in Nigeria power sector, many business areas in engineering and financial aspect of power system utilities, system operator and generators will be enhanced and equipped greatly with ability to project their load demand and priceaccurately for better operation and decision making.Economically, this will also save time and cost. Most importantly effective utilization of resources and forecasts days ahead demand for real time application will becomes easy for becomes utilitiesand system operators.

1.4 Statement of Research Problem

The earlier STLF studies on Nigeria electricity grid operation are very limited and few of these research works are done on regional base while many are in the periods before the full deregulation and privatization of the power sector which has low significant impact on present electricity markets operation. In other word, utilities are still face with crucial task of how to accurately forecast their loads requirement at various time. Many conventional/single load forecasting methods have been utilized to address this problem, but most of them often lead to large forecasting errors due to inability to adapt to changing nature of load behaviour especially with the present dramaticchanges occurring in the structure of the utility industry because of deregulation and competition. Large forecasting errors can have an adverse effect on the power system as well as the economic viability of a utility.

Therefore, to reduce the forecasting error, Artificial neural network method has also been applied but the design of optimal network structures has not yet been successfully implemented, because of trial by error training approach which made the algorithms suffer vanishing gradient, poor precision, slow convergence and get trapped into some local optima. It has been found that ANN had difficulty modeling seasonal patterns in time series. As such, developing a more accurate short-term load forecasting model for a day -to- day operation and planning of power utilities systems' grid using an improved seasonal autoregressive integrated moving average (SARIMA) based on nonlinear autoregressive neural network with exogenous input (NARX) model becomes compulsory so as to improve the forecast precisionof our local grid by reducing the forecasting error. Whilst this will enhancethe grid and save cost.

1.5 Aim and Objectives

The research work aimis todevelop an improveshort-term daily peak load forecasting model based on seasonal autoregressive integrated moving average (SARIMA) and nonlinear autoregressive neural network (NARX) for Nigeria power system grid.

The objectives of this research are as follows:

I. To characterize the signature of chaos in the daily peak loads demand time series data collected for 120 months (2006 – 2015) using nonlinear dynamics analysis approach;

II. To develop an improved local STLF model to forecast daily peak load demand of Nigeria power system grid based on SARIMA and NARX neural network implemented in MATLAB 2015a Environment with its structures optimized with the tenets of chaos theory.

III. To validate the accuracy of the proposed model by comparison of the simulated results obtained with the actualdaily peak load data of 2015 collected from TCNusing Mean absolute percentage error (MAPE) performance metrics.

1.6 Methodology

The methodologies adopted for this research towards development of multivariate STLF model using concatenation of SARIMA and ANN optimized with chaos theory approach for Nigeria power system grid in MATLAB 2015a Environment are as follows:

I. To characterize and quantify the degree of chaos in the daily loads demand time series data collected for 120 months (January 1st, 2006 – December 31st, 2015) using nonlinear dynamics analysis approach requires the following steps:

1. Collect 10 years data (peak load demand in MW and weather records data)from Transmission Company of Nigeria (TCN) and Nigerian metrological Agency (NiMet);
2. Data pre-processing usingcubic spline interpolation and moving average for smoothening and fixing missing data.
3. Adopting the algorithms for nonlinear dynamics analysisto characterize and quantify the signature of chaos in peak load data using flow chart in Figure 3.1 (Echi *et al.*, 2015) to

determinethe metrics such as time series plot (visualization of variables), phase space reconstruction and phase portraits, and Lyapunov exponent;

II. Development of an improved SARIMA based on NARX model optimized with chaos theory approach for a STLF in MATLAB 2015a Environment for daily and weekly ahead peak load prediction. To achieve this objective requires concatenation of the two models' A and B.

A. Adopt Multiplicative Seasonal ARIMA Model from the work of in Yi*et al.*, 2013 and Maged & Elsayed, 2017. And replication of the model involves the following Box-Jenkin methodology steps (identification, estimation, diagnostic checking and forecasting).

1. Identification of t $SARIMA\ (p, d, q)(P, D, Q)s$ model structure:

(i) Carryout time series data preprocessing to detect dependencies between data using:

a) Time series plots for observation analysis;
b) Apply unit root and stationarity teststo the data series using Kwiatkowski, Phillips, Shmidt, Shin (KPSS) test. If the observed series in (i) is not stationary, applied differencing process 'd' to stationaries the data (Kwiatkowski *et al.*, 1992). Repeat the process until the data series is stationary.
c) Compute the model identification parameter to determine $(p, q)(P, Q)$ usingfit statistic considers the goodness-of-fit and parsimony; i.e. the simplest model with the least assumptions/variables, minimumSchwarz-Bayesian Information Criterion (SBIC) method

2. Estimate the model unknown parameters $(p, q)(P, D)$ using the maximum likelihood estimator and conditional least square method;
3. Carryout the diagnostic checking of the estimated residuals resultsin (iii) using goodness-of fittest statistics to confirm if the estimated models' assumption are satisfied, else repeat step $1. i(d)\ and\ (2)$above;
4. Use the model to forecastahead base on time lag$s$ (i.e. time horizon) selected.
   a) Plot the forecasted output and actual data for comparison
   b) Evaluate the model performance using MAPE

B. Developnon-linear autoregressive neural networks with exogenous input (NARX) model architecture using the following steps:
   (i) Data selection and inputting,
   (ii) Create and configure the network;
   (iii) Initialize the weight and biases;
   (iv) Training the network, using Levenberg Marquardt (LM) and Bayesian regularization (Br) training algorithms
   (v) Validate and test the performance of the network.
   (vi) Use the network for prediction and forecasting
C. Concatenate step II (A) and II (B) to form the proposed model as shown in Figure 3.4 and applied it on data obtained from TCN and NiMET for daily and weekly ahead STLF
   (i) The proposed model was used for daily and weekly prediction of peak load demand of year 2015 and the forecast results obtained were validated using
   a) Comparison between actual data and forecast output plot of SARIMA and NARX using LM algorithm
   b) Comparison between actual data and forecast output plot of SARIMA and NARX using Br algorithm
   c) Evaluation of the model results performance usingMAPE.
   d) Residual results evaluation using regression plot, performance function Vs epochs, error autocorrelation plots etc.
   e) Forecasting of daily peak generation for Nigeria power system grid in year2017 and 2018.

Finally, the implementation of the item A, B, C, D and flow charts in Figure 3.4 are converted into codes using MATLAB software (R2015a i.e. version 8.3.0.532) environment.

1.7 Dissertation Organization Outline

The general introduction has been presented in chapter one. The rest of chapters in this dissertation is organized as follows: Details of the Review of literatures comprises of the review of fundamental concepts and similar works is presented in chapter two. Chapter three contains the research materials and methods, while chapter four gives details of the results obtained and followed by discussion of such results. Finally, chapter five presents the conclusion and recommendations which gives the summary and provide guidelines for future work. Quoted references and Appendices are also provided at the end of the dissertation.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The literature review is categories into two sections, the review of fundamental concepts (theoretical framework) and the review of similar works which includes a review of some journals, conference proceedings, theses, dissertation, books and others.

2.2 Review of Fundamental Concepts

In this section, various theoretical frame work and concepts relevant to this research which includes concept of electrical power systems operation & planning management, short term load forecasting, application and needs for short-term load forecasting, Time series load model, short term load forecasting with ANN, nonlinear dynamics analysis of time series. The factors affecting load pattern are also highlighted.

2.2.1 Electrical Power System Planning & Operations

The electric power industry has evolved over many decades, from a low power generator, serving a limited area, to highly interconnected networks, serving many countries, or even continents.Nowadays,running this very large system is a really difficult task because system current situation is expected to be run in an efficient manner while given properly insights to the future load. This has caused numerous problems to be solved by both the educational and the industrial bodies (Seifi & Sepasian, 2011).

The word operation is the normal electric power term used for running the current situation. Referring to the future, the power system experts use the termplanning to denote the actions required for the future. The past experiences are always used for efficient operation and planning of the system.Power system planning is a process in which the aim is to decide on new as well as upgrading existing system elements, to adequately satisfy the loads for a foreseen future (Rafal, 2006). However, load forecasting is basically an important component of power system planning issues.Although some other words, such as, demand and consumption are also used instead of load, but load is the most common term. The actual term is electric load but sometimes called energy instead of load term, load could also mean the power. It is well agreed that both the energy (MWh, kWh) and the power (MW, kW) are the two basic parameters of a load (Seifi & Sepasian, 2011; Rob *et al.*, 2008).

2.2.2 Application and Need for Power System Planning

The planning in power system operations is required for the efficient operation of the system. Planning is a strategy used primarily to determine the most economicalway an electrical utility can schedule generation resources without compromising on the reliability requirements, operational constraints, policies and physical environmental and equipment limitations (Banda & Folly, 2015, Nitin & Mohanty 2015).Another application of the system planning is to assist transmission system operator (TSO), utilities, generators, investors and other market operators on future short-term and long-term system, equipment and facilities expansion in engineering side and financial perspective (Mansour & Najmeh 2011; Afshin & Sadeghian, 2007). Planning for the future expansion of a power system involves determining both the capacities and the locations of future components; namely, generation facilities, transmission/sub-transmission/distribution lines and/or cables and various substations (Rob *et al.*, 2008; Adepoju *et al.,* 2007), however, better planning requires load forecasting information (Muhammed & Sanusi, 2012).

2.2.3 Electric Power Peak demand

The peak demand planning is important aspect of power system operation as it determines investment requirements for adequate electricity generation or supply (Yusri & Hajime, 2013). A deficiency in electricity supply results in power cut thereby disrupting the smooth running of the economic machinery in a country and even undermining business confidence. Peak demand forecasting is thus a key exercise undertaken to avoid power failure at peak hours (Badurally *et al.,* 2011). However, a major challenge to the utilities is how and best way to formulate peak demand forecasts that safeguards electricity supply and eliminates the risk of power blackout while at the same time not engaging a country into excessive wasteful spending on electricity generation. Accurate prediction of daily peak load demand is very important for decision makers in the energy sector (Yusri & Hajime, 2013). This helps in the determination of consistent and reliable supply schedules during peak periods. Accurate short-term load forecasts enable effective load shifting between transmission substations, scheduling of startup times of peak stations, load flow analysis and power system security studies (Caston & Delson, 2010; Sanjib, 2008).

2.2.4 Factors Affecting the Load Profile Patterns

Good understanding of the electric power load characteristics helps to design reasonable forecasting models and select appropriate models in different situations. Electrical power load is characterized as non-linear, non-stationary and stochastic processes that undergo rapid changes in seasonality due to influential factors (Amakali, 2008). Identifying factors that affect load patterns is one of the steps in modeling load forecast. Some of these factors highlighted by Mohammed *et al.,* (2002) are:

1) **Economic Factors:** An economical condition in one area could affect the load shape. This condition could cover issues like the type of customers, demographic conditions, industrial activities, population growth. These conditions would mainly affect the long-term load forecasting.
2) **Time Factors:** Time factors include seasonal, weekly, and holiday effects. Examples for the seasonal effect include the number of daylight hours in one season, which affects the load pattern. Industrial load on weekdays will be higher than that of weekends. Holidays will have much effect on the load pattern as loads decrease below normal.
3) **Random Disturbances:** Large industrial consumers, like steel mills, may cause sudden load changes. In addition, certain events and conditions can cause sudden load changes such as popular TV shows or the shutdown of an industrial operation.
4) **Price Factors:** In electricity markets, fluctuation in the factors such as currency exchange rate, Goss Domestic Product etc., affect electricity pricesignificantly and this volatile could present a complicated relationship with system loads demand generation. This make it among the important factor in load forecasting (Ellen, 2015).
5) **Weather Factors:** Weather is defined as the atmospheric conditions existing over a short period in a location. Literature has established that weather components such as temperature, humidity, solar radiation, rainfall, wind speed, cloud cover, precipitation etc., play a major impact in affecting the load behavior at every hour of the day. This simultaneously changes the daily load curves. The characteristics of these components are reflected in the load requirements although some are affected more than others (Amakali, 2008).

6) **Other Factors***:*A load shape may be different due to geographical conditions. For example, the load shape for rural areas is different from that of urban areas. The load shape may also depend on the type of consumer. For instance, the residential load shape could be different from that of commercial and industrial consumers.

2.2.5 Short Term Load Forecasting

Short-term load forecastingis an essential part of daily operations of the utilities. No utility is able to work without it. Moreover, nowadays, STLF has become an urgent matter due to the complexity of loads, the system requirements, the stricter power quality requirements, and deregulation. The error in forecasting would lead to increased operational cost and decreased revenue (Dwijayanti, 2013). STLF in power systems operation has several functions, but the principal objective of short-term load forecasting (STLF) is to provide the load prediction for basic system operation planning (Mohammad *et al.*, 2002). For the ISO, load forecasting has several applications, including generation scheduling functions, providing information on generation reserve of the system. STLF provide the input data for power system security studies (load flow studies and contingency analysis) in case of loss of generator or of line so as to provide corrective measure to mitigate such occurrence (Harun, *et al.*, 2009; Banda & Folly, 2015). In addition, STLF would be useful for utility engineers in preparing the corrective plan for the different types of expected faults (Aqeel *et al.*, 2012; Adepoju *et al.*, 2007).

STLF plays an important role in the traditional monopolistic power systems. In a deregulated power system, utilities, GENCOs etc., would have to forecast the system demand and its corresponding price in order to make an appropriate market decision (Seifi & Sepasian, 2011; Mohammed *et al.*, 2002).On the other hand, short term forecasts use historical load, price and weather data. Introducing seasonality effect by using day of week, hour of the day and holiday information as input has shown to increase performance (Bozkurt & Biricik, 2017). Hence, reliable and accurate forecasts will be useful to the authorities such as GENCOs who are currently contemplating additional generation capacity, from renewable and non-renewable sources, and Nigerian Bulk Electricity Trader (NBET) who are due to renegotiate power purchase agreements (PPAs) with the major Nigerian independent power producers (NIPPs).

2.2.6 Nonlinear dynamics and Chaos and ItsApplication in Electric Power System

In section, the theoretical concept of nonlinear analysis and the application of chaos theory approach in electrical power system characterization will be discussed. The importance of this section is to examine the system behavior over the years under study and determine the appropriate techniques to be used in the modeling and prediction of the peak power load of Nigeria.

2.2.6.1 Definition of chaos

Chaos is usually associated with the sensitivity of a deterministic system to infinitesimal perturbations in initial conditions. The identification of chaos in atmospheric systems is due to an accidental discovery by Edward Lorenz, a meteorologist in 1961. This idea was first publicized by Lorenz in 1963, when he constructed a very crude model of the convection of the atmosphere as it is heated from below by the ground. Lorenz discovered, much to his surprise, that his model atmosphere exhibited chaotic motion which, at that time, was virtually unknown to Physics.

Numerous nonlinear dynamics researches have indicated that the characteristics of power load curve behavior is time dependent (time series) and this behavior is sometimes defined as stochastic in nature. However, these characteristics is neither periodic nor stochastic, but rather it is chaotic (Li Li *et al.*, 2011). Chaotic characteristics of any system can be best described with a nonlinear dynamic approach to reveal the seemingly stochastic complex dynamics behavior inherent in the system, which is one of the focuses of this research. Power system loads are a set of time series which exhibit chaotic characteristic in most cases. Chaos theory can analyze chaotic characteristics of time series and reveal the sequence itself of the objective regularities to avoid the predicted human subjectivity and improve the accuracy and credibility of load forecasting. At present, phase-space delay coordinate reconstitution method is employed to analyze chaotic characteristics of time series. Generally, the dimension is very great even infinite (Zhang *et al*., 2008). In fact, phase-space delay coordinate reconstitution method can expand the given time series to three-dimensional and even higher-dimensional space to allow the information in the time series visible for more accurate extractionand classification (Li Li *et al.,* 2011).

Hasselblatt &Anatole (2003) gave the most universally accepted mathematical definition and characterization of chaos which goes thus: for a dynamical system to be classified as chaotic, it must be generally characterized by the following properties:

1. It must be sensitive to initial conditions;
2. It must be topologically mixing (i.e. the system must be transitive); and
3. Its periodic orbits must be dense.

2.2.6.2 Basic Concepts and Analytic Tools for Detecting Chaos in Time Series

Nonlinear analysis involves the application of analytical tool to time series for qualitative and quantitative measurement of the signatures of chaos in any time series dataset. The qualitative tools are serve as a preliminary observation analysis tools that help us in visualization and identify the signatures of chaos in a time series. These includes: time series plots,power spectrum andphase portraits (i.e. the tools provide graphical visualization of variables whichreveal some of the features of their deterministic structure). Lyapunov exponent is used as a quantitative tool to test for chaos and compute system's predictability (Zeng *et al.,* 1992; Bedri & Akin, 2005).

A. Qualitative Tools

1) Time Series Plot:

Electrical load is a typical time series, since it consists of successive hourly or daily measurements. The time series plot of it often very useful step in chaotic analysis as they help in visual inspection of the data. If the data behavior shows irregularity, aperiodic or unpredictable characteristic, it is classified as random (stochastic) or chaotic systems. And if the behavior shows regular and repeating pattern, it can be classified as a periodic behavior and such system is stable at a fixed time interval (Echi *et al.*, 2015).

2) Power spectrum analysis:

Shannon (2013) stated that the power spectrum of a signal shows how a signal's power is distributed throughout the frequency domain. To convert the time domain series data to the frequency domain, we use the Fast Fourier Transform (fft). The power spectral density (power per Hertz) is obtained from the square of the absolute value of the Fast Fourier Transform (Burrus and Parks, 1985):

$$Power/Hz = abs\{fft[x(t)]\}^2 \quad (2.1)$$

A power spectrum plot is usually a graph of the power/Hz against the frequencies obtained as the inverse of the time values in the time series. The mean period of a time series is estimated from the power spectrum as the reciprocal of the dominant frequency (dominant peak) of the power spectrum plot (Telgarsky, 2013).Power spectrum analysis is a way of characterizing the attractors in the time series and is often used to qualitatively distinguish different periods embedded in a chaotic signal either is quasi-periodic or chaotic behavior. Power spectrum is mainly used for decomposition of the peak load time series in order to identify the series spectrum i.e. if the time series exhibit sharp spectral lines it means a periodic and it is quasi-periodic if the spectra has flat a broadband spectrum.Otherwise, it is chaotic signals (deterministic chaos) if the signal output is characterized by the presence of wide broadband noise in their power spectrum, with a continuum of frequencies in their oscillations (Ozer &Akin, 2005).

3) Phase portraits

Phase portrait is a two-dimensional projection of the phase space. It represents each of the state variable's instantaneous state to each other (Perry *et al.*, 2000). For a discrete time-series (e.g. peak load time series) a plot of $x_{n+\tau}$ against x_n points in phase space gives a phase portrait while for a continuous system such as a driven pendulum, angular velocity-position graph is a phase portrait. Chaotic and other motions can be distinguished visually from each other according to Table 2 (Ozer and Akin, 2005).

2.2.6.3 *Phase space reconstruction*

The first and most necessary step in the computation of certain metrics, such as correlation dimension and Lyapunov exponents. The construction of the phase space of the system or, in terms of observability, the construction of the phase portrait of the trajectories generated by the system (Li Li *et al.*, 2011; Lisheng *et al.*, 2013). In general practice, we observe only a single degree of freedom (one dynamical variable), in its time series, which is assumed to be generated by the dynamical system. The technique of reconstructing a system's dynamics in phase space from the observation of a single dynamical variable is the cornerstone of the empirical investigation of systems whose dynamical equations are not known. The basic method for the reconstruction of phase space, called the 'MethodofDelays (MOD)', which was being developed separately, but almost simultaneously, by Takens (1980) and Packard *et al.* (1980), has been used by almost all practical applications in the field.

In 1980, Taken's embedding theorem showed that the state of a deterministic dynamic system can be accurately reconstructed by a time window of finite length sliding over the observed time series called phase space. He further explained that when dealing with a time series $\{x(t_i) = 1, 2, 3, 4, \dots \dots \dots . \dots \dots \dots, N\}$,, where Nis the number of observations, the attractor can be reconstructed in an m-dimensional phase space of delay coordinates τ, by forming the vectors:

$$X(i) = [x(i), x(i+\tau), x(i+2\tau), \dots, x(i+(m-1)\tau)]^T, \quad i = 1,2,..M \tag{2.2}$$

where the intergers τ and m are the time delay and embedding dimensions respectively. In more quantitative terms, the delay time (or time lag) τ, is the shortest time over which there are clearly measurable variations in the observable signal (Takens, 1980). The embedding dimension m is the minimum dimension of the space in which you can reconstruct a phase portrait from your measurements and in which the trajectory does not cross itself, or in other words, it is the minimum number of independent variables (degrees of freedom) required to describe the dynamics of the system in phase space (Letellier, 2013). Thus, phase space vector $X(i)$ is an $M\ x\ m$ matrix, and the constant $m, M, \tau,$ and N are related by:

$$M = n - (m - 1)\tau \tag{2.3}$$

where M is the number of reconstructed phase space points (Rosenstein *et al.*, 1992), and τ is the delay time, i.e. the empirical time delay (or time lag), express as $\tau = k\Delta t\ (k = 1, 2 \dots \dots)$.The method of delays is illustrated in Table 2.1 as follows:

Table 2.1: Illustration of Reconstructing of Phase Space (Rosenstein *et al.*, 1992).

Phase Space Vectors	Reconstructed Phase Space Dimension				
	1	2	. . .	m-1	M
X_1	x_1	$x_{1+\tau}$		$x_{1+(m-2)\tau}$	$x_{1+(m-1)\tau}$
X_2	x_2	$x_{2+\tau}$		$x_{2+(m-2)\tau}$	$x_{2+(m-1)\tau}$
.	.	.	. . .	.	.
X_i	x_i	$x_{i+\tau}$		$x_{i+(m-2)\tau}$	$x_{i+(m-1)\tau}$
.	.	.		.	.
X_N	$x_{M+(m-1)\tau}$	$x_{M+(m-2)\tau}$		$x_{M+\tau}$	x_M

Table 2.1, can be seen that the scalar time series is converted into a slightly shorter series of vectors with overlapping entries. In topological terms, each vector can be considered to provide the coordinates of a point in an m-dimensional phase space, and these vectors also provide a way to reproduce the dynamics of the real unknown system and give us a view of its attractor (Karytinos, 1999).

B. Estimation of the time delay (τ) and embedding dimension

The time delay, τ is a positive integer number which determines the sampling rate of the data (the measurable variable). The delay time τ can be evaluated using the method of Average Mutual Information (AMI) postulated by Cellucci *et al.* The average mutual information is defined by (Rasoul *et al.*, 2014):

$$I(\tau) = -\sum_{ij}^{N} P_{ij}(\tau) ln \frac{P_{ij}(\tau)}{P_i(\tau)P_j(\tau)} \qquad (2.4)$$

Where $P_i(\tau)$ and $P_j(\tau)$ are the probabilities of finding $x_i$in the ith and $x_{i+\tau}$ in the jth interval, and $P_{ij}(\tau)$ is their joint probability. A plot of $I(\tau)$ versus the lag length τ is made and the first local minimum of the curve corresponds to the optimum selection of the delay time τ. However, the minimum embedding dimension m for phase space reconstruction in this work will be computed using the method of "false nearest neighbors (FNN)" developed by Kennel *et al.* (1992). This method works by checking the neighborhood of points embedded in projection manifolds of increasing dimension and eliminating the 'false neighbor'.

A natural criterion for catching embedding errors is establish if the increase in the distance between two neighbored points is large when going from dimension m to $m + 1$. Then, output produced by the function is the percentage of FNN versus increasing embedding dimension and has a monotonic decreasing graph. The minimum embedding dimension usually can be found where the percentage of FNN drops to almost zero (Kennel *et al.,* 1992). These will be shown by the results of the analysis in this work. Conclusively, to reconstruct phase space equivalently and predict peak load generation accurately, the embedding dimension and time delay are not chosen randomly. Here, adopts reconstruction spread of Taken embedding theorem illusive adjacent area algorithm to compute them.

C. Quantitative Tool:

1) Computation of the Largest Lyapunov Exponents:

Lyapunov Exponents(λ) is the average rates of exponential divergence or convergence of nearby orbits in phase space (i.e. it quantifies the exponential divergence of initial close state-space trajectories and estimates the amount of chaos in a system). It shows the long-term behavior of the time series and is a fundamental property that characterizes the rate of separation of infinitesimally close trajectories (Hakki, 2006). One of these methods is Sato's algorithm for the computation of the Largest Lyapunov Exponents λ using the relation:

$$\lambda(i) = \frac{1}{i.\Delta t} \cdot \frac{1}{M-i} . \sum_{j=1}^{M-i} \ln \frac{d_j(i)}{d_j(0)} \quad (2.5)$$

where Δt is the sampling period of the time series, and $d_j(i)$ is the distance between the jth pair of nearest neighbors after i discrete-time steps, i.e., $i.\Delta t$. Mis the number of reconstructed points. Rosenstein *et al.*, (1992) developed a new method for computing the largest Lyapunov exponent of small data sets due to the limitation of normalizing $d_j(i)$ from Sato's algorithm. This method proceeds by assuming the jth pair of nearest neighbors diverges approximately at a rate given by the largest Lyapunov exponent:

$$d_j(i) \approx C_j(i) e^{\lambda_1 (i.\Delta t)} \quad (2.6)$$

where $C_j(i)$ is the initial separation in $d_j(i)$. By taking the natural logarithm of both sides, they obtained the equation:

$$\ln d_j(i) \approx C_j(i) + \lambda(i.\Delta t) \quad (2.7)$$

The above equation denotes a set of approximately parallel lines (for $\mathrm{j} = 1, 2, \dots \dots \mathrm{M}$), each of which has a slope roughly proportional toλ_1. If a typical plot (solid curve) of the average value of $\ln d_j(i)$ i.e. $\langle \ln \mathrm{d_j(i)} \rangle$ aganist $\lambda(\mathrm{i}.\Delta\mathrm{t})$ is constructed; the slope of the dashed line fitted to the curves has a slope equal to the theoretical value of λ_1(the largest Lyapunov exponent), and this can also be easily and accurately calculated by using a regression line with the method of least squares, and the slope of this line is the largest Lyapunov exponent.

Echi *et al.*, (2015) defined the Lyapunov (error-folding) time or predictability T in terms of the largest Lyapunov exponent as the time within which it is possible to predict the dynamics of the system forward and gave an expression for it as:

$$T = \frac{\Delta t}{\lambda_1} \quad (2.8)$$

where $\Delta t$is the sampling interval and $\lambda_{1,}$ is the largest Lyapunov exponent.The bench mark test system for characterizing the dynamics of a time series by Ozer & Akin, 2005 in Table 2.2:

Table 2.2 Summary of the qualitative and quantitative tools for characterizing the dynamics of a time series (Ozer & Akin, 2005)

Solution of Dynamical System	**Fixed**	**Periodic**	**Quasi Periodic**	**Chaotic**
Time series plot	-	Shows regular cycles of oscillation with fixed amplitude and frequency	Shows nearly regular cycles of oscillation with slightly variable amplitude	Shows irregular cycles of oscillation with variable amplitudes and different frequencies
Power spectrum	-	Single dominant peak	Dominant peak and other sub-peaks	Broad band noise with continuum of frequencies; may peak at $f_o = 0$
Phase portrait	Point	Closed Curve	Torus	Distinct Shapes
Lyapunov exponent	-	$\lambda < 0$	$\lambda = 0$	$0 < \lambda < \infty$

Really, it is of no doubt that by adopting delay coordinate to reconstruct phase space in the process of studying the peak load forecast in this work for the given one-dimensional peak load time series, we can expand it tomultidimensional space, which can expose the hiddeninformation of it totally to carry on prediction accurately.

2.3 Autoregressive Integrated Moving Average (ARIMA) Modeling

A variety of different forecasting approaches are available to forecast time series data and it is important to realize that no single model is universally applicable. An approach presented here is the Box-Jenkins autoregressive integrated moving average (ARIMA) model. For more than half a century, the Box-Jenkins ARIMA linear models have dominated many areas of time series forecasting. One of the attractive features of the Box-Jenkins approach for forecasting is that ARIMA processes are a very rich class of possible models and it is usually possible to find a process which provides an adequate description to the data (Narizan *et al.*, 2011). Generally, a non-seasonal time series can be modeled as a combination of past values and past errors(Thays *et al.*, 2016;Narizan *et al.*, 2011), denoted as ARIMA(p,d,q) or expressed as (Diaz-Robles *et al.*, 2008):

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-i} + \epsilon_t \quad (2.9)$$

where $\phi_i$is the *i*-th autoregressive parameter, $\theta_j$is the *j*-th moving average parameterand $\epsilon_t$is the error term at time *t*. The equation (2.9) is of order (p,d,q) where it *AR* component can be decompose as $\phi(B)\nabla^d Y_t$ which can be mathematically express as (Xiping &Ming, 2012):

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t \quad (2.10)$$

where p is the number of the autoregressive terms, Y_t is the forecasted output, Y_{t-p} is the observation at time t_p, and $\phi_1, \phi_2 \dots \dots \phi_p$is a finite set of parameter. The ϕ terms are determined by linear regression while $d$indicates the amount of differencing and e_{t-i} indicates white noise. The θ_0 term is the intercept or 'constant term' while e_t is the error associated with the regression.The moving average (*MA*) component is given as$y_t = \theta(B)\varepsilon_t$ the formulation (Thays *et al.*, 2016):

$$Y_t = \mu - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} + e_t (2.11)$$

where $q$is the number of the moving average terms, $\theta_1, \theta_2, \dots, \theta_q$ are the finite weights or parameters set, and μ is the mean of the series. In summary, equation 2.10 and 2.11 can be generally express as (Diaz-Robles *et al.*, 2008):

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \mu - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} + e_t \quad (2.12)$$

The terms of AR-I-MA is defined by Nau (2014) as follows:

1) Lags of the stationarized series called "autoregressive" *(AR: p)* terms,

2) A series which needs to be differenced to be made stationary, i.e. an "integrated" *(I: D)* series, and

3) Lags of the forecast errors called "moving average" *(MA: q)* terms.

In most cases, the observe time series requires that the data be 'stationarized' to remove trend and stabilized the variance before the ARMA or ARIMA model fit is specified. This transformation is by applying either differencing, power transformation, logging, or deflating techniques to the data. A time series is stationary if all its statistical properties (mean, variance, autocorrelations, etc.) are constant. That is, it has no trend and "heteroscedasticity" (Zhang, 2003). Differencing has advantages over fitting a trend model to the data. It does not require estimation of any parameters, which makes it a simple approach. The other advantage is that differencing can allow a trend component to change through time (Dwijayanti, 2013).

2.3.1 Seasonal Autoregressive Integrated Moving Average Model (SARIMA)

In practice, many time series contain a seasonalperiodic component, which repeats every s observation.To deal with seasonality, the ARIMA model is extendedto a general multiplicative seasonal ARIMA (SARIMA) represented by $SARIMA\ (p, d, q) \times (P, D, Q)s$, (Narizan *et al.*, 2011; As'ad, 2012), where p, d, q and P, D, Q represent the non-negative integers polynomial order of autoregressive (*AR*), integrated (I), moving average (*MA*) parts for non-seasonal and seasonal component in the model. The *SARIMA* model is express mathematically as:

$$\Phi_P(B^s)\psi_p(B)\nabla^d\nabla_s^D\ y_t = \theta_q(B)\phi_Q(B^s)\varepsilon_t \qquad (2.13)$$

where the polynomial and the operators are defined as follows:

y_t is the forecast variable (i.e. power generation daily peak load demand at time t)

$\Phi_P(B^s)$is the seasonal *AR* polynomial of order $\mathrm{P} = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps}$

$\phi_Q(B^s)$ is the seasonal*MA* polynomial of order $\mathrm{Q}\ = 1 - \phi_1 B^s - \phi_2 B^{2s} - \cdots - \phi_Q B^{Qs}$;

$\psi_p(B)$ is the non-seasonal *AR* polynomial of order $p = 1 - \psi_1 B - \psi_2 B^2 - \cdots - \psi_p B^p$;

$\theta_q(B)$ is the non-seasonal *MA* polynomial of order $q = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$;

d is the order of non-seasonaldifferences; D: order of seasonaldifferences;

$\nabla^d = (1 - B)^d$ and $\nabla_s^D = (1 - B)^D$

The differencing operator Δ^d and the seasonal differencing operator$\Delta_s^D$is used to eliminate the non-seasonal and seasonal non-stationarity in the data (Stylianos e*t al.*, 2016). The back-shift operator$B$ is used to operates on the observe time seriesy_t by shifting it one-point time (i.e. $B^k(y_t) = y_{t-k}$). The term ε_t is a white noise error (i.e. random error zero with variance σ attime t) and m defined the seasonality period. The model development was based on Box-

Jenkins methodology which is a practical approach that consists of four iterative steps: (a) Identification, (b) Estimation, (c) Diagnostic checking, and (d) Forecasting (Zhang, 2003; As'ad, 2012; Stylianos *et al.,* 2016). In this research work, multiplicative SARIMA will be adopt from the work of Yi *et al.,* 2013. The four steps above can be implemented using Figure 2.1:

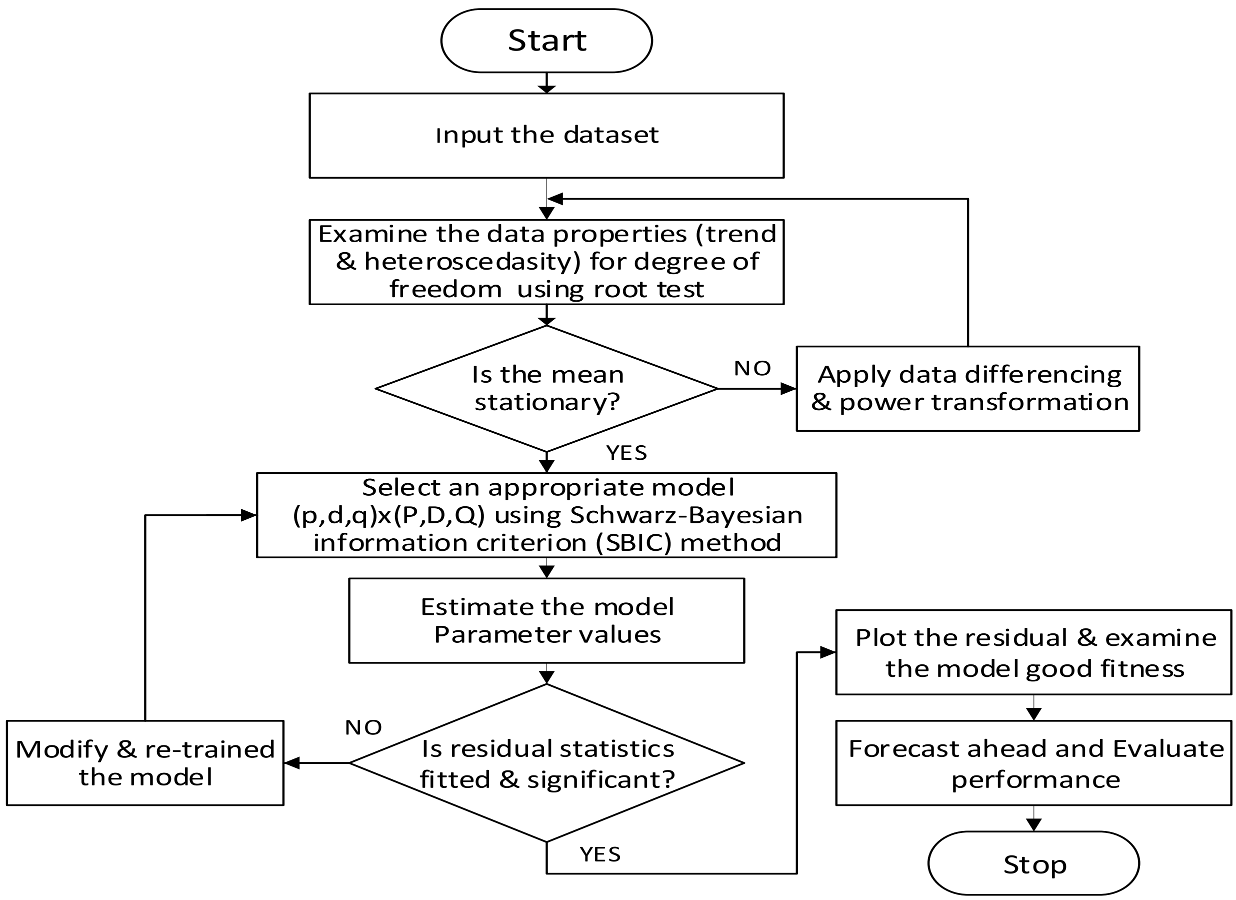


Figure 2.1: Flow chart of SARIMA Model (Stylianos *et al.*, 2016)

2.4 Artificial Neural Network (ANN)

Artificial neural network (ANN) is a type of artificial intelligent (AI) or computational intelligence (CI) method inspired by the way the biological systems of humans such as the brain process information (Banda & Folly, 2015). The human brain is made up of billons of interconnected neurons. The neurons consist of four different parts: cell body, dendrites, axion and synapses as shown in Plate 2.1. Dendrites receive information inform of electric potential via this connection (Manoj, 2009).

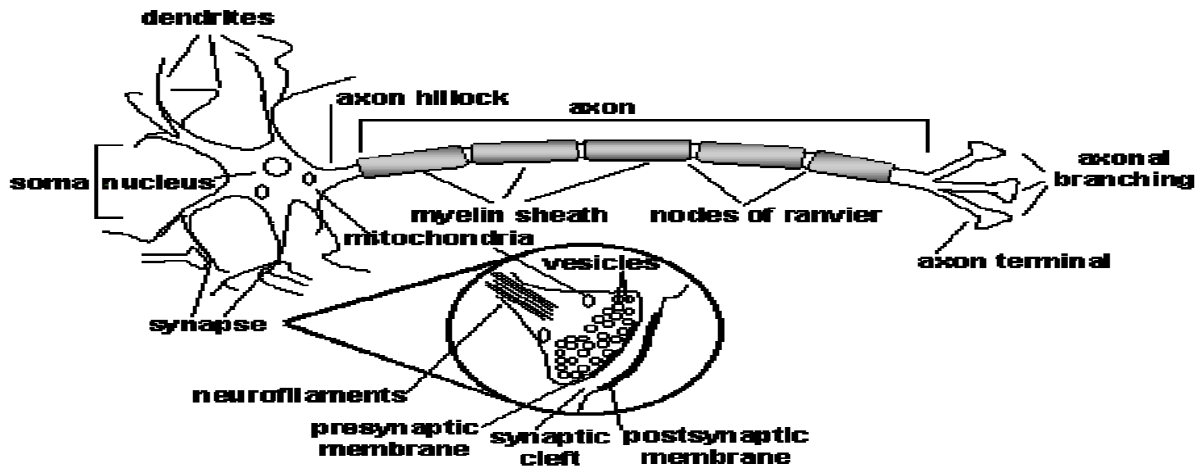


Plate 2.1: Schematic Diagram of a Biological Neuron (Manoj, 2009).

The axon connects many other neurons through connection points called synapses which produced a chemical reaction in response to an input, these potentials were given weight by synapses (Mohammed, 2002). All the neurons in the brain work in unison to make sure that all information received is processed as efficiently and accurately as possible (Banda & Folly, 2015). The artificial neurons try to simulate this kind of behavior displayed by the real neurons in the brain. ANN models can recognize trends, patterns, and learn from their interactions with the environment (Haykin, 2009). The evolution of artificial neural network (ANN) theory could be traced back to the experiment performed by McCulloh and Pitts in modeling bio-systems using nets of simple logical operations back in 1943. The soma sums all potentials given by the dendrites. If the sum of all potentials exceeds a certain value (threshold), the soma will fire an action potential through an axon and deliver to another neuron, then reset the potential.Artificial neural networks can be most adequately characterised as computational modelswith properties such as the ability to adapt or learn, to generalise, or to cluster or organise data, and which operation is based on parallel processing.

2.4.1 The Multilayer Perceptron (MLP) Neural Network

The perceptron or artificial neuron, was invented by Rosenblatt in 1957 at the Cornell aeronautical laboratory to mimic human memory, learning, and cognitive processes prior to his invention on the first robot that could learn to recognize and identify optical patterns and is modelled in a similar manner as the biological neuron (Amakali, 2008).The MLP neural networkis the most common type of feed-forward networks made of a series arrangement of perceptron in layers as shown in Figure 2.2. The MLP topology consist three layers i.e., an input, hidden and output layer. The neurons in the input layer only act as buffers for distributing the input signals $x_i(i = 1, 2 \ldots n)$to neurons in the hidden layer. Each neuron j in the hidden layer sums up its input signals x_i after weighting them with the strengths of the respective connections $w_{ji}$from the input layer and computes its output $y_j$as a function f of the sum (Beale *et al.*, 2014).

$$y_j = f\left(\sum_{i=1}^{n} w_{ji} x_i\right) (2.14)$$

Equation (2.14) can be rewritten in literal terms as:

$$output = f(input1 \times weight1 + input2 \times weight2 + \cdots)(2.15)$$

f is a simple transfer function such as a sigmoidal, hyperbolic, tangent or radial basis function.

The basic topology of an MLP is shown in the Figure 2.2:

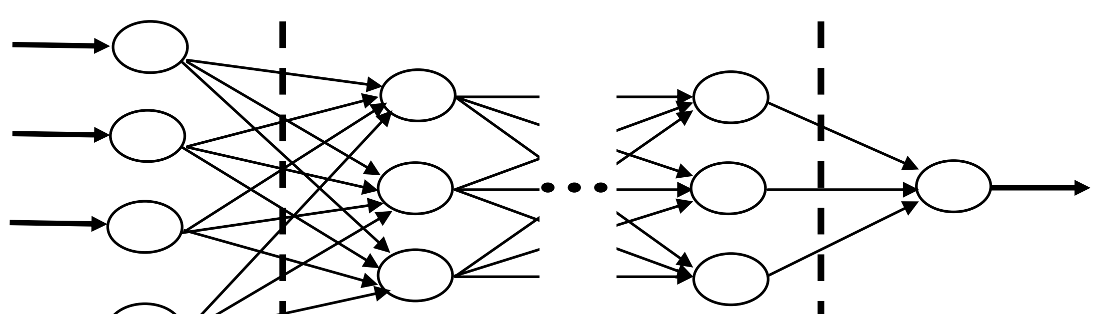


Figure 2.2:Basic Topology of Feed-Forward MLP Neural Network Model

The output from the neurons in the output layer is computed in a similar manner.

2.4.1.1 Training of artificial neural networks

A neural network has to be configure such that the application of a set of inputs produces (either `direct' or via a relaxation process) the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to `train' the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule (Ben & Patrick, 1996).

2.4.1.2 Paradigms of learning

The process of adjusting the weights and bias of a network in such a way that for every given input, the correct outputs are achieved is referred to as learning algorithm (Di-Piazza*et al.*,.2016). The most widely applied learning rule in STLF is categorise into two: supervised learning and unsupervised learning.

1) **Supervised learning**: in supervised learning,the network training is properly guided by providing it with set of input variables (e.g., weather variables,calendar variables, and preceding hour loads) and matching output (e.g., load). The learning idea isto find the coefficients or weights (w_{ij}, w_{jk}) that provide the best fit by adjust the weight until the error between the network output (y) and target function value (t) meet the specified threshold. Supervised networks include multilayer, radial basis, learning vector

quantization (LVQ), time-delay, nonlinear autoregressive (NARX), and layer-recurrent network.

2) **Unsupervised learning**: In an unsupervised learning the network is not guided with any pre-information. Therefore, the weights and biases are modified in inputs only, unsupervised networks include self-organizing maps and competitive layers.

2.4.1.3 Back Propagation Algorithm

The back-propagation algorithm and a gradient descent algorithm are the most commonly adopted training and learning algorithms in MLP. The BP learning algorithms is a generalization of the Widrow-Hoff error correction rule (Lee *et al.*, 1992). In artificial neural network training BP method finds the difference between the actual output and the reference or target output. The weights are then adjusted according to the errors and then propagated back into the system until the error is minimized. However, the main drawback with this training algorithm is that the training process can become very cumbersome and time consuming (Banda & Folly, 2015).The back-propagation algorithm gives the weight change Δw_{ij} in the connection between neurons i and j as follows (Beale *et al.*, 2014):

$$\Delta w_{ij} = \eta \delta_j x_i \tag{2.16}$$

or

$$weight\ change = learning\ rate \times error \times input \tag{2.17}$$

The error is given by (Jacobson, 2014):

$$error = target\ output - network\ output \tag{2.18}$$

The output $\hat{y}$ in a multilayer perceptron is mathematically generated from the expression:

$$\hat{y} = f\left(\sum_{i}^{p}(w_i x_i + b)\right) \tag{2.19}$$

where w_i are the synoptic weights, x_i are the model input vectors, and b denotes the biases.η is the learning rate and δ_j is an error gradient factor depending on whether neuron j is an input neuron or a hidden neuron. For output neurons:

$$\delta_j = \left(\frac{\partial f}{\partial net_j}\right)\left(y_j{}^{(t)} - y_j\right) \qquad (2.20)$$

and for hidden neurons (Al Shamisi*et al.,* 2011);

$$\delta_j = \left(\frac{\partial f}{\partial net_j}\right)\left(\sum_q w_{jq}\delta_q\right) \qquad (2.21)$$

In equation (2.20), $net_j$is the total weighted sum of input signals to neurons j and $y_j^{(t)}$ is the target output for neuron j while $\left(\frac{\partial f}{\partial net_j}\right)$ is the gradient. Since there are no target outputs for hidden neurons in equation (2.21), the difference between the target and actual output of a hidden neurons j is replaced by the weighted sum of the $\delta_q$terms already obtained for neurons q connected to the output of$j$. The process begins from the output layer, then the δ term is computed for neurons in all layers and then the weight updates evaluated for all connections iteratively. The weight updating process can happen after the presentation of each training pattern (pattern-based training) or after the presentation of the whole set of training patterns (batch-based training). One training epoch is completed when all training patterns have been presented once to the MLP.

Another commonly deployed method used to speed up the training process is the addition of a "momentum" term to equation (2.22) and effectively allowing the previous weight change influence the new weight change as follows (Jayawardena & Fernando, 1998):

$$\Delta w_{ij}(I+1) = \eta\delta_j x_i + \mu\Delta w_{ij}(I) \qquad (2.22$$

where $\Delta w_{ij}(I+1)$ and $\Delta w_{ij}(I)$ are weight changes in epochs $(I+1)$ and (I)respectively and μ is the "momentum" coefficient.Some other training functions are introduced to address these drawbacks of BP algorithm are Levenberg-Marquardt, Bayesian Regularization, Scaled Conjugate Gradient etc., which are more efficient than the Jacobian calculations (Amakali, 2008).In this research work,nonlinear autoregressive model with exogenous inputs (NARX) will be used to address the problem of vanishing gradient.

2.4.2 Nonlinear Autoregressive neural network with Exogenous inputs (NARX)

The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network, which has feedback connections covering several layers of the network.NARX has various advantages over MLP neural network such as less sensitivity to the problem of long-term dependencies and a very good learning capability and generalization performance. It addresses the problems of vanishing gradient and likely over-fitting through tapped delay memory (Arash & Farid, 2009). The model equation for the NARX model is (Mohammed *et al.*, 2016; Beale *et al.*, 2014):

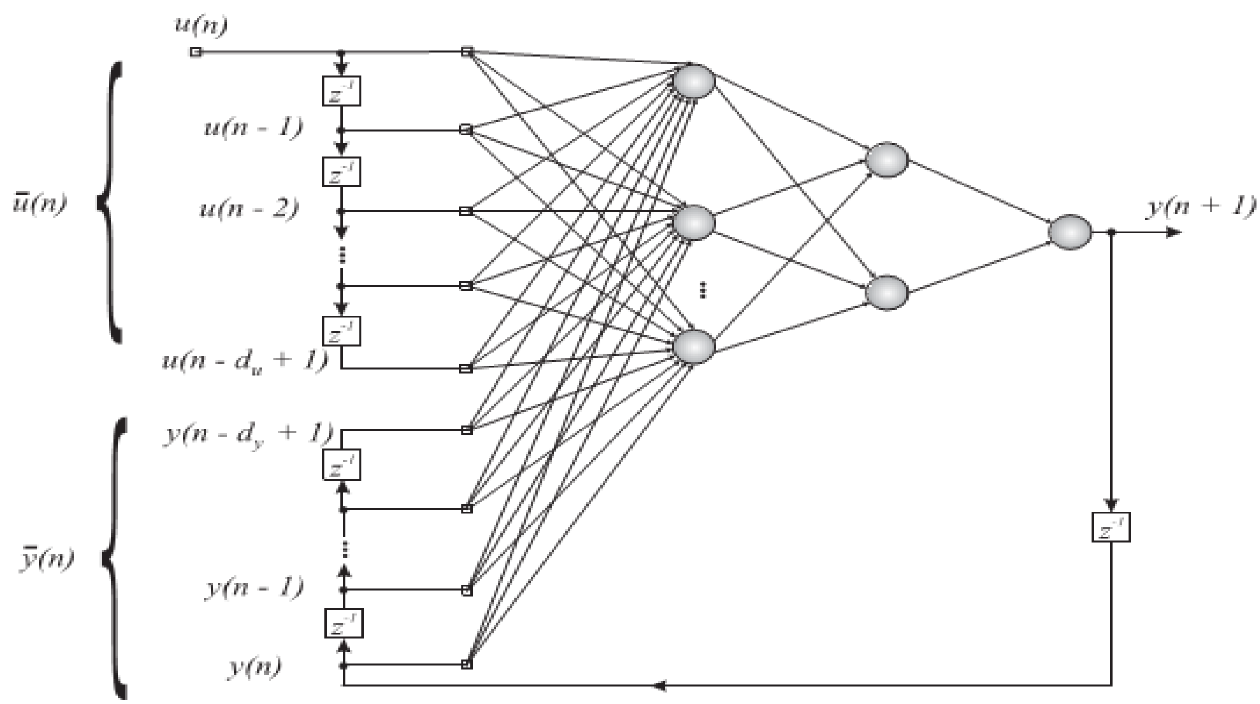
$$y(n+1) = f\left[y(n), y(n-1), .., y\left(n-d_y+1\right); u(n), u(n-1), ., u(n-d_u+1)\right] \quad (2.23)$$

where $u(n)$ and $y(n)$ are real valued and denote respectively, the input and output regressors of the model at time step n, while $d_u \geq 1$ and $d_y \geq 1, d_u \leq d_u$, are the input and output delays, respectively. Equation (2.23) can be simplified as:

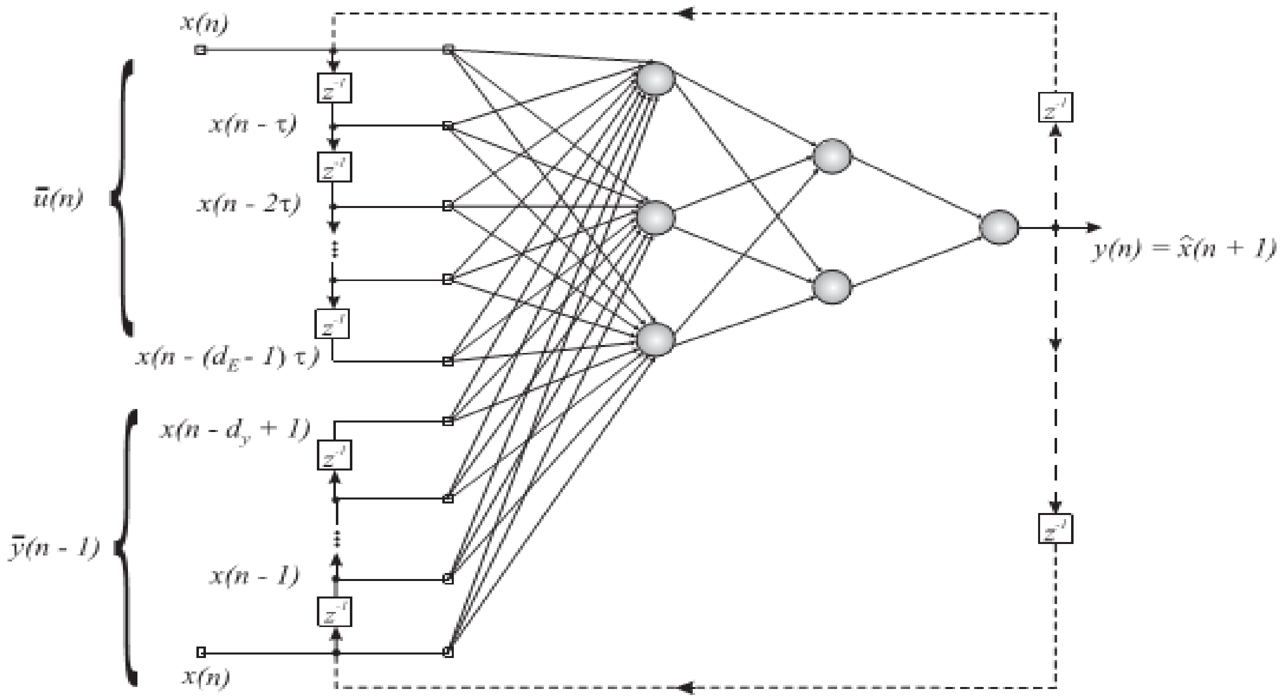
$$y(n+1) = f\,[y(n);\, u(n)] \quad (2.24)$$

The nonlinear mapping $f[\cdot]$ is generally unknown and can be approximated, using a feed forward MLP network (Menezes-Jnr &Barreto, 2006).

The next value of the dependent output signal $y(n+1)$ is regressed from the preceding values of the output signal and preceding values of an independent (exogenous) input signal. NARX network can be used to predict the next value of the input signal, as a nonlinear filter, in which the target output is a noise-free version of the input signal and in the modeling of chaotic nonlinear dynamic systems (Diaconescu, 2008). The structure of NARX architectures is illustrated in Plate 2.2.



(a.)



(b.)

Plate 2.2: NARX network during training and testing with d_u delayed inputs and d_y delayed outputs. (a.) Series-Parallel Architecture, (b.) Parallel architecture. The feedback loops (dotted lines) are required only during testing (Menezes-Jnr and Barreto, 2007a).

2.4.3 How to Build aNARX Neural Network

In building the network, some key parameters required to build a suitable network were specified. These include: the network name, the number of hidden layers, neurons in each layer, transfer function in each layer, training function, weight and bias learning function, data division function and performance function.

2.4.3.1 Name of the network

A suitable nomenclature was assigned for the network. The name chosen was chosen in such a way that it did not conflict with any function or filename in MATLAB else an error message will be received during the training. Examples of names commonly used in MathWork, 2016 are: 'net', 'netc', 'MyNetwork', my_net', 'my_network', 'nar_net', 'narx_net' etc.

2.4.3.2 The number of hidden layers

Normally, the number of hidden layers is usually selected at random and the network is trained and tested. The value that gives the best performance i.e. least mean square error and largest correlation coefficient is adopted as the optimal value for the number of hidden layers. On the other hand, in this work two hidden layers were used in addition to the input and output layers. This is in accordance with Takens' embedding theorem described in subsection 2.2.6.2 (Menezes-Jnr & Barreto, 2007a) and reflected in the next section.

2.4.3.3 The number of neurons

In determining the number of neurons in each hidden layer, here also the selection is done by starting with 10 neurons and increasing the number after the network is trained and tested until an optimal value is reached, i.e. the value that yields the best performance in terms of least mean square error and largest correlation coefficient. But in this work, trial by error approach will be avoided by using chaos theory to optimize the number of neurons in the 1st and 2nd hidden layers ($N_{H,1} and\ N_{H,2}$) by using the expression (Menezes-Jnr & Barreto, 2007a):

$$N_{H,1} = 2m + 1, \ \ N_{H,2} \cong \sqrt{N_{H,1}} (2.25)$$

where $N_H$is the number of hidden neurons and m is the embedding dimension of the target data. This is an extension of Takens' embedding theorem described in subsection 2.2.6.2.

2.4.3.4 Choice of network training functions

Neural network training functions are used to adjust the weights and biases in the network, to check whether the error goal has been met and recognize patterns in the data (function approximation and regression). Training functions are standard numerical optimization algorithms selected to optimize the performance function, and demonstrate very satisfactory performance for pattern recognition in neural network training using gradient and the Jacobian are computational technique called the backpropagation algorithm (Jaime & Shihab, 2017). In this work, only Levenberg-Marquardt (LM) training function and Beyesian regularization training function will be consider because they are faster and require less memory and optimizes better in overall performance for chaotic time series using NARX network. However, use gradient or Jacobian based methods and were adopted in this work, is shown in Table 2.3 (Hagan *et al.*, 1996):

Table 2.3: ANN training functions used in the research (Hagan *et al.*, 1996)

Function	**Name**	**Algorithm**	**Syntax**
Trainlm	Levenberg-Marquardt	Updates weights and biases according to Levenberg- Marquardt optimization: $jj = jx * jx$ $je = jx * E$ $dx = -(jj + I * mu)\backslash je$	*net.trainFnc ='trainlm'* *[net,tr] = train(net,…)*
Trainbr	Bayesian-Regularization	Updates weights and biases according to Levenberg- Marquardt optimization but also minimizes a combination of sum squared errors and weights	*net.trainFnc='trainbr'* *[net,tr] = train(net,…)*

2.4.3.5 *Choice of network transfer functions*

MATLAB built-in transfer functions were used in transferring data across the hidden layers via neurons while training the network. These functions which are also called ‘activation or threshold functions. The function considered in this work are: linear (purelin)and Hyperbolic Tangent Sigmoid (tansig). The pure-linear function was kept fixed for the output layer in all cases, so that the output would be display as single column vector after denormalized. The tangent-sigmoid (tan-sig) was preferred as the best choice for hidden layer because of its fast learning rate and sensitivity towards changes in number of data points and neurons in the hidden layer. Details of these functions are in Al Shamisi *et al*., (2011).

2.4.3.6 *Network learning functions*

In this work, the gradient descent with momentum (*learngdm*) was employed as the more preferred learning function because it is highly adaptive and fast and is set as the default learning function when creating a network using MATLAB.

2.4.3.7 *Data division functions*

The two main functions used in this research for dividing data into training, validation and test sets are *dividerand* (the default) and *divideblock.* These functions and their algorithms are illustrated in MathWorks, (2016). In this work, the divideblock function was used to divide the data into three parts for training, testing and validation in the ratio 70% : 15% : 15% respectively. For much large data sets, 60% : 20% : 20% can also be used. The fraction of data that was placed in the training, validation and test sets are expressed below (MathWorks, 2015):

$$training\ fraction = \frac{trainRatio}{(trainRatio + valRatio + testRatio)} \tag{2.26}$$

$$validation\ fraction = \frac{valRatio}{(trainRatio + valRatio + testRatio)} \tag{2.27}$$

$$test\ fraction = \frac{testRatio}{(trainRatio + valRatio + testRatio)} \tag{2.28}$$

2.4.3.8 Training and validation of the network

During the training process, the weights were adjusted systematically until the predicted output generated was close to the target (measured) output of the network. The comparison of the output signal with the desired response or target output consequently produced an error signal. In each step of iterative process, the error signal activates a control mechanism which applies a sequence of corrective adjustments of the weights and biases of the neuron via the weight/bias learning functions. These corrective adjustments continued until the training data attained the desired mapping to obtain the target output as closely as possible. After a series of iterations (also called *training epochs*) the neural network was trained and the weights were saved.

2.4.3.9 Forecast Performance Evaluation Metrics

The performance metrics are statistical error indices that is used to evaluate the accuracy of forecast output. This is achieved by finding the error difference between the actual measure data and the predicted output. This is done in order to determine how well model works and to compare betweenthe models.For this reason, various statistical methods of measuring error commonly used as indicators or performance metrics in literature are listed in Table 2.3 (Pillai *et al.*, 2014; Ellen, 2015; Maged & Elsayed, 2017):

Table 2.4: Additional Forecast Evaluation Metrics (Diaz-Robles *et al.*, 2008)

Parameter	**Mathematical formulation**	**Description**
Mean Squared error, *MSE*	$\frac{1}{n}\sum_{i=1}^{n}\left(L_i - L_f\right)^2$	Measure the mean of the square of the error
Root Mean Squared Error *RMSE*	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(L_a(i) - L_f(i)\right)^2}$	
Mean Absolute Percentage Error, *MAPE*	$\frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{L_a(i) - L_f(i)}{L_a(i)}\right\rvert \times 100$	

Coefficient of determination, *D*	R^2	Measures the proportion of the total variance in the observed data.
Schwarz-Bayesian information criterion, $\boldsymbol{SBIC}$	$n\log(SSE)+m\log n$ SSE =sum squared errors, n =no. of data points, m = no. of parameters in the model	Measures the goodness-of-fit of the model and also penalizes the number of model parameters.
Akaike information criterion, *AIC*	$\log v+\frac{2d}{n}$ v = loss function $= det[\sum_1^N \varepsilon(t,\theta_N)(\varepsilon(t,\theta_N)^T)]$ d = no. of estimated parameters, N = no. of values in estimated data set, θ_N = estimated parameters	Akaike's Information Criterion (AIC) provides a measure of model quality by simulating the situation where the model is tested on a different data set; the most accurate model has the smallest AIC.

For a good performance, the values of R^2 should be close to one and the optimal model can also be selected where the AIC/SBIC is lowest (Anctil & Rat, 2005).

2.4.4 Concatenation of Seasonal ARIMA and NARX Model

The word “concatenation” as defined by Oxford dictionary means a series of things depending on each other as if linked together. The concatenation of multivariate SARIMA and NARX artificial neural networkmodel for prediction is a chain of two prediction models (linear model and non-linear model) being linked together as in a union and its simulation process are done sequentially to giver better prediction performance in term of accuracy with reduce errors. NARX artificial neural network is an intelligent model used to increase accuracy linear models. What makes NARX artificial neural network different is that it can catch the nonlinear relationship between data rather than explaining an observation as a linear sequence of past values as in SARIMA case. Many literatures have misunderstood the sequential process of concatenation models to a hybrid models in resent past. Concatenation of ARIMA-ANN model has shown to be better than both conventional ARIMA model and ANN model separately. The Figure 2.3; illustrate the block diagram of a concatenation model:

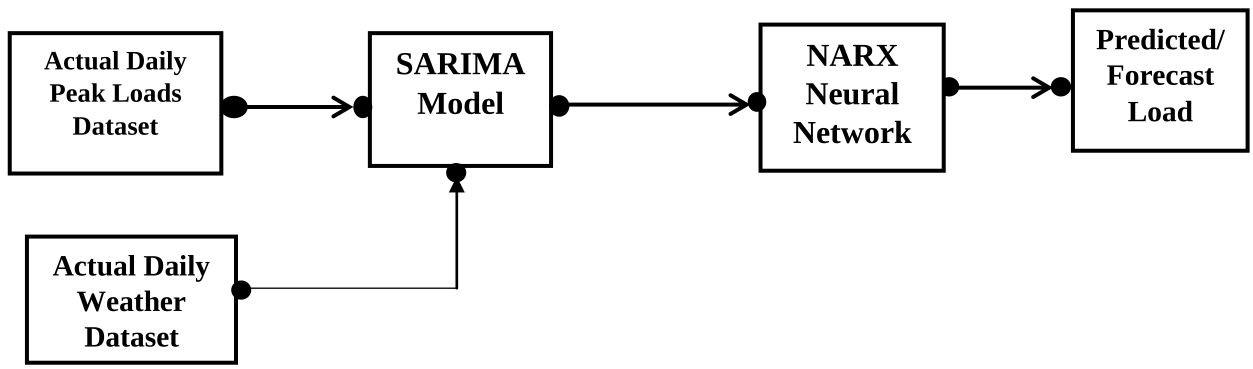


Figure 2.3: Block diagram of concatenation of SARIMA Based on NARX Neural Network Model for Short-term load forecasting

Many literatureshave proven that the integration of different models effectively improves their predictive performance(Khashei & Bijari, 2011).Parametric methods require making assumptions on the rules of underlying system, while ANNs require minimum number of assumptions to find out the relation between input and the output. A time series $Y_t$is considered as a function of a linear and nonlinear component i.e $f(L_t, N_t)$ as reported by Ozozen *et al.*, (2016). Therefore, it may be reasonable to consider the actual peak load demand time series of Nigeria power system grid to be composed of linear autocorrelation structure and nonlinear component as express in equations (Zhang 2003);

$$Y_t = L_t + N_t \quad (2.29)$$

Fit $L_t$using SARIMA, and e_t to be the residuals

$$e_t = y_t - L_t \quad (2.30)$$

The non-linear relations can be modeled from past residuals

$$e_t = f(e_{t-1}, e_{t-2}, \ \ldots, e_{t-n}) + \varepsilon_t \tag{2.31}$$

and

$$\varepsilon_t = \ y_{t,prediction} - \ Y_{t,actual} \tag{2.32}$$

where L_t is the forecast results of fit SARIMA in the linear and non-linear and $N_t = e_t$ denote nonlinear component.

2.4.5 Cubic spline interpolation

Cubic spline interpolation is a technique used to smoothen data, filter noise and can also be used to fill in missing data from a time series. Jeffrey (2002) noted that sometimes a function $f(x)$ which is assumed to be smooth is only known in the form of a set of discrete values $y_i = f(x_i)$ at a set of arguments $x_1, x_2, \ldots, x_n$ such that $x_1 < x_2 < \cdots < x_n$. When this occurs it often becomes necessary to estimate the value $f(\alpha)$ when α lies between two of the known arguments x_i. This process is called the interpolation of the function $f(x)$ between its known values, and the interpolated value $f(\alpha)$ is estimated using some or all the known values y_i. Cubic spline method is a very good method used to interpolate large series of data points and is hard to beat as a global interpolate.

The power of the cubic spline interpolation is that it uses not only all the data points (x_i, y_i) tabulated but generates the second derivative of the function $(x_i, y_i) \equiv (x_i, k_i)$ at the tabulated points and makes use of the k_i as well in the interpolation. The second derivatives functions are obtained from the solution of the linear system:

$$k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1}) = 6\left(\frac{y_{i-1} - y_i}{x_{i-1} - x_i} - \frac{y_i - y_{i+1}}{x_i - x_{i+1}}\right);$$

$$i = 1, 2, 3, \ldots, n-1 \tag{2.33}$$

With $k(a) = k(b) = 0$, since a and b are the extreme points for a natural spline (Kiusalaas, 2010). If the data points are equally spaced at intervals h, then $x_{i-1} - x_i = x_i - x_{i+1} = -h$, and hence equation (2.36) simplifies to:

$$k_{i-1} + 4k_i + k_{i+1} = \frac{6}{h^2}(y_{i-1} - 2y_i + y_{i+1}), i = 2, 3, \ldots, n-1 \qquad (2.34)$$

For x ∈ $(i, i+1)$, the spline interpolating function $f_{i,i+1}(x)$ is given by Kiusalaas (2010):

$$f_{i,i+1}(x) =$$

$$\frac{k_i}{6}\left[\frac{(x-x_{i+1})^3}{x_i-x_{i+1}} - (x - x_{i+1})(x_i - x_{i+1})\right] - \frac{k_{i+1}}{6}\left[\frac{(x-x_i)^3}{x_i-x_{i+1}} - (x - x_i)(x_i - x_{i+1})\right] + \frac{y_i(x-x_{i+1})-y_{i+1}(x-x_i)}{x_i-x_{i+1}} \qquad (2.35)$$

Equations (2.30) and (2.31) are used for the implementation of cubic splines interpolation and are implemented using the 'Spline' function in MATLAB.

2.4.6 Correlation coefficient

The Linear correlation coefficient (Pearson product moment correlation coefficient) *R* measures the strength and the direction of a linear relationship between two or more variables. It is mathematical given by (MathBits.com, 2015):

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2}\sqrt{n(\sum y^2) - (\sum y)^2}} \qquad (2.36)$$

The value of *r* is such that it always takes a value between –1 and+1, with +1 or –1 indicating a perfect correlation (i.e. all points would lie along a straight line, having a residual of zero). A correlation coefficient close to or equal to zero indicates no relationship between the variables. A correlation greater than 0.8 is generally described as *strong*, whereas a correlation less than 0.5 is generally described as *weak*. Correlation coefficients (r) range from -1 to 1 (Ellen, 2015).

2.5 Review of Existing Similar Works

Many research has been done to improve the accuracy of load forecasting which result into a variety of methods for various load forecasting related problems particularly in Short-term load forecasting (STLF). In this section, the similar works done by different researches in recent times are being related to the focus of research are reviewed with objective criticisms.

Li Li *et al.*, (2011)employed chaos theory to analyze the characters of the load time series and applied into power load forecasting. The load data for Shaanxi province power grid in China was used to complete the short-term load forecasting. Phase space was reconstituted and Lyapunov exponents of the load data were computed to judge whether the system was a chaotic system. Delay time and embedding dimension calculated were used to determine the structure of artificial neural network (ANN). Improved back propagation (BP)algorithm based on genetic algorithm (GA)was also used to optimize, train and forecast. The results show that the model in that paper is more effective than classical standard BP neural network model. Since their model was univariate, only historical values of load data were used. Although GA is a good optimization tool, but has some drawbacks when applied to improve BPNN such as difficulty in define fitness function and find the several sub-optimum solutions did not guaranty an optimal solution because of long simulation time to obtain the solution and that leads to high computational costs. In addition, the model only focused on one-hour ahead forecasting and intra.

Lei *et al.*, (2012) proposed a combined ANN model based on multi-dimension embedding phase space reconstruction for the analysis and prediction of wind power generation time series in the work. The wind power time series for a period of 2006. 1. 1 to 2006. 12. 31 from Fujin Wind Farm were used for training the prediction model and the wind power time series from

2007. 1. 1 to 2007. 12. 31 used to test the prediction results. Three models with dimensions $m_1 = 9,\ m_2 = 10,\ m_3 = 11$ were built using the quadprog function in MATLAB optimization toolbox environment, and the optimal combination weights $w_1 = 0.1866,\ w_2 = 0.5667,\ w_3 = 0.267$ of the three prediction models are obtained by self-learning. The analysis showed that the wind power system indicates typical chaotic characteristics and it prediction accuracy is affected by different embedding dimension which was difficult to choose in phase space reconstruction. In addition, proposed model's results were compared using normalized root mean square error (NRMSE) as performance metrics and it showed that wind power prediction with combined linear model have little smaller error reduction of 9.21% when compared with the pure chaos-based ANN model of error 9.35%. However, the limitation of the work is that the node number of hidden layer is determined by cut-and-try method which increases the training time which significantly caused low forecast accuracy and data loss.

Feilat *et al.*, (2011) presents a neural network (NN) approach for mid-term load forecasting based on historical monthly peak load data, temperature, humidity and wind speed data for (2006 -2010) of three regions in Oman. The monthly peak load was modeled as a function of monthly data, month index, temperature, relative humidity and wind speed. Three neural networks and a multiple linear regression model were developed to forecast the monthly peak load for months ahead. The results obtained by the neural networks were compared with the classical multiple linear regression models results and found that non-linear neural network outperforms classical multiple linear regression model with Mean Squared Error (MSE) of 56.95 to 144.3 and Mean Absolute Error (MAE) values of 5.16 to 1372. The drawback of this work is that degree of correlation coefficient between the weather variables used and their influence on peak load demand were not quantify with evidence in the results, several trial and error approach was used to found NN architecture and performance goal which increases their simulation time. However, the improvement in the accuracy of the model was not evident as more training and testing data were not available.

Kandananond (2011) developed a prediction models using three different approaches autoregressive integrated moving average (ARIMA), artificial neural network (ANN) and multiple linear regression (MLP) to forecast the electricity demand in Thailand based on the yearly historical data (i.e. population, gross domestic product (GDP), stock exchange index (SET index), and amount of export) from 1986 to 2010. The performance of these three models were assessed and compared. The results based on the error measurement performance indices (MAPE) of each model showed that ANN model was superior to other approaches (ARIMA and MLP), while paired tests (Wilcoson signed-rank test and paired t-test) used for comparison pointed out that there was no significant difference among the errors. In addition, other shortcoming of this model was that in ANN model the amount of network used for training was set at 200 and the fewer training samples were available which made the number of training limited. However, training small data samplegives low-generalization of the real trend and pattern of the system which makes the model precision accuracy not guarantee.

Hongzhan *et al.*, (2012) developed a hybrid model using ARIMA and Support Vector Machines (SVMs) for a short-term load forecasting of electric power company in Heilongjiang of China. The hybrid model used ARIMA to forecast the linear basic part of daily load and then used SVMs, to correct the deviation of forecasted non-linear sensitive part of load. To a large sample prediction, the simulation results showed that the hybrid model is much better than the two separate models and had a good prospective in applications. However, the main problem with SVM is the determination of its hyperparameters, which requires practitioner experience. Unsuitably chosen kernel functions or hyperparameter settings may lead to significantly poor performance. From a practical point of view, the drawbacks in this work was that, quadratic programming (QP) optimization and exponential kernel function were used for classification and training in order to correct the deviation sample which required high-dimensional space. And after a non-linear transformation was applied to the problem, the number of attributes in the new spaces become huge which made the training becomes cumbersome, thus runs slow in test phase. In addition, the result from graph curves showed that the actual load deviation curve compared with the forecasted load deviation curve using SVM model was not very accurate this leads to significantly poor performance of the model.

Hui*et al.*, (2012) employed the use of pure time series ARIMA, a hybrid ARIMA-Artificial Neural Networks (ANN) and ARIMA-Kalman Filter (KF) model for multi-step ahead prediction of wind speed data. The performance of the two hybrid models was better than that of the pure ARIMA model, and the performance of the ARIMA-Kalman model was better than that of the ARIMA-ANN model. The drawback of the method is that the ARIMA model parameters were selected intuitively, only week day load and weather parameters were used and the effect of weekend and other important factors were not considered in the model which make the model limited. In addition, the author claims that traditional models utilizing only ANN networks do not have the capability of adapting to sudden changes in weather conditions. The authors do not discuss the accuracy of their forecast which makes the methodsintricate for real time application.

Buhari & Adamu (2012) developed an ANN short-term load forecasting model for 24 hours ahead operational load planning for 132/33KV substation, Kano, Nigeria. The load forecast model was designed with multilayer feedforward neural network architecture and simulated in MATLAB R2008b ANN toolbox environment using Levenberg-Marquardt training algorithm.

Therefore, the model was tested to predict the next hour load using the previous daily average temperature data and year 2005 load data in MATLAB R2008b obtained from the power utility company in Kano. The results of the forecast for the 24-hourly peak loads of the next day were obtained based on the stationary output of the ANN with a performance mean squared error (MSE) of $5.8\ x\ 10^{-6}$ and compares favorably with the actual power utility data. This model is site-dependent, and requires that the load forecasters have adequate knowledge load and weather patterns. Thus, the model training parameter the were selected on trial and error approach.

Abdollah *et al.*, (2014) proposed a new hybrid forecasting method based on the wavelet transform, autoregressive integrated moving average (ARIMA) and artificial neural network (ANN) for short-term load forecasting. In the proposed model, the autocorrelation function and the partial autocorrelation function were utilised to see the stationary or non-stationary behavior of the load time series. Then, by the use of Akaike information criterion (AIC), the appropriate order of the ARIMA model was found. Thus, ARIMA model was used to captures the linear component of the load time series and the residuals would contain only the nonlinear components, while the nonlinear part was decomposed with discrete wavelet transform into its sub-frequencies. Finally, the outputs of the ARIMA and ANNs were summed and the empirical result claimed that the proposed hybrid method improved the load forecasting accuracy suitably.However, the main drawback was the complexity in discretization process i.e. extracting the coefficients of fine scales of the residual became computationally complex. while the approximation functions of the error signal and its coefficients does not represent the real signal which significantly limit the accuracy of the model in time domain and make this method not guaranty optimal solution.

Jason *et al.*, (2014) employed a real-time energy monitoring system using a customer data logging based on artificial neural networks (ANNs) model for hours ahead prediction. The model was implemented to manage customers' cumulative peak demand electrical consumption problems in a large government building in United State which always resulted in high electric utilities billing. The primarily focus of the work was to develop a model useful for efficient and cost-effective peak demand energy management of multiple government or corporate building complexes which is scalable and can be implemented for multiple building peak demand control. The performance of the developed model was compared against three other bench mark models: Multivariate Adaptive regression splines (MARSpline), Simple Moving Average (SMA) and linear regression. The result of ANN outperformed the other forecasting methods tested with MAPE of 3.9%. However, their model was univariate, only historical values of load data were used and ANN optimal model strutre was selected using trial by errorapproach with gradient descent backpropagation as training algorithm. This makes the training process cumbersome and likely suffer slow convergence with high possibility of being trapped in local optima.

Di Piazza *et al.*, (2014) presented a paper which focused on the prediction and forecast of climate time series, particularly useful for planning and management of the power grid, by artificial neural networks. In their work, they applied Artificial Neural Networks (ANNs) to the field of wind power generation. Two dynamic recurrent ANNs; the focused time-delay neural network (FTDNN) and the nonlinear autoregressive network with exogenous inputs (NARX), were used to develop a model for the estimate and forecast of daily wind speed. The daily wind speed and the daily maximum and minimum temperature in the period between 2010 and 2012 registered on Palermo weather station, in the northeast of Sicily, were used as dataset to train the ANNs. The ANNs-based models were experimentally validated and they both showed good performance since reliable and precise representations of daily wind speed were obtained. Results obtained were further applied to a turbine model to allow the produced power to be estimated in advance for energy management and planning purpose in smart grids.

Ina-Khandelwal *et al.*, (2014) proposed a novel technique of forecasting by segregating a time series dataset into linear and nonlinear components through Discrete Wavelet Transform (DWT). At first, DWT was used to decompose the in-sample training dataset of the time series into linear (detailed) and non-linear (approximate) parts. Then, ARIMA and ANN models were used to separately recognize and predict the reconstructed detailed and approximate components, respectively. In this manner, the proposed approach tactically utilized the unique strengths of DWT, ARIMA, and ANN to improve the forecasting accuracy. And, the hybrid method was implemented and tested on four real-world standard benchmark time series dataset which includes Annual number of lynx trapped in Mackenzie river district (1821–1934), Weekly exchange rates from British pound to US dollar (1980–1993), Monthly mining data of India (April 1981 to March 1998) and average monthly temperature of Las Vegas, US (June 1986 to May 2011) in MATLAB. While the ARIMA and ANN models are implemented through the Econometric and Neural Network toolboxes. The forecasting results are compared with those of ARIMA, ANN, and Zhang's hybrid models. Results clearly showed that the proposed method achieves best forecasting accuracies for each series. However, drawback of method is that the model was not apply to the real time practical problem and selection of appropriate wavelet was not considered which significantly increase the error and reduces the accuracy of the forecast.

Banda & Folly (2015), proposed a hybrid artificial neural network (ANN) and particle swarm optimization (PSO) method for next day's half hour short term load forecasting to reduce large error in forecasting because of changing environment and load characteristics. The model utilized 3 years (2009-2011) data set obtained from a real distribution network and South Africa weather services (SAWBS) for the analysis. PSO was used to alter the weights of ANN such that the resulting MSE for the training data was reduced. The simulation result shows that hybrid PSO-ANN performed better than the ANN using back propagation with MAPE of 2.84%. However, in this paper only focused on forecasting for weekdays, and the numbers of hidden layer neurons were determined by trial and error procedure thereby increasing the high computational time to achieve optimal network setup.

Narendra *et al.*, (2016), proposed a hybrid ARIMA–ANN model which combined the concepts of the recently developed moving average (MA) filter based hybrid ARIMA–ANN model, with a processing technique involving a partitioning–interpolation (PI) step. The improved prediction accuracy of the proposed partitioning–interpolation (PI) based hybrid ARIMA–ANN model is justified using a simulation experiment. Further, on different experimental TSD like sunspots TSD and electricity price TSD, the proposed hybrid model is applied along with four existing state-of-the-art models and it was found that the proposed model outperforms all the others, and hence is a promising model for TSD prediction.

Bozkurt *et al.*,(2017), created two separate STLF models based on SARIMA and ANN for Turkish Electricity. The model was used to forecast the last weeks of each month by including weekends and special days as test sets for fair and unbiased performance evaluation. The model evaluated the contribution of globally known factors on forecast performance, such as electricity price, weather parameters and currency, and the performances of the model were compared based on average of 12 test weeks. The authors claim better MAPE performance with the ANN than SARIMA forecasting method with a lower error of 1.80% and 2.60%

respectively. But in forecasts after holidays, SARIMA performs better than ANN. Many dependent factors with large training dataset and simpler feature sets were used in the model but selection of model number of hidden layers neurons was achieved through trial by error approach which make training very cumbersome and long. However,more using combination of many features makes the model becomes more complex and this decrease the precision of their forecast performance.

From the literature that was surveyed, it can be deduced that combine methods produce better results as compared to single methods. This principle will be investigated in this research by using concatenation of SARIMA and NARX neural network based model with its structure optimize with chaos theory approach. However, building this robust load forecast model will amplify the advantages of the individual models and minimize their limitations. The proposed model will considerthe effect of some weather factors affecting the load demand such as minimum and maximum temperature, wind speed, solar radiation on improving the accuracy of forecast. It is pertinent to conduct a correlation study between the electrical load and the weather elements to determine the variables with the most significant impact. In this way, one can avoid utilising variables that have little to no impact on the load profile. This research uses the correlation method in order to determine the load affecting variables.

As Nigeria power system grid currently undergoing many faces of restructuring in the area of generation, transmission and distribution networks, it becomes imperative for all active player in the market such as utilities, TSO, generators, power marketer, investors and other stakeholders in the industry to have up-till date look of hours, day and weeks ahead load demand as well as expected demand at peak period so as to save time and cost. This will enable them make important decision and plans their resources for efficient monitoring and control of resources for optimal allocation in the aspect of unit commitments and generation scheduling for economic dispatch of power, timely evaluation of systems security, equipment maintenance scheduling, power transactions evaluation etc. Presently Nigeria power system grid operation is currently facing a lot of challenges as many power plants (generators) are not operated at full capacity due to frequent dropping of load by distribution and poor evacuation of power by the TCN. The available power generation in Nigeria is currently below 4500MW daily (TCN daily broadcast 2017). However, demand for constant electricity increases day by day. Therefore, there is a need for large power user customers, and utilities as well as investors to have future knowledge of load growth on the network peak. As a solution, the developed model was used to forecast days' ahead peak load demand in year 2017 and 2018 for Nigeria power system grid operations.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

In this chapter, the materials required for this research, the study area of the research and the methods used were highlighted accordingly. The methodology and steps followed to achieve the stated objectives are detailedin this chapter and further summarize in a flowchart.

3.2 Materials and Equipment Required for the Study

In this section, the materials and equipment used for this research are highlighted.

3.2.1 Materials Required

The materials required for this research work are daily measurement of electrical peak load demand of Nigeria in megawatt (MW) recorded for a period of ten year i.e. January 1st, 2006 to December 31st, 2015. Others are daily average secondary measurement of daily weather elements data that affectsgeneration and demand in the study area.This weather data informationcollected was recorded from weather stations across Nigeria over a period of one decadei.e. from January 1st 2006 to December 31st, 2015. These includes: minimum temperature (℃),maximum temperature (℃), wind speed (km/hr), and humidity (%) and solar radiation ($MJm^{-2}day^{-1}$). That is, the model will be based on ten-years of daily data information; approximately 365* 10 = 3650 data points per variable, see Appendix C for raw data.

3.2.2 Equipment requirement

The equipment required used are as follows:

i. A laptop computer with the following specifications, 4GHz Intel Pentium (core i-series) processor, 4GB RAM and 1000GB Hard disk drive (a super computer or mainframe waspreferable),

ii. MATLAB software (R2015a i.e. version 8.3.0.532 and above) installed in the computer.

3.2.3 Data Sources

The data were collected from two agencies in Nigeria. The daily averages peak load (MW) data was collected from System Operation and Planning Department in Transmission Company of Nigeria (TCN), National Control Centre, Osogbo, Osun State. The data is for 330kV power grid network operations and is captured by Supervisory Control Data Acquisition System (SCADA). While weather data is obtained from the Forecast Services Department of Nigerian Metrological Agency (NiMet) Lagos.

3.3 The Study Area

The research study is case study of Transmission Company of Nigeria (TCN). TCN is own by Nigeria government and is among the 18-power operator company in Nigeria that succeeded Power Holding Company of Nigeria (PHCN) after deregulation in 2006. Through the System Operator (TSO), the company coordinate the operation and transmission of bulk electric power generated by the generation company stations and transmit it to the distribution company stations at 330kV and 132kV voltage level through the vertically integrated grid system.

3.4 Data Pre-processing

The secondary raw data collected were pre-processed to solve the following problems in the data which include:

1. Smoothening the data to solve the problem of missing data and outlier by using moving average and cubic spline interpolation methods,
2. Correlation analysis of data.
3. Normalization of the data

The detailed procedures explaining this process are shown in subsection 3.3.1.1 and 3.3.1.2.

3.4.1.1 Data Smoothening

Cubicspline interpolation method described in subsection 2.5.4, equation 2.34 and 2.35 and moving average were used to smoothen the data and remove the bad data (outliers) and fix the missing data. A few cases of leap years data in February of year 2008 and 2012 were removed in order to reduce the complexity and ambiguity in simulations' running time as well as to make the data correspond to 365. The details of the MATLAB code used for the process is shown in Appendix A1.

3.4.1.2 Normalization of the data

Data normalization was done using feature scaling method adopted from the work of Jaime & Shihab, 2017 because each dataset had different magnitudes. All values were brought within the range in the range of [-1, 1] or [0, 1] using the equation (3.1). The MATLAB commands "*mapminmax"* and "*mapstd"* were used in achieving the normalization of the data:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.1}$$

where X is a value within an initial dataset before normalization takes place, $X_{min}$and X_{max} represent the minimum and maximum value of the initial datavariable.

Furthermore, after all calculations were finalized, the output of the network was unnormalized using "$mapminmax\ ('reverse')$" and "$mapstd('reverse')$. The code simulates the network that was trained in the previous code, and then converts the network output back into the original units(MathWorks, 2016)

$$y_D = y_N(x_{max} - x_{min}) + x_{\min} \tag{3.2}$$

where $y_N$is the network output, $y_D$is the denormalized network output, x_{max} and x_{min} are the normalization parameters of the input datasets as described earlier in equation (3.1).

3.4.1.3 Correlation Analysis of Data

Selecting the correct combination of input parameters is the key to create an effective electrical load forecasting system. In order to attain a good combination, data are collected from sources related to some of the factors mentioned above. To have a model that can provide sufficient prediction accuracy with the minimum set of input, a correlation study was done to determine the weather variables that play a significant role in influencing the load demand using equation (2.36). The variables analyzed are daily minimum and maximum temperature values, humidity, wind speed and solar radiation measurement. Wind speed and humidity were removed after correlation study and forecasting, as some of the data were incomplete or missing and thereby gives poor linear relationship with load. The results of the correlation analysis contained in Table 3.1 and the detail of the complete MATLAB program used for the analysis is shown in Appendix A2.

Table 3.1: Correlation Analysis of the Input variables

Parameters	Solar radiation	Minimum temperature	Maximum temperature	Wind Speed	Humidity
Peak load	0.5581	-0.5224	0.6741	0.2808	0.0433

The results in Table 3.1 show that both maximum temperature and solar radiation have a stronger positive correlation of 0.67 and 0.5581 to the load than wind speed (0.2808) and humidity (0.0433), while minimum temperature has a strong negative linear relationship with load. These parameters were included as part of the input vector to the neural network in the proposed model.

3.5 Methodology

The details of the methods and steps adopted in achieving the itemized objectives in section 1.5 subsection A to C are discussed in this section.

3.5.1 Characterization of Peak Load Data Using Nonlinear Dynamic Analysis Approach

Electrical load demand in a large system is a variable with many complexities. The varying nature of power consumption corresponds to a countless number of scenarios that cannot be analytically solved by reflecting individual consumption from small to large power users. It is then important to take a first step by looking into the nature of the data and drawing some conclusions about its behavior and fit existing techniques to model it. Chaos theory can analyze chaotic characteristics of time series and reveal the sequence itself of the objective regularities to avoid the predicted human subjectivity and improve the accuracy and credibility of load forecasting.

In this section, the analytical tools for characterization of chaotic time series using an algorithm of nonlinear dynamic approach was adoptedform the work of Echi *et al.,* (2015). The algorithm was applied to analyse the ten-yearpeak load demand time series data collected. The essence was to quantify and characterize the dynamics of the data. The analysis will be done usingqualitative and quantitativeanalysis tools for detecting the signature of chaos in the series.The qualitative includes: time series plots, power spectrum and phase portraits. These are preliminary observation analysis tools that help us in visualization and identify the signatures of chaos in a time series. While Lyapunov exponent is used as a quantitative tool to confirm the presence of chaos the data and also to compute system's predictability.The details steps to implement of thesealgorithms is as follow:

1) Carryout quantitative analysis of the data to visualized the behavior of the system over the years and computation of phase portraits and phase space reconstruction parameters using detailed MATLAB program in Appendix A3 and A4. In power spectrum analysis the attractors in the time series is qualitatively distinguish by different periods embedded in a chaotic signal either is quasi-periodic or chaotic behavior. If the time series exhibit sharp spectral lines it means a periodic and flat random broadband spectrum indicates quasi-periodic otherwise, if the signal output display broadband power spectra with random noisemeans is deterministic chaos.
2) Apply the qualitative analysis tool to the reconstructed data to compute Lyapunov exponents and predictability using coded MATLAB program in Appendix A5. The implementation of these processes is described with the flow chart in Figure 3.1 based on Rosenstein *et al.,* 1992.Thus, the results of each metrics compute will be compared with the standard in Table 2.2.

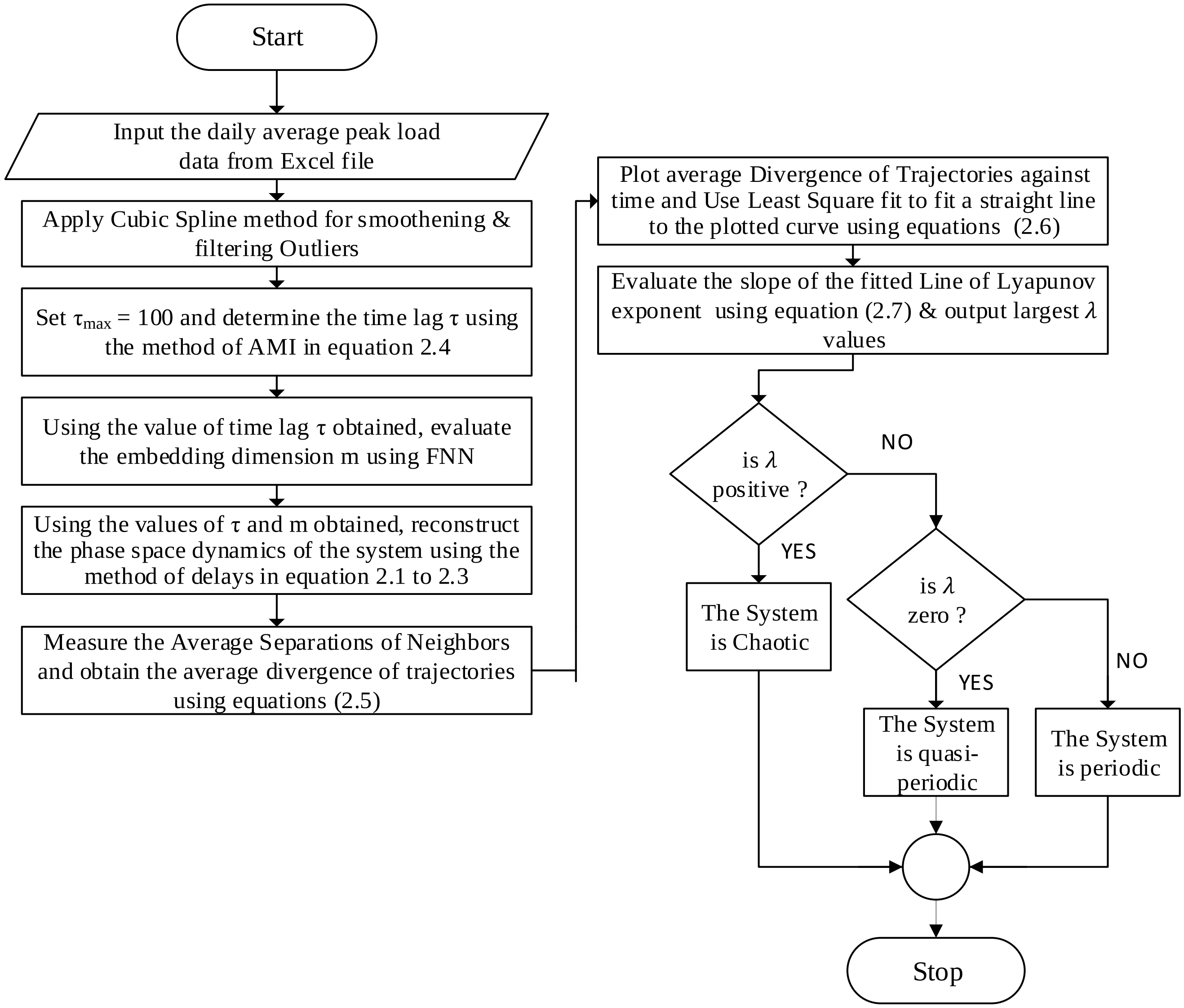


Figure 3.1: Flow Chart for Data Characterization Using Nonlinear Dynamics Analysis (Rosenstein *et al.*, 1992)

3.5.2 Development of an Improved STLFModel Based on SARIMA and NARX Neural Network Optimized with Tenets of Chaos Theory forDays Ahead Prediction

This section is used to achieved the itemized objective in section 1.5, subsection B.The development of the proposed method includes; a design and concatenation two models. First, adoptionof SARIMA model and second, is design of a NARX neural network. In the design of NARX structure, chaos theory approach was used for hidden layers neurons optimal selection.In the concatenation stage and forecasting stage, SARIMA is used to fit the data and the its residual predicted (outputs) will be used asan input into NARX neural network for simulation such that final forecast output will be an improved forecast from both SARIMA and NARX model. The detailed steps are as follows:

A) Adopting Multiplicative Seasonal ARIMA Models

This work adopted Box-Jenkin Multiplicative Seasonal ARIMA Model from the work of Yi *et al.*, 2013 and Maged & Elsayed, 2017. The formulation of the model includes four steps: model identification, parameter estimation,diagnostic checking and forecast future outcome.

1. **Model identification:** The identification of the SARIMA $(p, d, q)(P, D, Q)s$ model structure involvesthe following steps:

   i. Carryout time series data preprocessing to detect dependencies between data using:

   a) Time series plots for observation analysis;

   b) Apply unit root and stationarity tests to the series data usingKwiatkowski, Phillips, Shmidt, Shin (KPSS) test; and If the observed series in (i) is not stationary, applied degree of differencing process 'd' to stationaries the data (Kwiatkowski *et al.*, 1992). Repeat the process until the data series is stationary.

   c) Compute the model identification parameter to determine $(p, q)(P, Q)$ using the model fit statistic considers the goodness-of-fit and parsimony; i.e. the simplest model with the least assumptions/variables, minimum Schwarz-Bayesian information criterion (SBIC) and greatest loglikelihood selected (MathWorks, 2016).

$$SBIC = n \log(SSE) + m \log n \quad (3.3)$$

where $SSE = sum\ squared\ errors, n = no.$ of data points, m= no. of parameters in the model.;

2. **Estimate the model parameters:**the model unknown parameters $(p, q)(P, D)$ of the model is estimated using the maximum likelihood estimator and conditional least square (OLS) method;
3. **Diagnostic checking:**After applying the above steps, residuals of the model are analyze using the goodness-of-fit testto confirm if the statistical model models selected meet the following conditions:

   a) Normally distributed b) Zero mean c) No autocorrelation.

If not repeat the process in step $1. i(d)\ and\ (2)$ above;

4. **Forecast future outcomes based on known data:**the model is used to forecast ahead base on given time lag $s = 365$ (i.e. time horizon) selected.
   a) Comparison between the actual data and forecast output using time series plot
   b) Evaluate the model performance using MAPE
   c) Write the model equations

The forecasted output data of the fit SARIMA model will be used as inputs into the ANN model. The purpose of this final step is to forecast the error sequence of the fit model according to a specific time horizon, which cannot be defined using linear method like SARIMA.In summary, we implementedthe four iterative steps of SARIMA model in MATLAB environment and the details of the complete MATLAB program used is shown in Appendix B1 and B2.

B) Development of Improved NARX Neural Network Model for STLF

The development of Improved NARX neural network model includes the design of a NARX network, its structure, inputs and outputs, and simulationof the modelin MATLAB environment using a data sets described in subsection 3.3.The NARX neural network design will takes exogenous inputs, such as weather and day variables, and endogenous input (peak load), and produces a daily forecast for 365 days.Design of NARX Network structure: most important features that need in the designing a well performing NARX neural network model requires the following systemic steps.

(i) Data preprocessing

(ii) Data selection and inputting,

(iii) Building the network,

(iv) Configure the network and Initialize the weight and biases

(v) Training, validate and testing the performance of the network.

(vi) Use the network for prediction and forecasting

Step 1: Data selection and input

The input data were selectedbased on Table 3.1. In this work, to evaluate the proposed model two cases of time horizons (daily and weekly) will be used in simulation. Data input (data collection) in this step was done by importing each of the data from excel file outside the MATLAB environment using this script:

```
% Input the data from the excel file 'NiMet & TCN'
fid=fopen('exppp.txt','w+');
yData1 = xlsread('C:\Users\Abdul\Desktop\data\full2.xlsx',-1);
t=load('exppp.txt');
y=t(:,2);

%% load the input data into NARX
s=load('expsr.txt');
sr=s(:,2);
t1=load('expmint.txt');
mnt=t1(:,2);
t2=load('expmaxt.txt');
mxt=t2(:,2);
```

The data input comprises of 9-years data i.e. from 1st Jan. 2006 – 31st Dec., 2014 correspond to 3, 285 data point for training and 1-year data (1st Jan. – 31st Dec., 2015) will be used for evaluation.

Step 2 to 4: Building, configuration and initialize weight and biases of NARX Neural Network

The ANN used in the proposed framework is first trained in open-loop, i.e., the training set includes all the historical data for weather variables and also the historical data for the daily loads. In this sense, it is a feedforward network. Once the node weights are found in open-loop, the network is used to calculate the output, which is then fed as input to the NARX neural network, making it a recurrent network (feedback). Figure 3.2 shows the proposed structure. The reason why the closed-loop is used is that during training the actual output y(t) is available and it is fed to the neural network in order to determine the network weights, but during the forecasting phase the actual output is not available and therefore the predicted delayed output is used to produce a forecast.

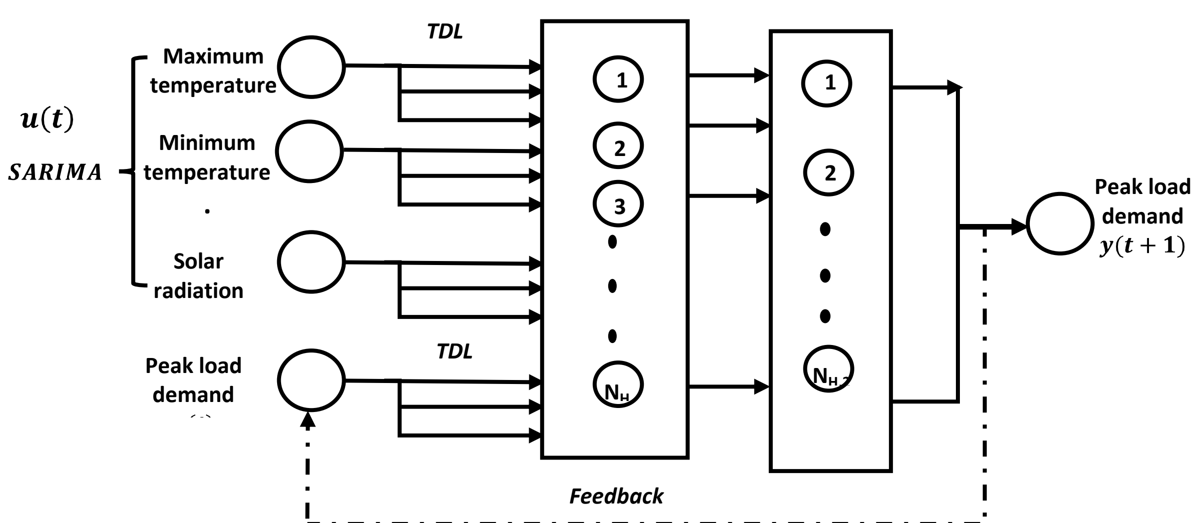


Figure 3.2: Proposed NARX Neural Network Structure

The following steps were taken to build and train a NARX network for time series prediction using command-line functions:

(i) A network was created using the syntax *'narxnet'* and it was assigned a name. The *narxnet* created has three inputs: the external input $u(t)$ and the feedback connection $y(t)$ from the network output $y(t+1)$. Each of these inputs has a tapped delay (TLD) line to store previous values as shown in Figure 3.2. The delays associated with each tapped delay line was selected using the time delay (τ) estimation method of average mutual information (AMI) described earlier in section 2.2.6.2 equation (2.3), (2.3) and (2.4). Also, in order improve the learning rate of the NARX model, the number of hidden layers and neurons were obtained using equation (2.25) as described in subsection 2.5.3.2 and 2.5.3.3. These steps were implemented in MATLAB environment, with command as follows:

```
inputDelays = 1:27;
feedbackDelays = 1:27;
hiddenLayerSize = [17 4];
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize);
```

where the hiddenlayersize vector [17 4] indicates that the model has two hidden layers containing 17 and 4 neurons respectively. While feedback delays size(number of tapped delay line) is given as 27.

(ii) The data were prepared for training network with tapped delay lines. It is crucial to fill delays with initial values of the inputs and outputs of the network. In MATLAB, the *'preparets'* function was used to facilitate this process as this returns the initial conditions needed to fill the tapped delay lines in the network and the modified input and target series. The function was implemented in MATLAB as follows:

[inputs, inputStates, layerStates, targets]

= preparets(net, inputSeries,{}, targetseries)

(iii)Partitioning the training data into train, test, and validation sample size using data division function *'dividerand'*in MATLAB. If *net.divideFcn* is set to *'dividerand'* (the default), then the data is randomly divided into the three subsets using the division parameters *net.divideParam.trainRatio, net.divideParam.valRatio*,and*net.divideParam.testRatio*. The ratio used for training, testing and validation is70% : 15% : 15% respectively. This is implemented by MATLAB program below:

```
% Set up Division of Data for Training, Validation, Testing
net.divideFcn = 'divideblock';
net.divideMode = 'sampletime';
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

Step 5: Training the network

(iv)The training functions usedare Levenberg-Marquardt algorithm *(trainlm)* and the Bayesian Regularization *(trainbr) training function.* The GUI program outputis shown in Figure 3.3:

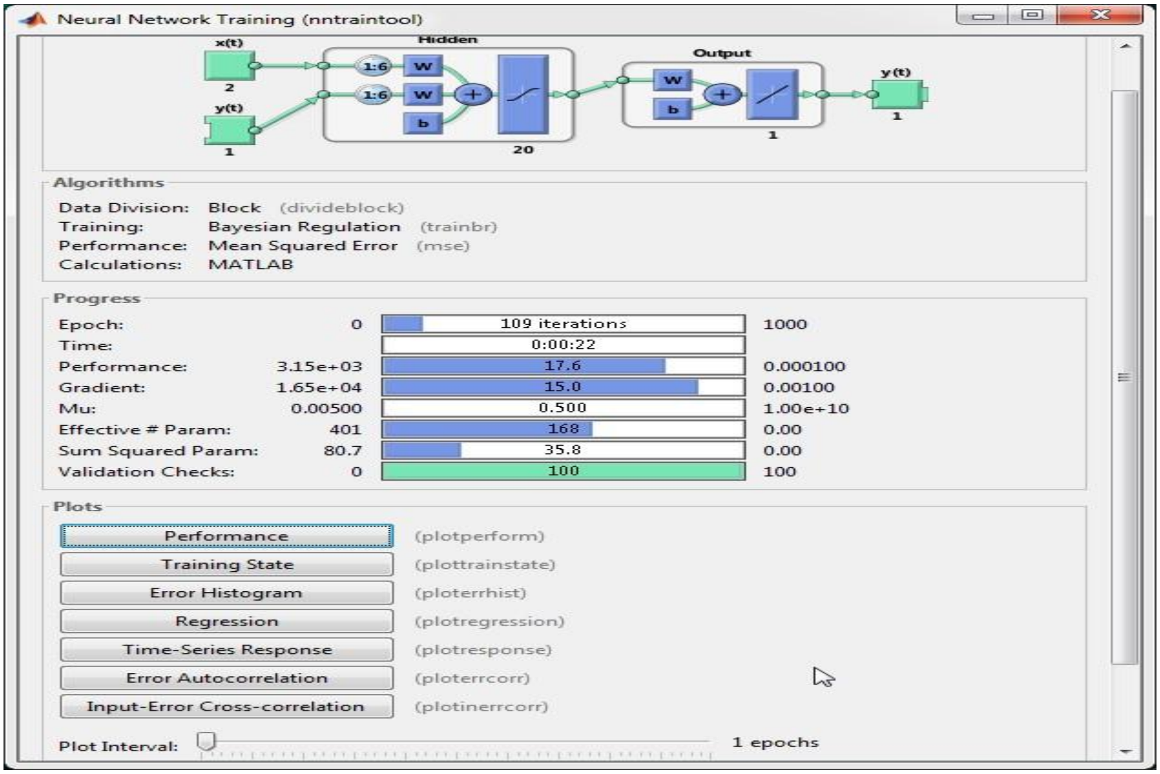


Figure 3.3: Display of a NARX training window, training terminated when validation error increased for 100 iterations which occurred at 109 epochs (MathWork, 2015a).

In each step of iterative process, the error signal activates a control mechanism which applies a sequence of corrective adjustments of the weights and biases of the neuron via the weight/bias learning functions. These corrective adjustments continued until the training data attained the desired mapping to obtain the target output as closely as possible. After a series of iterations (also called *training epochs*) the neural network was trained and the weights were saved.

Step 6: Testing the Performance of the network.

(v) Testing and validatingof the network. The network output, errors and overall performance were computed using the command-line:

pred = net(inputs, inputStates, layerStates);
errors = gsubtract(targets, pred);
performance = perform(net, targets, pred);

(vi)The performance training record was plotted and overfitting was potentially checked.

(vii) Retraining of the multiple ANN network to fit the given data set to avoid overfitting and improve accuracy. However, the complete MATLAB program implementation is shown in Appendix B3

Due to the seasonal and cyclical behavior of the peak load demand data collected from TCN only linear regression model cannot accurately describe and predict the data.Therefore, there is a need to improve the accuracyof forecast model by combined CI method (ANN). Thus, this justify the reason for combine SARIMA and NARX neural network model as a solution used in this work. The detail of the model results is shown in chapter four. However, other research hasshown many advantages of using NARX neural network for nonlinear time series prediction. Among them is that NARX has better ability to generalize periodic behavior using it large memory effectas time-lagged inputs associated with dynamic network. In addition to long-term dependencies effects, it storesoutput feedback. Research by Lin*et al.*, (1998) also proven that the feedback connections in NARX propagate the gradient in a more efficient manner, making it a better choice for time-series forecasting.

Step 7: Designing an Improve SARIMA based on NARX NeuralNetwork Model

The improve SARIMA based on NARX model was achieved by concatenation of the SARIMA and NARX models. The concatenation process is achieved by using the forecasted output from SARIMA as input into the NARX model. Thus, the model will retrain the data to mimic the trend and forecast ahead using sigmoid activation function with LM algorithm or sigmoid activation function with Bayesian regularization (Br) algorithm at each training performance goal is set at 1000 epoch. The training will be performed for ten iterations and the result is average. The details of the complete MATLAB program used for implementation by simulation are shown in Appendix B3. Then developed model used the metrological data and peak load data of TCN 330kV power grid of Nigeria from Jan. 1st, 2006 to 31st Dec. 2014to training and forecasting while the model will be validated by comparing the day ahead daily peak forecast result obtained in year 2015with the actual measurement of daily peak load demand of year 2015 obtained from TCN. After the prediction, the following performance evaluation were carryout on the results.

(1) Evaluation of the proposed model simulation result with the real measured data of year 2015 obtained from the TCN using standard performance indices RMSE and MAPE.
(2) Evaluate the proposed model using training performance plots.
(3) Plot predicted output and actual data for comparison

3.5.3 Evaluation of the Proposed Model Forecast Performance

The measure of performance used in this work are Mean Absolute Percentage Error (MAPE). These are mathematically defined as:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|AE| \times 100\% \qquad (3.4)$$

where; $AE = \frac{L_i^{actual} - L_i^{forecast}}{L_i^{actual}}$, L_i^{actual} and $L_i^{forecast}$ are the actual load data and forecasted outputat i^{th} point respectively, and n is the total number of data points.

The proposed model was achieved by concatenating the SARIMA ability to capture cyclic trend and non-stationary characteristic of time series and very good learning capability and generalization performance of NARX neural network tune with chaos into a single unit.Figure 3.4shows the block diagram of the proposed model description. In the proposed model training, the error will be calculated in two instances: in open loop, the error corresponds to the fit error, i.e., the difference between the actual value and the output value of the open loop network for the validation set. This error is used to select the best network, which is the one that will be used for forecasting. Once the forecast is produced in closed-loop, the error is calculated between the actual value of the load, which is known to us because we are using past data for validation, and the closed-loop forecast output. This error is the error reported in the results as forecast error. Figure 3.5 shows the flow chart implementation of the proposed model which was later program in MATLAB software (R2015a i.e. version 8.3.0.532) environment.

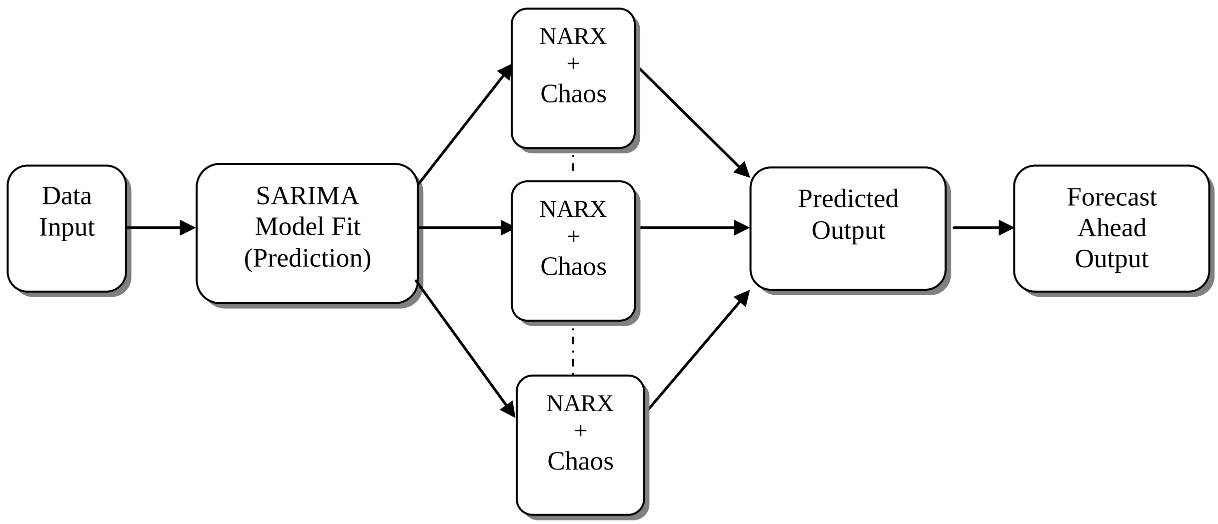


Figure 3.4: Block diagram of the Proposed Model Description

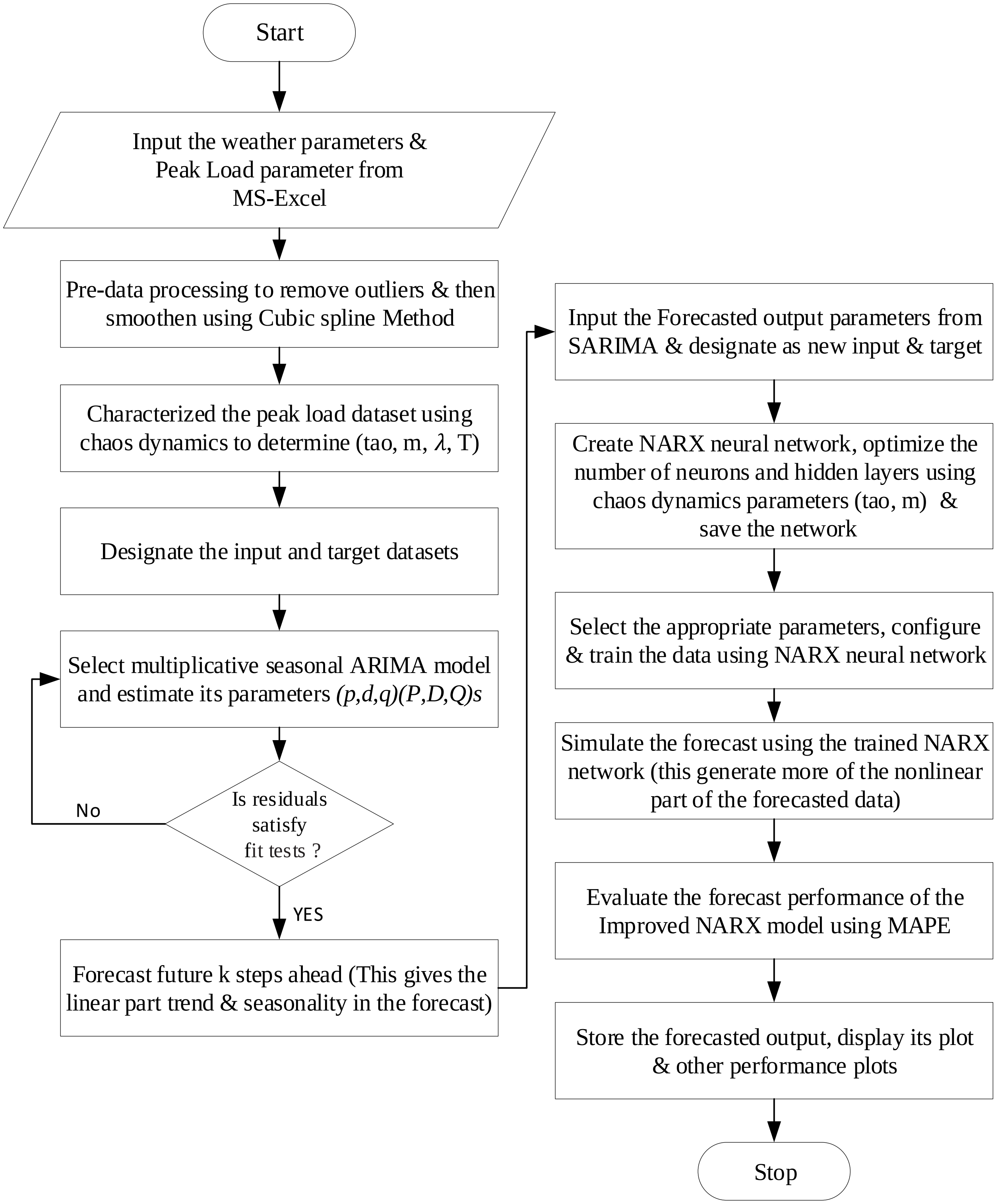


Figure 3.5: Flow Chart of Proposed Improve SARIMA Based on NARX Neural Network Model for Daily Short-term Daily Peak Demand of Nigeria Power System Grid

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the discussion of simulation results obtained from the implementation of the proposed model. Simulation was carried out according to the flow chart in Figure 3.4, for two short-term forecast cases (daily and weekly).The results are presented in form of tables, figures and graph/charts according to each sections outcome. Finally, the performance of the developed model was evaluated by comparing the prediction of simulation results and the real (actual) measured data obtained from TCN.

4.2 Results of the Characterization of Peak Load Data Using Nonlinear Analysis

This sectionpresents the results of the nonlinear analysis of the data where quantitative and qualitative tools were used to characterized the dynamics of the system by implementing the flow chart in Figure 3.1. Firstly, after smoothening, the descriptive statistics analysis of the data was carryout and the results presented as follows:

4.2.1 The Data Descriptive StatisticsAnalysis Results

The data variables used in in this work were analysed using descriptive statistics tools (central tendency and dispersion) and thesummary of results is shown in Table 4.1. Each variable has a total 3,650 data points (i.e. 365 per year), all leap years' data wereremovedfor equality. The results also show the distribution of the data after filtering all the outliers and the series equivalent signal to noise ratio.

Table 4.1: Descriptive statistics of the involved variables P, Peak load; T_{min}, Minin temperature; T_{max}, maximum temperature; WS, wind speed; SR, solar radiation, HR, Relative humidity

Variable	Mean (μ)	Standard deviation (σ)	Coefficient of variation (CV)	Signal-to-noise ratio (SNR)	Kurtosis	Skewness
P (MW)	3313.9	587.81	0.1774	5.6375	2.7128	- 0.1094
$\boldsymbol{T_{min}}$, (℃)	20.5850	3.7579	0.1824	5.4837	2.9324	- 0.6061
$\boldsymbol{T_{max}}$, (℃)	33.2100	3.6482	0.1099	9.1030	2.7294	- 0.0609
WS (km/hr)	3.8985	1.6140	0.4140	2.4155	3.8591	0.6583
SR ($MJ^{-1}m^{-2}day^{-1}$)	21.4658	4.0748	0.1898	5.2678	2.9568	- 0.4762
HR (%)	78.8702	11.2323	0.1424	7.0218	8.2497	-1.9660

4.2.2 Results of the Analysis of the Peak Load Demand Data using Time series plot and power spectrum (visualization of variables)

1. Time series plot

The daily peak load demand data of Nigeria from January 2006 to December 2015 were obtained from Transmission Company of Nigeria. The time series plot of data showed the behaviour of the grid in the last decade is given in Figure 4.1. It is obvious that the data is non-stationary and there is clear increasing nonlinear trend and the variance of the data is also not constant.The Figurealso shows that in the last decade, at peak generation, the loads demand on the grid systems exhibit some irregular cycles of oscillation with variable amplitudes and different frequencies. This indicate unpredictable behavior, subjectively this was taken as the first signature of chaos observed in this study. However, this observation is not sufficient to drawn general conclusions because a random data can also exhibit similar characteristics.

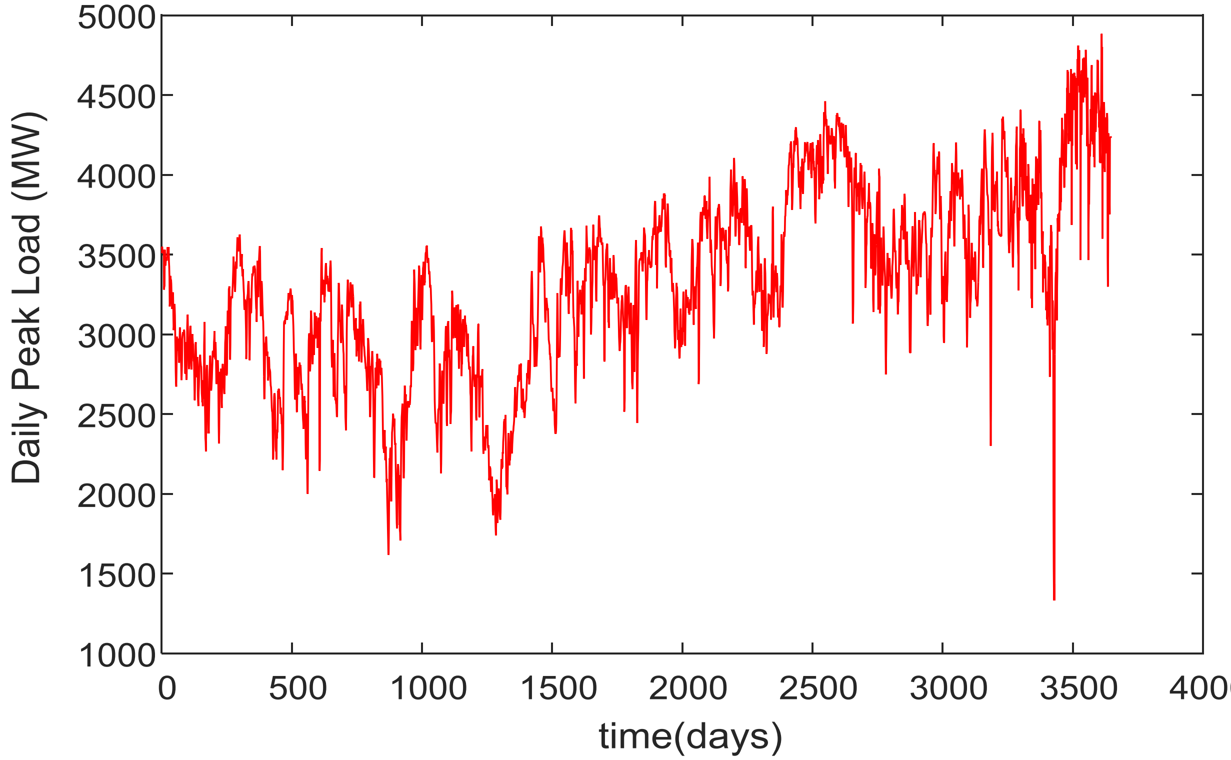
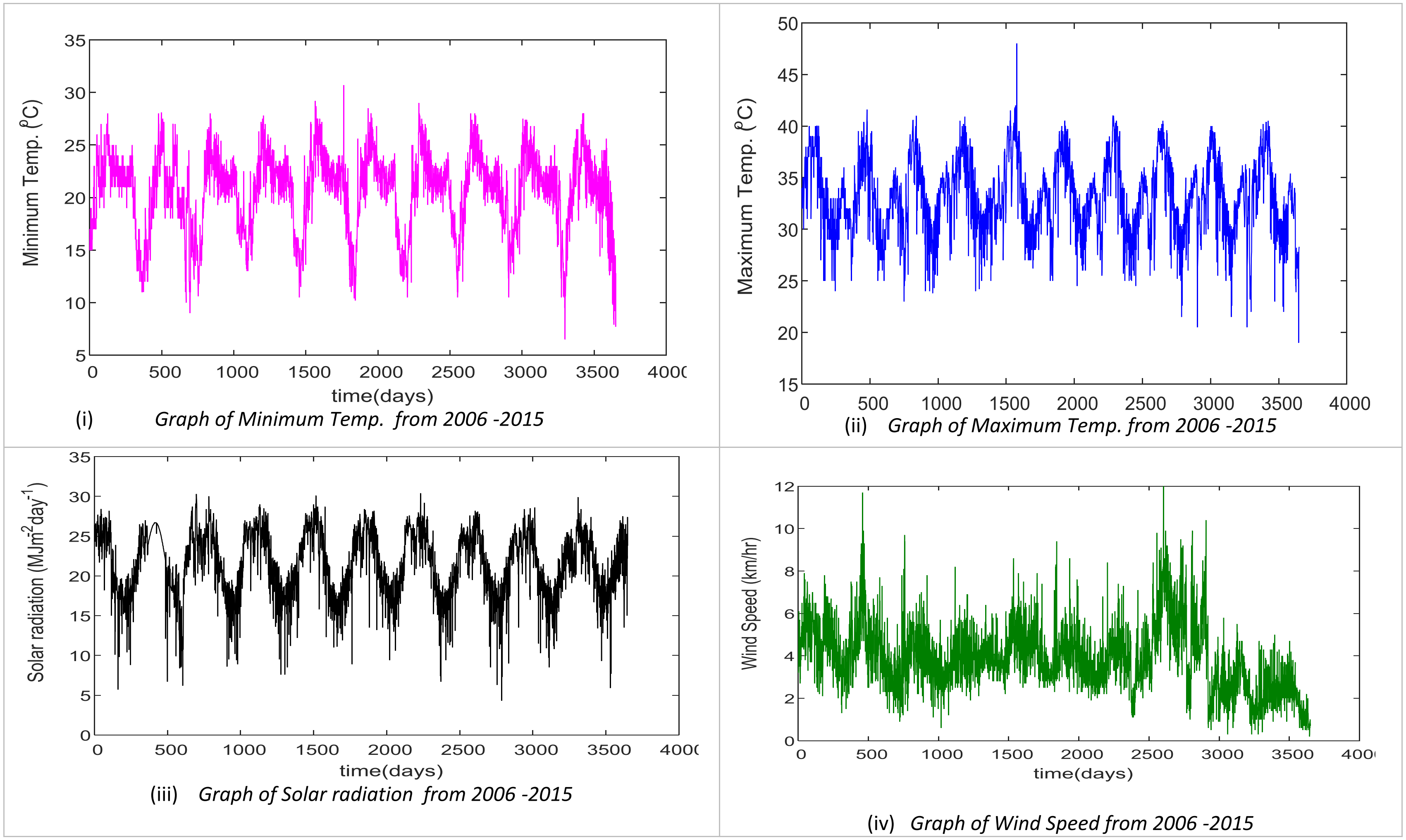


Figure 4.1: Time Series Plot of Nigeria Power Systems Grid (330kV Network) Daily Peak Load Demand from January 2006 to December 2015

Similarly, Figure 4.1, further shows clearly that there is significant similarity in load demand profile on the same day of each week. It is clear that there is seasonality in the load demand values of dry and wet seasons.These suggested that regular time series methods cannot be applied to this data without making transformation and differencing to remove unconstant variance and trend. In addition, the seasonality and trend ofmetrological variables was also verify by time series plot see Figure 4.2.

Plot of Time Series Analysis of Weather Variables used



(i) *Graph of Minimum Temp. from 2006 -2015*

(ii) *Graph of Maximum Temp. from 2006 -2015*

(iii) *Graph of Solar radiation from 2006 -2015*

(iv) *Graph of Wind Speed from 2006 -2015*

Figure 4.2: Time Series Plot of Daily Weather Parameters Data from 2006-2015 Obtained from NiMET

2. Result of the analysis using power spectrum

In order to further quantify the signature of chaos peak load time series, power spectrum tool for nonlinear dynamic analysis was applied to the data, using equation (2.1). The result of the analysis is presented in Figure 4.3. From the Figure it can it was observed thatthe attractors in the data series showswide broadband power spectra with random noise-like and a dominant periodicitiesfrequency of $0.001534Hz$ at peak.This means that spectrum shows a continuum of frequencies in their oscillations near zero. From all the attribute of the signal in the analysis result, it can be concluded that the peak load series data exhibit a noisy and chaotic behavior.

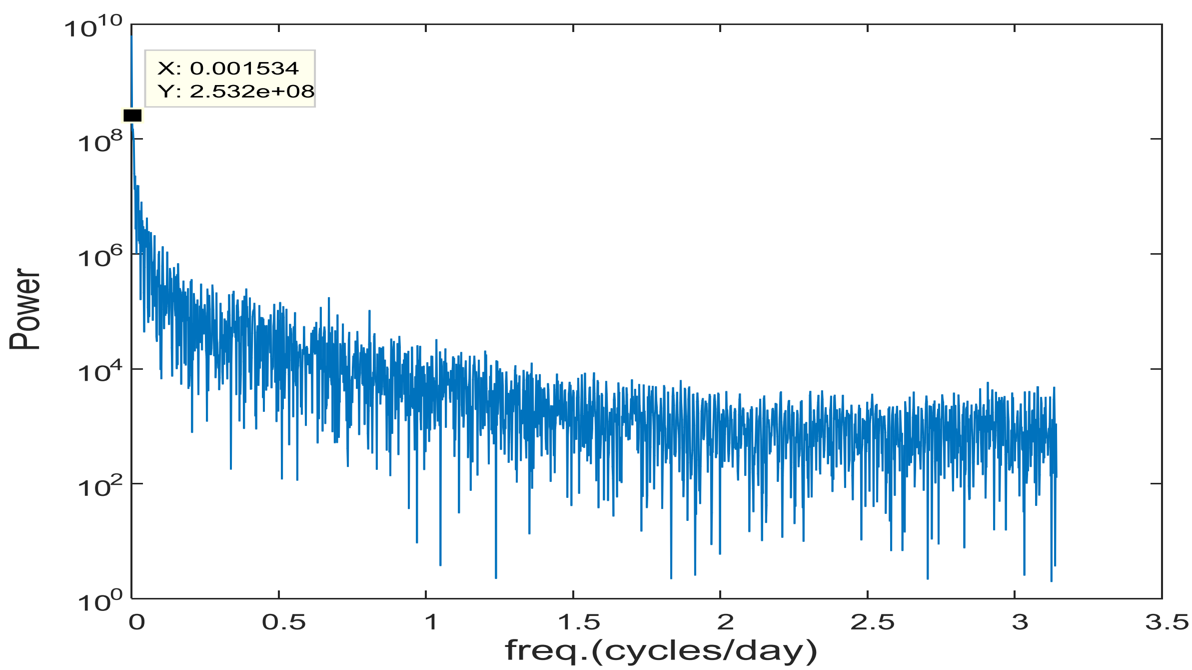


Figure 4.3: Power Spectrum Analysis Plot of the Peak Load Time Series

The data was further analysed using a quantitative analysis to confirm the signatures of chaos in the data, the result is presented in section 4.2.3.

4.2.3 Result of Phase Portrait Analysis

The attractor in the peak load time series data were plotted in state space using the reconstructed embedded space point $(x_{i+\tau})$of viewi.e., $m + \tau$and x_i as state variables. The result is presented in Figure 4.4. The Figure displayedthem-dimensional trajectories of the data attractor in state space which indicates the number of degree of freedom that describes the dynamics of the system.Also from the Figure,the data attractor points displayed, clearly shows a set of distinct shape entwined together forming spongy bird's nest-like structure which indicate a clear manifestation of chaotic behavior in the peak load time series as described in Table 2.2 (Ozer and Akin, 2005). Hence one can infer the presence of deterministic chaos in the peak load data recorded across the 330kV grid by TCN in this study.

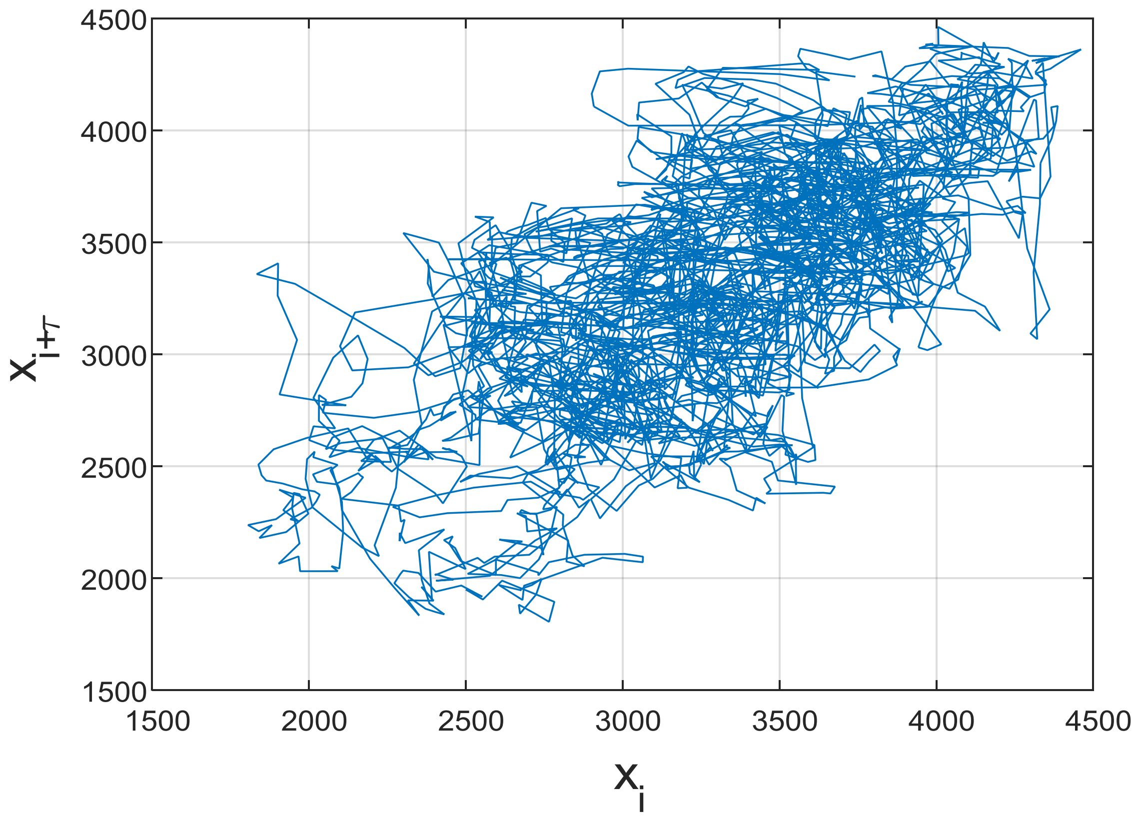


Figure 4.4: 2-D Plot of Phase Portrait of Peak Load Demand Time SeriesData.

4.2.4 Results of The Phase Space Reconstruction and Computation of Lyapunov Exponent

Before computing the Lyapunov exponents, phase space reconstruction was computed so as to draw out a multi-dimensional description of system in an embedded space called state space.This was achieved using the method of delay (MOD). The data point (attractor) in the daily peak load data were reconstructed in a m-dimensional phase space of delay coordinates by forming the vectors using time delay and embedding dimension. The results are presented as follows:

1. Results of computation of Timedelay

Figure 4.5 shows result of computation of time delay obtained to be$\tau = 27\ days$andFigure 4.6 present the embedding dimension resultsas $m = 8$ from the overall nonlinear analysis for ten (10) years' data. In practical terms, this implies that the delay time (or time lag) τ has a shortest time over which there are clearly measurable variations in the observable. For $m = 8$indicate the level of chaos and to accurately discribe the dynamics of the system in phase space requires 8 degrees of freedom.Similarly, in this work trial by error approach was avoided by using chaos theory to optimize the number of neurons in the 1st and 2nd hidden layers ($N_{H,1} and\ N_{H,2}$) by using the equation (2.25), the parameter of $\tau\ and\ m$ obtained were used for optimal selection of NARX structure hidden layers neurons which was later used in training and forecasting ahead. That is, $\boldsymbol{\tau} = 27\ days$correspond to 2number of tap delay lines (TLD) requires for output feedback memory. While 17 and 4 represent the number of neuronsrequire in the 1st and 2ndhidden layers to be used for the optimal configurstion of NARX neural network structure.

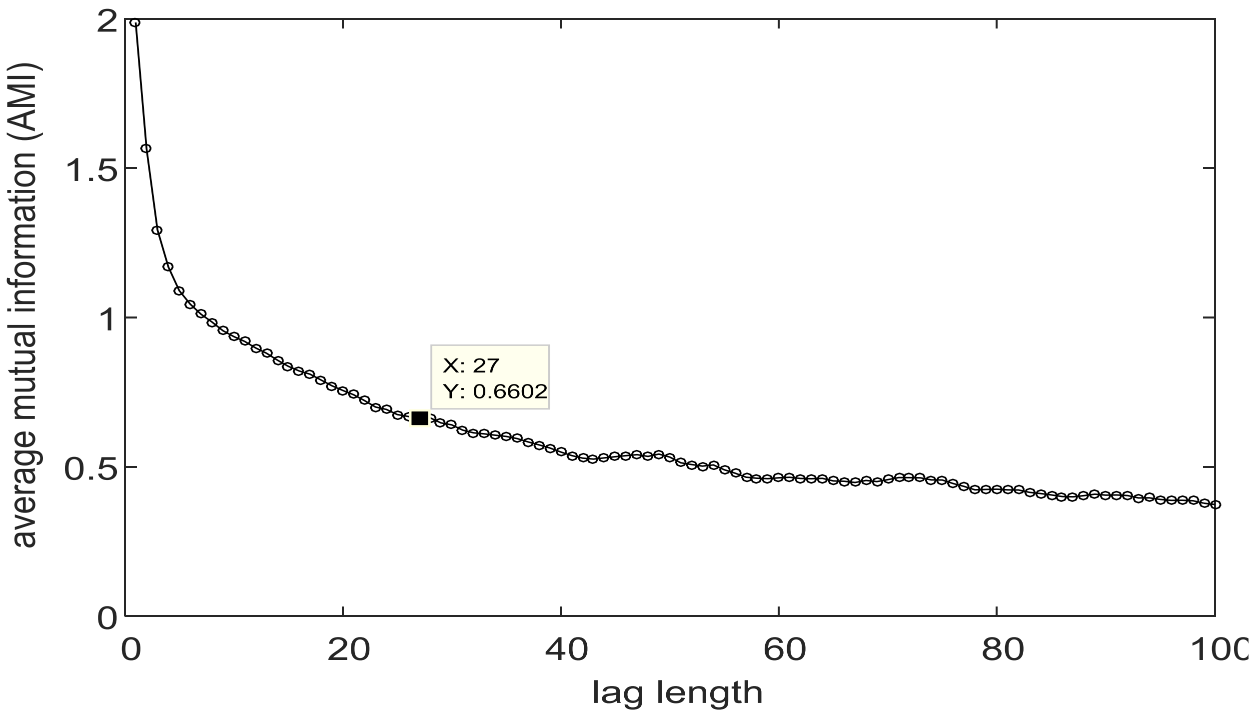


Figure 4.5: Estimation of Time Delay for daily Peak Load Prediction; Time Delay, $\tau = 27\ days$which correspond to 27-tap delay lines (TLD) for NARX Network Configuration

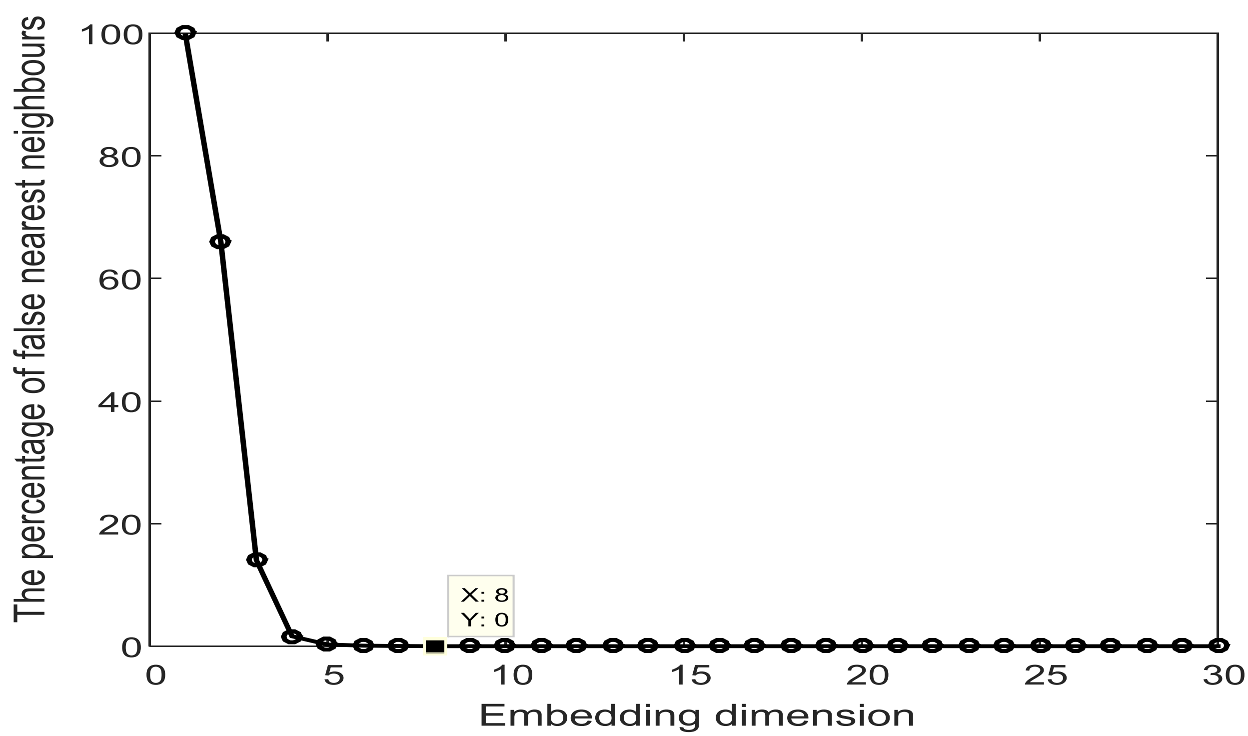


Figure 4.6: Estimation embedding dimension parameters for daily peak load prediction is $m = 8$, for NARX network hidden layer configuration is 17 neurons.

2. Results of computation of Lyapunov Exponents

In order to further confirm signature of chaos in the peak load data analyzed Lyapunov exponent method was applied to the data. The data was group into daily and yearly form and their nonlinear analysis were computed, the results is shown in Table 4.2 and 4.3 respectively. In Table 4.2, the value of Lyapunov exponent (λ)results was$0.009071 (bit\ /day)$. This meansλ $>$ 0,and when compared with standard given by Ozer and Akin, (2005). The result confirmed that the systemis chaoticat peak load. Thus, toaccurately model and predict the system requires nonlinear analysis tool to accurately model and forecast ahead. In summary, thevalue of time delay and embedding dimension recorded are 27 and 8. This implies that this system can be best modelby 8 independent coordinates (differential equations or degree of freedom) to fully describe systems dynamics.

Table 4.2: Result of Quantitative Analysis for Daily Peak Load Demand Data Nigeria (TCN)

Year	Delay Time τ (day)	Embedding dimension ($\boldsymbol{m}$)	Lyapunov Exponent λ (bits/day)	Predictability T (day)	Coefficient of variation (CV)
2006 – 2015	27	8	0.009071	123.5	0.1773

Also, average exponential divergence index of the attractor is plotted against time as seen in Figure 4.7, this shows the regression line (slope). The regression line (red) fitted to the curves has a slope equal to the theoretical value of λ (the largest Lyapunov exponent) with alow-variance (CV) of 0.1773.The time lag for predictability was also calculated as 124days. This estimated valueindicated that we can only have accurate forecast horizon of 124 days ahead for peak load demand operation and planning of Nigeria power grid. This justify the choice using short-term load forecasting model is this research as we can have approximately 18 weeks daily plan ahead. Technically, the system operators are expected to use this information to deploy their resources for generation scheduling and economic load dispatch on short-term horizon planning for efficient and reliable operations of the grid across Nigeria.The detailed yearly analysis is shown in Table 4.3.

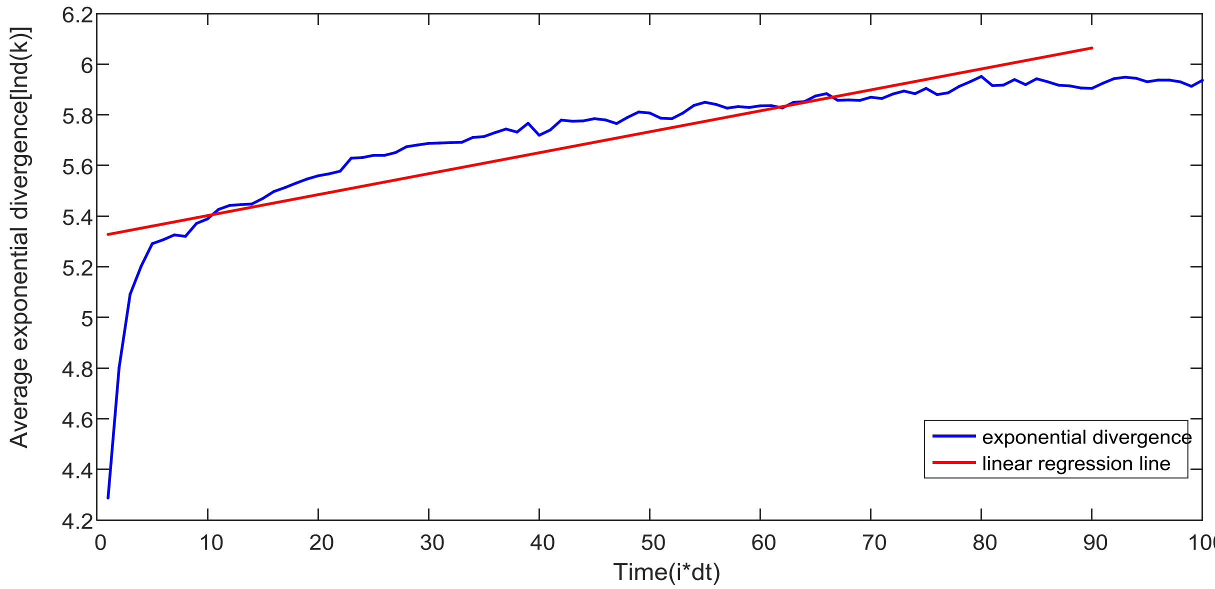


Figure 4.7: Lyapunov Exponent Computation of Peak Power Load Daily Data, $\lambda \cong 0.0091\ /day$

The Table 4.2, shows the results of the annual quantitative analysis (time delay, embedding dimension, Lyapunov exponent, and predictability values for average daily peak load data, as earlier stated the time delay and embedding dimension of the system were computed using autocorrelation function and the method of false nearest neighbor respectively as described in section 2.2.3.6 and used to construct the phase space. The year 2006, 2007 and 2008 data had a time delay and embedding dimension of (7, 6), (5, 11) and (7, 4) respectively. This implies that these systems need 6, 11 and 4 independent coordinates (differential equations or degree of freedom) to fully describe its dynamics. The 2009 peak load data on the other hand, have a delay time of 13 days with an embedding dimension of zero (0). This means that the degree of freedom to describe the dynamics of the system is infinite which can be attributed to large missing data in this year and causes the system dynamics to tend to deterministic chaos. The dynamics of the systems in other years i.e. 2010 to 2015 can be described using their respective degree of freedom. In addition, each embedding dimension value could also be used to described the structure of artificial neural networks (ANN) used load forecasting. Moreover, from the results displayed in Table 4.3 the peak load for 2009 exhibits a quasi-periodic behavior (near transition to chaos) having a positive Lyapunov exponent of 0.00/day, while peak load for 2006 to 2008 and 2010 to 2015 are unstable and chaotic having a Lyapunov exponent (λ) ranges from 0.00167/day to 0.01462/day.

Table 4.3: Results of quantitative analysis of the daily peak load demand data for year: 2006 - 2015

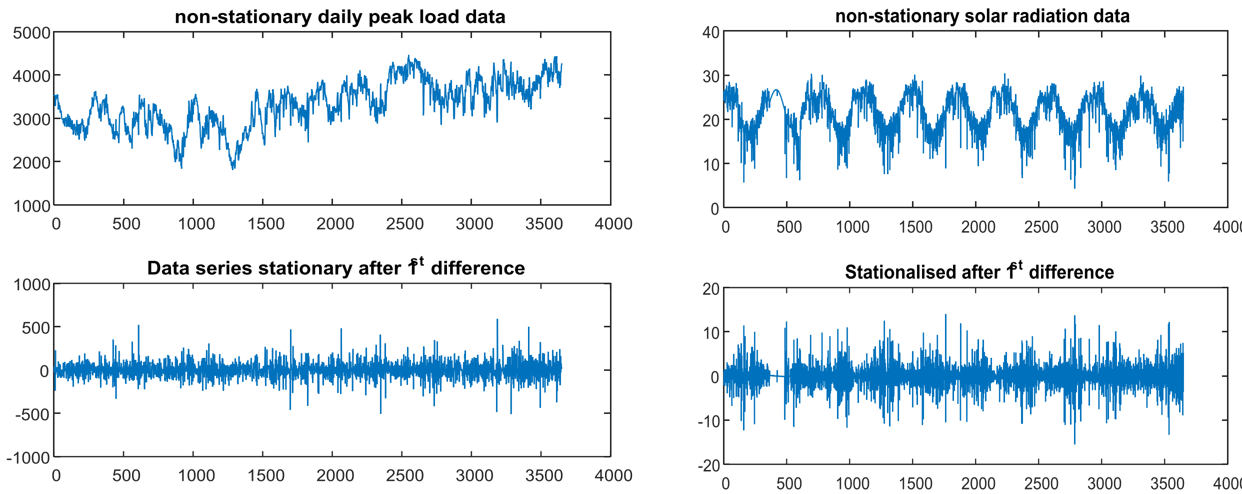
Year	Delay Time τ (day)	Embedding dimension ($\boldsymbol{m}$)	Lyapunov Exponent λ (bits/day)	Predictability **T** (day)
2006	7	6	0.0146	68.37
2007	5	11	0.0114	87.57
2008	7	4	0.0082	121.34
2009	**13**	**0**	**0.0000**	**-**
2010	11	7	0.0035	287.43
2011	9	10	0.0065	153.32
2012	5	5	0.0134	74.44
2013	8	6	0.0017	596.96
2014	10	9	0.0059	167.58
2015	6	4	0.0070	142.59

4.3 Simulation Results for Proposed Improve SARIMA Based NARX Network Model

Two set of simulation were run in MATLAB, one set for multiplicative SARIMA model which is used for performance comparison, as a basis to establish the limitations of using only conventional statistical model (SARIMA) for a chaotic time series short-term forecasting. This model cannot accurately describe and predict the data well as it results in large error due to the nature load demand data which shows a seasonal and cyclical behavior. And a second set is for a NARX neural network as described in section 3.5.2. Both set were run with the same number of iteration and the output of the first model was feed as an input into the second model by concatenation method with the structure optimized with tenets of chaos theory parameters (time delay and embedding dimension) obtained in section 4.2.3. The best network based on MAPE fit of the existing daily peak load data from TCN was selected in order to provide forecast. The results obtained are evaluated by comparison with the real(observed) data obtained from TCN.

4.3.1 Simulation Results for Multiplicative SARIMA Model

The Box-Jenkin methodology of formulation of multiplicative SARIMA model for Identification, Estimation, Diagnostic checking and Forecasting was run in MATLAB. In model identification stage, the work test for stationarity of the data using KPSS unit root test. The results of KPSS unit root test demonstrates that the original sequence of peak load and solar radiation data are not stationary, that is their null hypothesis H0 results shows that$H0 = 1$ and $pValue = 1.0000$ values which does not satisfied the null hypothesis condition of stationarity (this means that the data have trend and periodic and its mean and variance varies i.e. not constant) so it needs to be difference to stabilize. Therefore, non-stationarity in variance is corrected through square root transformation and non-stationarity in mean is corrected through appropriate first-order differencing(i.e. d = 1)and seasonal difference of order 1 (i.e. D = 1)which are sufficient to achieve stationary in mean and variance. The results are shown in Figure 4.8 (a) and Figure 4.8 (b), in both Figures, the null hypothesis results are$H0 = 0 \; and \; pValue = 0.1000$meaning that the data is stationary.



(a) (b)

Figure. 4.8 (a) and (b): Stationarity Test Result for Peak Load and Solar Radiation Data using KPSS Test

Then, the next step is to identify the values of*(p, q)* and *(P, Q)* parameterfor model structure. Bayesian Information Criterion (BIC) was employed to compute the best parameter values for SARIMA model and the SARIMA model for each variable are calculated and chosen based on minimum BIC values and the results is presented in Table 4.4.In this model, the non-seasonal autoregressive non-seasonal moving average items are AR(2) and MA(4), and the seasonal autoregressive and seasonal moving average items are SAR(1) and SMA (4). The detail is as follow:

Table 4.4: The SARIMA Model Structure

Variable	**Seasonal ARIMA model (p,d,q)(P,D,Q)**	**Bayesian Information Criterion (BIC) *1.0e+04**
Peak load (MW)	$(2,1,4)(2,1,4)^{365}$	4.2178
Solar radiation ($MJ^{-1}m^{-2}day^{-1}$)	$(2,1,1)(2,1,1)^{365}$	1.6116
Minimum Temperature(℃)	$(4,0,3)(4,0,3)^{365}$	1.4054
Maximum temperature(℃)	$(3,0,3)(3,0,3)^{365}$	1.5046

The best suitable model for peak load demand is $\text{SARIMA}(2,1,4)(2,1,4)^{365}$as with the lowest BIC.We then proceed to the next stage of the Box-Jenkins approach, where the parameter coefficients of the model were estimated using the maximum likelihood function and conditional least squares methods and the results found is presented in Table 4.5. However, results on Tableshows that all the coefficients of the parameters are statistically significant on 5% level of significance.

Table 4.5: Result of SARIMAModelEstimation using Least Squares Method

Variable Model	Variable Parameter	Coefficient	Std. Error	t-statistic	Prob.
Peak Load (2, 1, 4)(2, 1, 4)	AR{2}	0.03223	845.611	3.8114e-05	0.0000
	SAR{2}	0.03241	835.611	3.878e-05	0.0000
	MA{4}	-0.09436	0.06704	-1.40752	0.0000
	SMA{4}	-0.07431	0.04455	-1.66801	0.0000
Solar radiation (2, 1, 1)(2, 1, 1)	AR{2}	0.86952	0.02149	40.46161	0.0000
	SAR{2}	0.58177	0.00719	8.09133	0.0001
	MA{1}	-0.93822	0.01775	-5.28574	0.0000
	SMA{1}	-0.35164	0.05382	-6.53363	0.0000
Maximum Temperature (4, 0, 3)(4, 0, 3)	AR{4}	-0.62791	0.07945	-7.90320	0.0000
	SAR{4}	-0.68960	0.07945	-8.67967	0.0000
	MA{3}	0.38352	0.28511	1.34516	0.0000
	SMA{3}	0.85085	0.08429	10.09431	0.0000
Minimum Temperature (2, 0, 2)(2, 0, 2)	AR{2}	-0.94353	0.21532	-4.38198	0.0000
	SAR{2}	-0.78694	0.18351	-4.28827	0.0000
	MA{2}	0.76074	0.25607	2.97091	0.0000
	SMA{2}	0.62144	0.27681	2.24501	0.0001

Diagnostic checking is done on the residuals using the goodness-of -fit test. These tests were performed to examined how the estimated samples fitted the model residuals and Figure 4.9 (a) and 4.9 (b) showed the autocorrelation plots and partial autocorrelation plots for the residuals. They show that there is still quite a lot of information left in the residuals, especially periodic data as few of the correlogram lines (ACF and PACF) are not within 5% level of significant as it decays at lags 20.This indicate some nonlinear component presents in the residual.

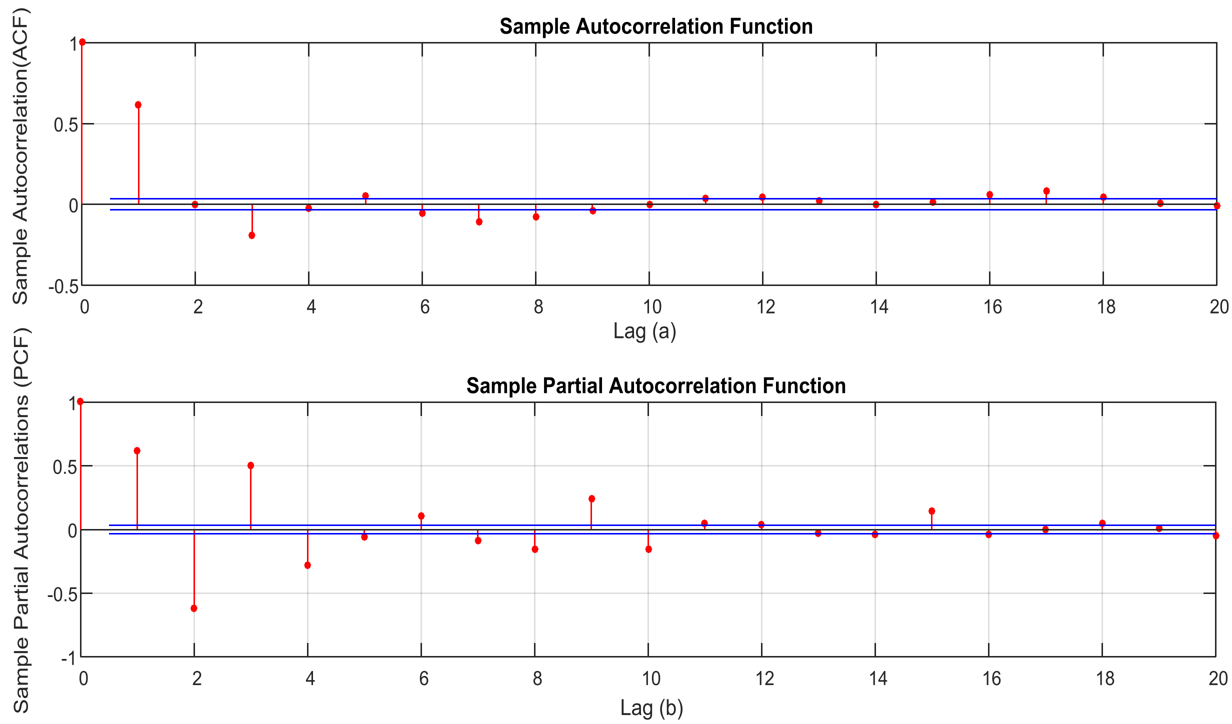


Figure 4.9:(a): Error Autocorrelation of the residuals where autocorrelation is observed with 365 day time lag; (b):Partial autocorrelation of the residuals of forecasted daily peak load

After applying the above steps, the statistical model selected yield residuals that does not fullysatisfy the normal distribution, because of presence of outlier having mean and no autocorrelation that are non-zero. Based on these assumptions, the residuals are examined for normality test to find the closeness of white noise and the results is shown in Figure 4.10 (a) and 4.10 (b)using Gaussian normal distribution an QQ-statistic plot. This indicates that the good fit for white noise situation. With Box-Ljung Q-test statistic result having a probability of 0.999,greater than 0.05 significant confident level, we may conclude that the selected seasonal autoregressive integrated moving average model is an adequate model for forecasting of daily peak load demand time series of Nigeria power system grid.

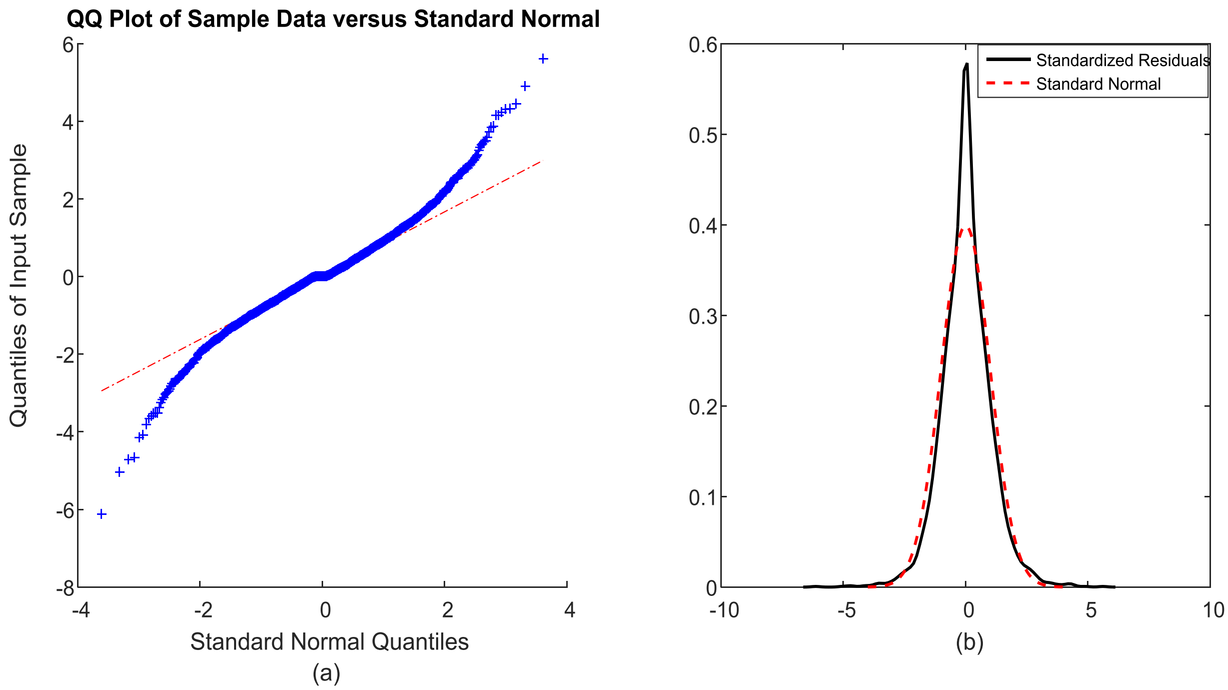


Figure 4.10: Results of residuals analysis (a): QQ plot of peak load data indicating the normal distribution (linear line) daily average components's distribution; (b):QQ Plot peak load data residual indicating the fitness of residual's to the normal distribution.(Note: QQ plots, ACF and PCF results for metrological variables are not included in this report).

4.3.1.1 Forecasting 365 Days Ahead Using SARIMA

The fitted model results (2, 1, 4)(2, 1,4)$_{365}$in Table 4.1 was appliedfor short-term ahead forecasting of daily peak load demand of Nigeria power system grid in megawatt (MW) from January 1st to December 31st, 2015 and the result is presented in Figure 4.11. The accuracy of the forecast is compared with the real data from TCN using the MAPE in equation (3.4). From the result, it shows that the model fit better to the observed sample data within 150 days (January – May) of prediction which is within the boundaryof time lag for accurate predictability given by Lyapunov exponent in subsection 4.2.3 (2).

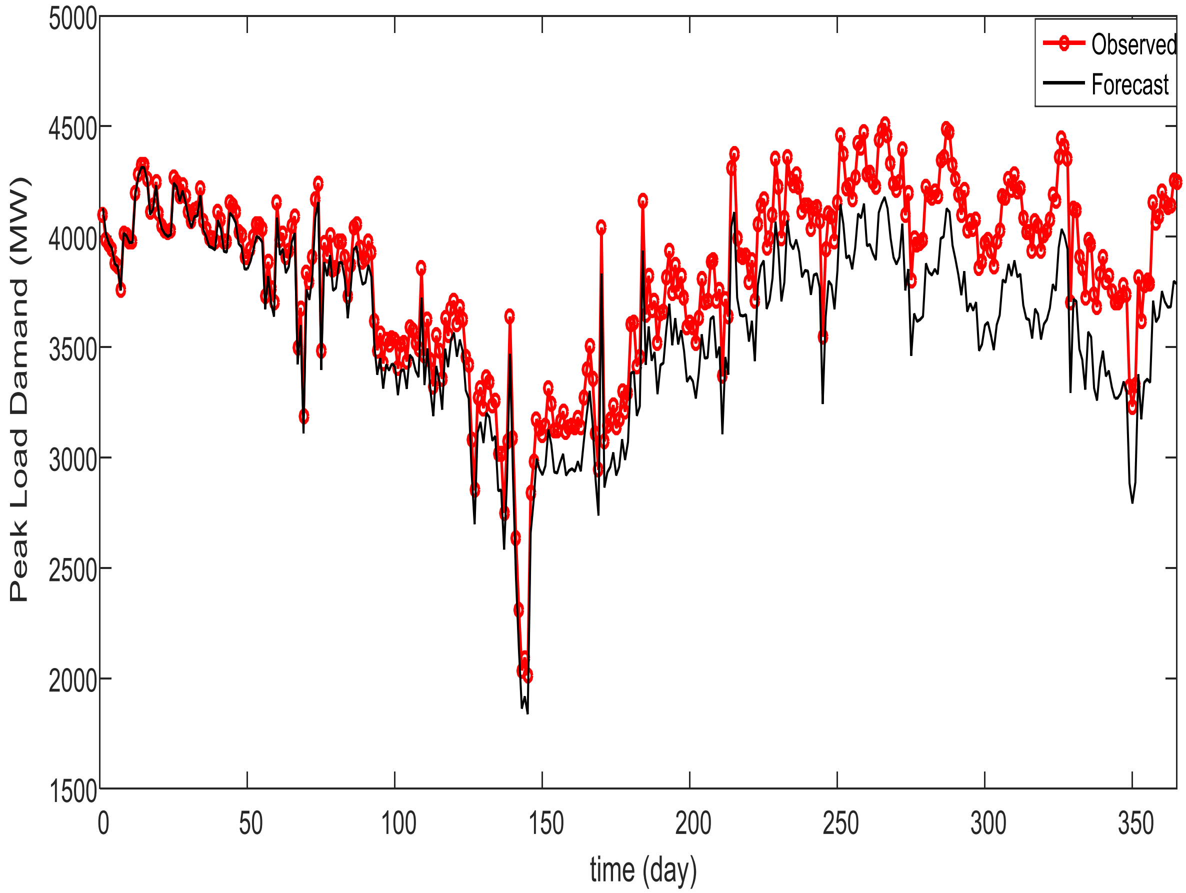


Figure 4.11: Time Series Plot of Actual (Observed) and Forecasted 365 Days Ahead Daily Peak Load Demand in 2015 using SARIMAModel.

The accuracy of the forecasts was checked with both in-sample and out-of-sample tests. In-sample test is simply the accuracy of the fit model, while out-of-sample test is real-time forecast for the future using the equation (3.3). The prediction error between the actual data and the forecast gives a correlation regression of 67.2%and MAPE of 6.6402%respectively. However, SARIMA models have the drawback that they can only produce linear forecasts. In the next section, the work will focus on increasing the accuracy of short term forecasting by using a nonlinear model - the neural network.In the next section, thework will describe the detail proposed solution using neural networks to generate a 365-days ahead forecast. As will be seen, NARX neural network address the shortcomings of the statistical approach.

4.3.2 Simulation Results for Proposed NARX Network for Daily Peak Load Demand of Nigeria Power System Grid

In this section, the work modeled and forecast the daily peak demand for Nigeria power system grid for generation scheduling and load dispatch of 365 days ahead, using the NARX neural network model to improve on the earlier forecast output from SARIMA model. The NARX neural network was trained in open-loop using fitted outputpeak load data from SARIMA as input (target) while solar radiation, minimum and maximum temperature were used as exogenous variables. Then,forecasting was done by running the simulation in a closed-loop using the calculated load as the input for the next step. Ten simulations were run using both Levenberg-Marquardt and Bayesian regularization training algorithm and Log-sigmoid activation function. And both sets were run with the same dataset and the same number of iterations on the following variables:

i. Number of data groups – 3 different data periods were used (input, target and testing data)

ii. Number of neural networks - 3 different runs for each iteration in order to separate the outcome from the initial conditions,

iii. Number of neurons in the hidden layer selected based on $(N_{H,1} = 2m + 1)$ - simulations were run from 15 to 19 neurons in increments of 2 with tap delay line ranges from 26 to 28.

4.3.2.1 Analyze NAARX Neural Network Performance After Training using Bayesian Regularization Training Algorithm (Br).

We used the network configuration and parameters described in section 4.3.2, and implementing the methodology in section 3.5.2 (B), thesimulations were run in MATLABby varying the parameters.When the training was completed, the network performance was checked to determine if any changes needed to be made to the training process, the network architecture, or the data sets. The results in Table 4.6, as follows: the results of the simulations for a 365 days ahead peak load forecast and the corresponding error performance comparison between the different neuron and tap delay line configuration. The NARX forecast was generated in closed-loop, i.e., the network was trained in open-loop by using known values of the load; then, the first day peak load demand forecast value is calculated with the trained network, and that value is fed back to the input in order to obtain the second day value recursively, and so on. An optimal structure of the network was selected with it 1st and 2nd hidden layers neurons and number of tapped delayed memory lines determined using the tents of chaos theory (Taken embedding theorem)results presentedin Figure 4.5 and 4.6. The Mean Absolute Percent Error (MAPE) in equation (3.4) was used as the performance measure for this work.

The simulation was done for training the model using Log-sigmoid activation function (LogSig).As seen on the results in Table 4.6, after 10 iterations, the best configuration of the network structure performance was 19 – 4 – 1 with 27- tapped delay memory lines; which produced smallest Mean Absolute Percent Error (MAPE) of 4.0420% and correlation coefficient 95.98% which shows a great improvement accuracy offered by the NARX network as compared to SARIMA model. Figure 4.12 shows the forecast comparison for each of the methods used.

Table 4.6: Forecasting Performance Evaluation Result of Develop SARIMA-NARX Neural Network Model using Bayesian Regularization (‘*trainbr*’) Training Algorithm with different No. of neurons and TDL Configuration

m	$N_{H,1}$ =2m+1	$N_{H,2} = round(\sqrt{N_{H,1}})$	No. of Time delay lines (TDL)					
			26		27		28	
			MAPE (%)	**R (%)**	**MAPE** (%)	**R (%)**	**MAPE** (%)	**R (%)**
7	15	4	4.3461	0.9232	4.31432	0.9516	4.2025	0.9497
8	17	4	4.2127	0.9497	4.1153	0.9419	4.1021	0.9521
9	19	4	4.1092	0.9261	**4.0420**	**0.9598**	4.0912	0.9513

*** smaller error (MAPE) means best forecast, accuracy and higher error means poor forecast.*

From the Table, it is clear that using the proposed method with Br instead of single SARIMA model, the MAPE are reduced with the error reduction percentage of 2.5998%.

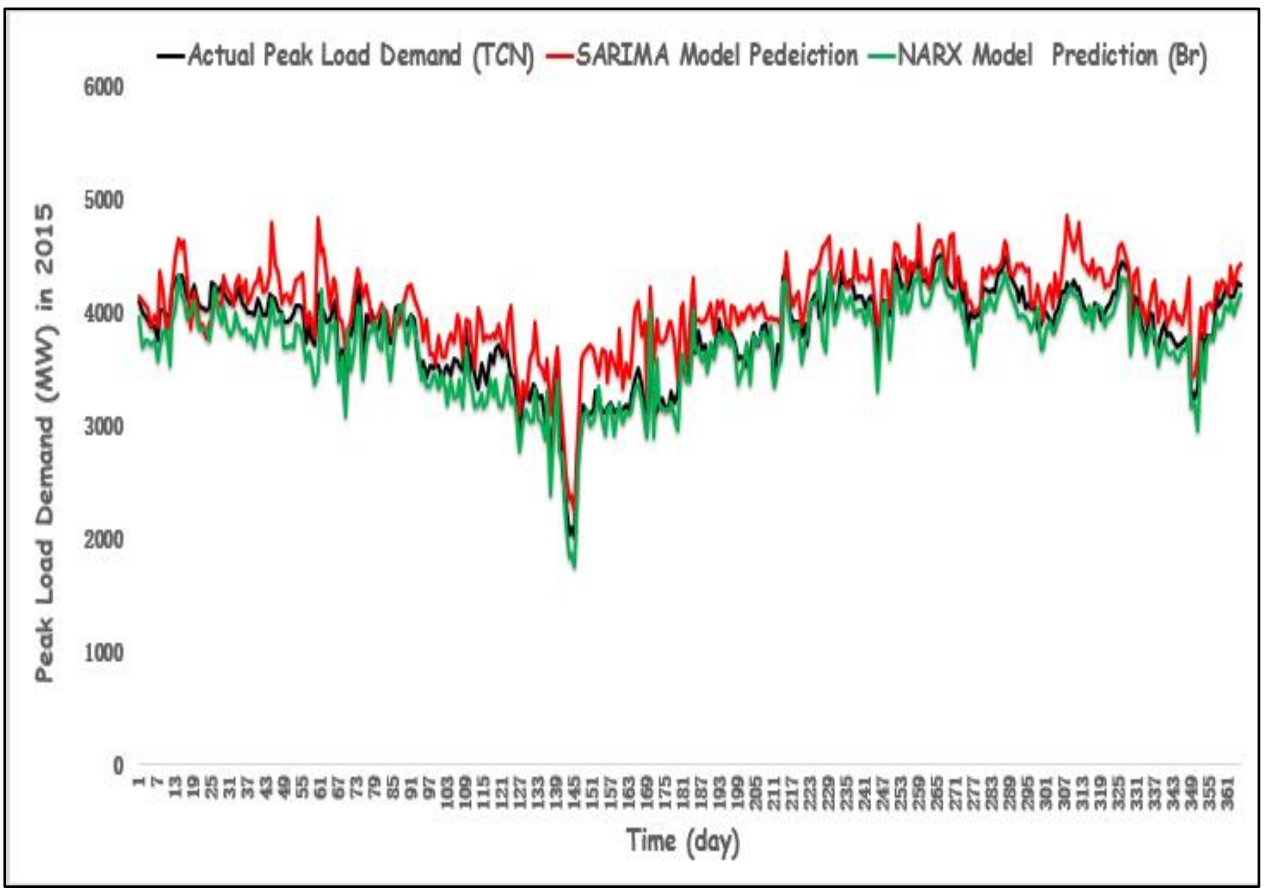


Figure 4.12: Time Series Plot of Actual Data, SARIMA and NARX (Br) Model Results for Forecasting 365-days of Peak Load Demand Ahead in2015.

Also, Figure 4.13 (a) shows the residual error autocorrelation of the forecast output result which has more spike above the acceptable confidence limit of ±0.1. The regression on the error between forecast and actual is a measure of performance, and it is plotted in Figure 4.13 (b). This indicate that more training algorithm is require more precision at reduced error.The training, by means of a Bayesian regularization backpropagation training algorithm converged after fewer than 32 epochs and it showed stability (no increase after converging) and no overshoot (no increase before converging), as shown in Figure 4.13 (c).

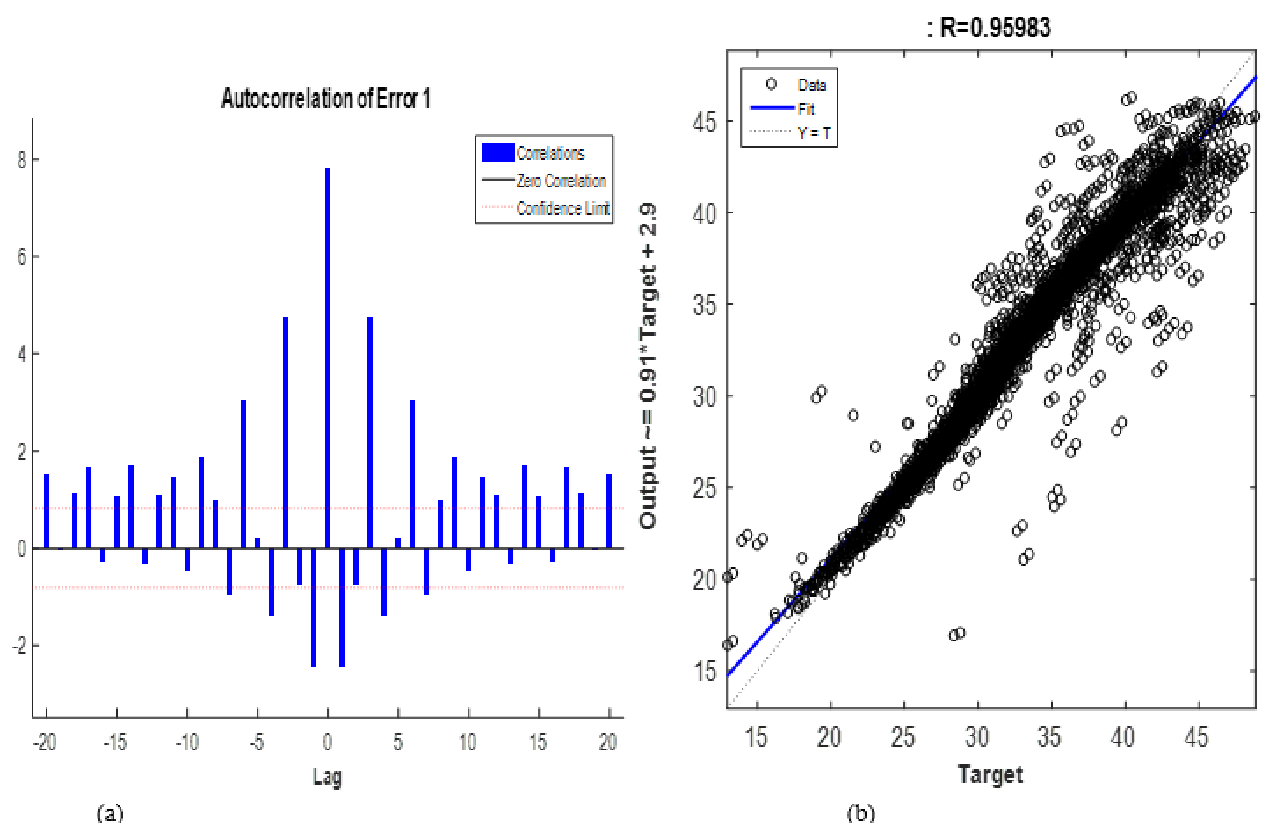


Figure 4.13: (a) Autocorrelation of Residual Error at Lag 20 and (b) Regression of Output-target and Forecast series for 365-days Ahead Prediction of Peak Load Demand using Br Training Algorithm.

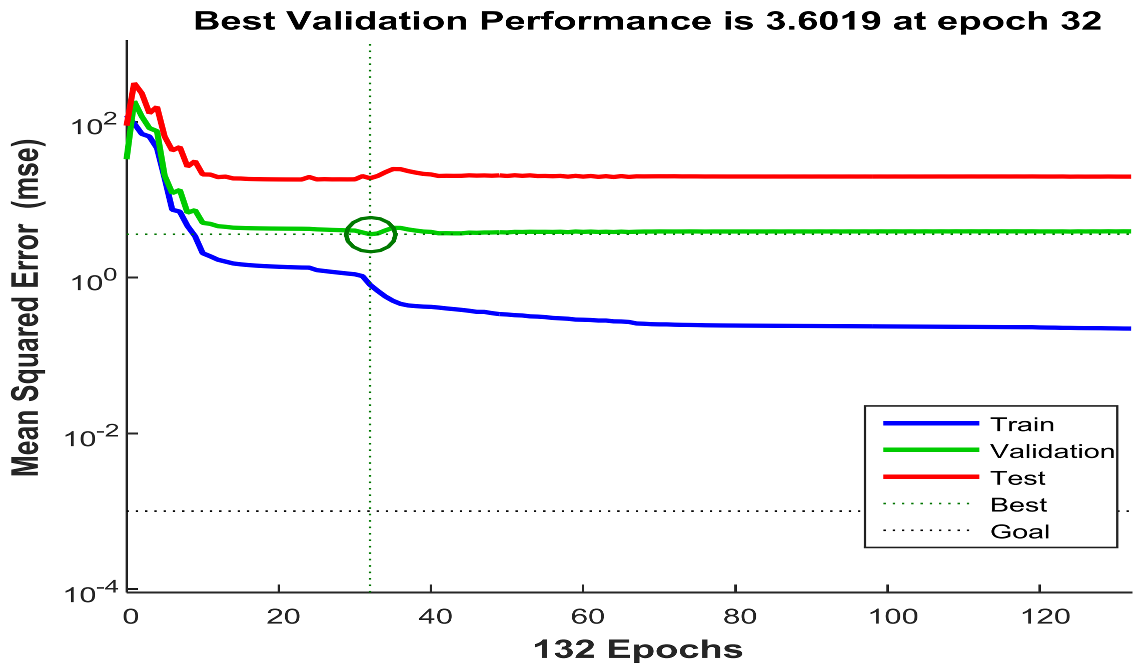


Figure 4.13 (c): Br Training Performance Plot; Best Validation Recorded was 3.6019 at Epoch 32.

4.3.2.2 Simulation Results for NARX Network using Levenberg-Marquardt Training Algorithm (LM).

Similarly, Table 4.7shows the same simulation but using the Levenberg-Marquardt training technique. In this case, the activation function improves in performance with theLM training algorithm, resulting in an improved error. The model simulations were run using LM training algorithm under many different conditions and configurations which includes: 15-4-1 with 26-TDL, 15-4-1 with 27-TDL, and 15-4-1 with 28-TDL. Others configuration was run with number of neurons in the 1sthidden layer increases by 2 and number of delayed inputs and feedback simulations increases by 1.A 28- tapped delay with 17- 4 -1 for inputs worked best, given anoptimal model with lowest forecast error ofMAPE of 2.4102%. Table 4.7 shows the detail of each training performance.

Table 4.7: Forecasting Performance Evaluation Result of An Improve SARIMA-NARX Neural Network Model using Levenberg-Marquardt ('trainlm') Training Algorithm with different No. of neurons and TDL Configuration

m	$N_{H,1} =$ 2m+1	$N_{H,2} = round(\sqrt{N_{H,1}})$	**No. of Time delay lines (TDL)**					
			26		**27**		**28**	
			MAPE (%)	**R (%)**	**MAPE** (%)	**R (%)**	**MAPE** (%)	**R (%)**
7	15	4	2.4461	0.9178	2.4392	0.9511	2.4970	0.9398
8	17	4	2.4227	0.9297	2.4194	0.9611	**2.4102**	**0.9659**
9	19	4	2.4192	0.9591	2.4528	0.9665	2.4861	0.9622

*** smaller error (MAPE) means best forecast, accuracy and higher error means poor forecast*

This means that, the 17-4-1 optimal NARX structure that have a better prediction in term of lowest error have 17 neurons, 4- neurons and 1-neuron used in the first hidden, second hidden and output layers respectively. As shown in Table 4.7, using the SARIMA-NARX with LM training algorithm instead of SARIMA-NARX with Br, the MAPE reduced with reduction percentage of 1.6318% given a better samples correlation coefficient of 96.59% at 365 days ahead forecast. However, Figure 4.13 shows the time series plot of the forecast tracking for each of the methods used in comparison with actual (observed) data obtained from TCN. It is evidenced from the figure that the proposed model trained with LM follow the actual load data more closely than single SARIMA.

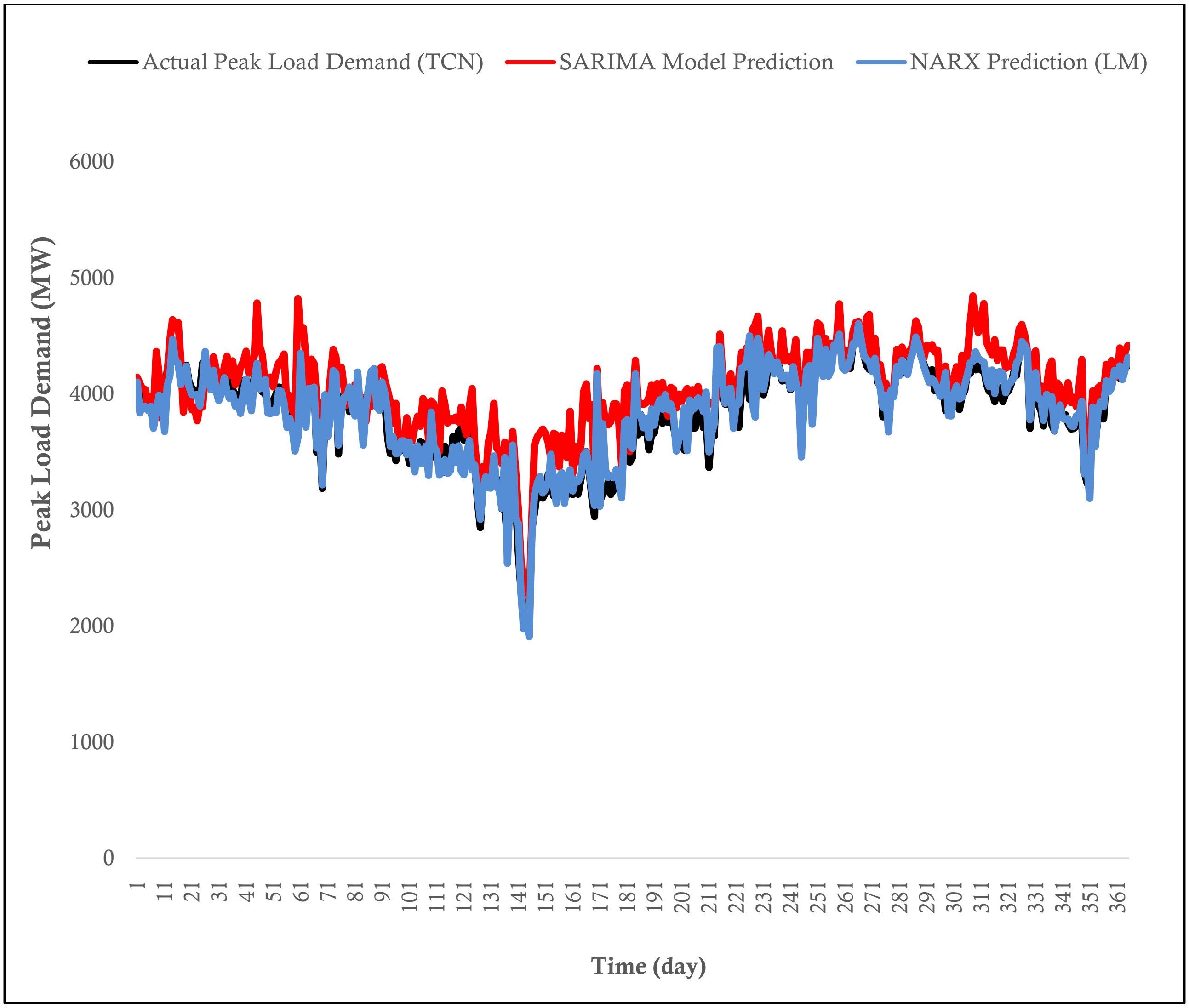
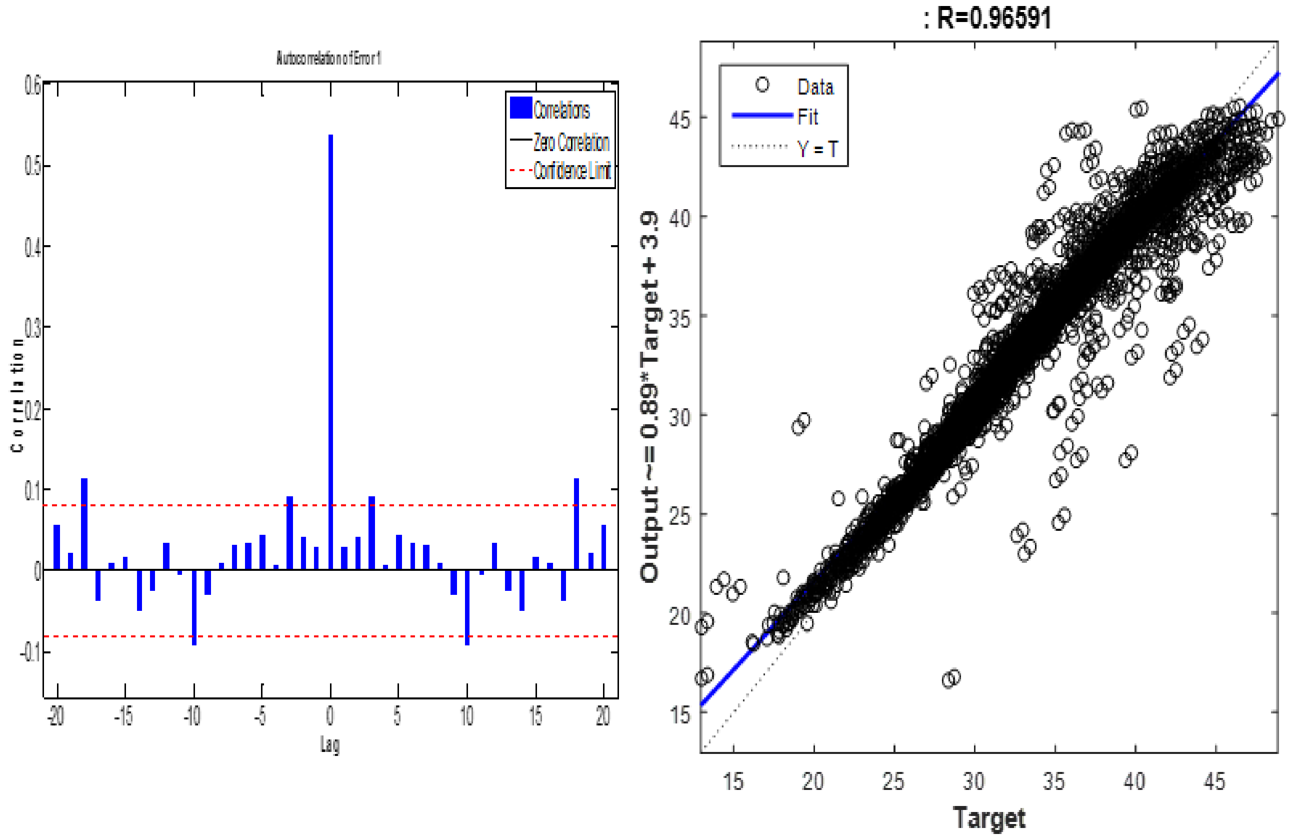


Figure 4.14: Time Series Plot of Actual Data, SARIMA and NARX (LM) Model Results for Forecasting 365-days of Peak Load Demand Ahead in 2015.

Figure 4.15 (a) shows the autocorrelation of error and the output-target autocorrelation of the proposed NARX neural network model, with the error autocorrelation falling within the acceptable confidence limit of ±0.1.The regression on the error between forecast and actual is a measure of performance, and it is plotted in Figure 4.15 (b). Detail forecast analysis of the results is shown in Appendix C.



(a) (b)

Figure 4.15: (a) Autocorrelation of Residual Error at Lag 20 and (b) Regression of output-target and forecast series for 365 Days, Ahead Prediction of Peak Load Demand using LM Algorithm.

The training, by means of a Levenberg-Marquardt backpropagation training algorithm converged after fewer than 21 epochs and best validation recorded was 2.3867 at Epoch 21it showed stability (no increase after converging) and no overshoot (no increase before converging), as shown in Figure 4.14(c).The work willshow the summary of the analysis and other observations in the context of the overall results in the next sections.

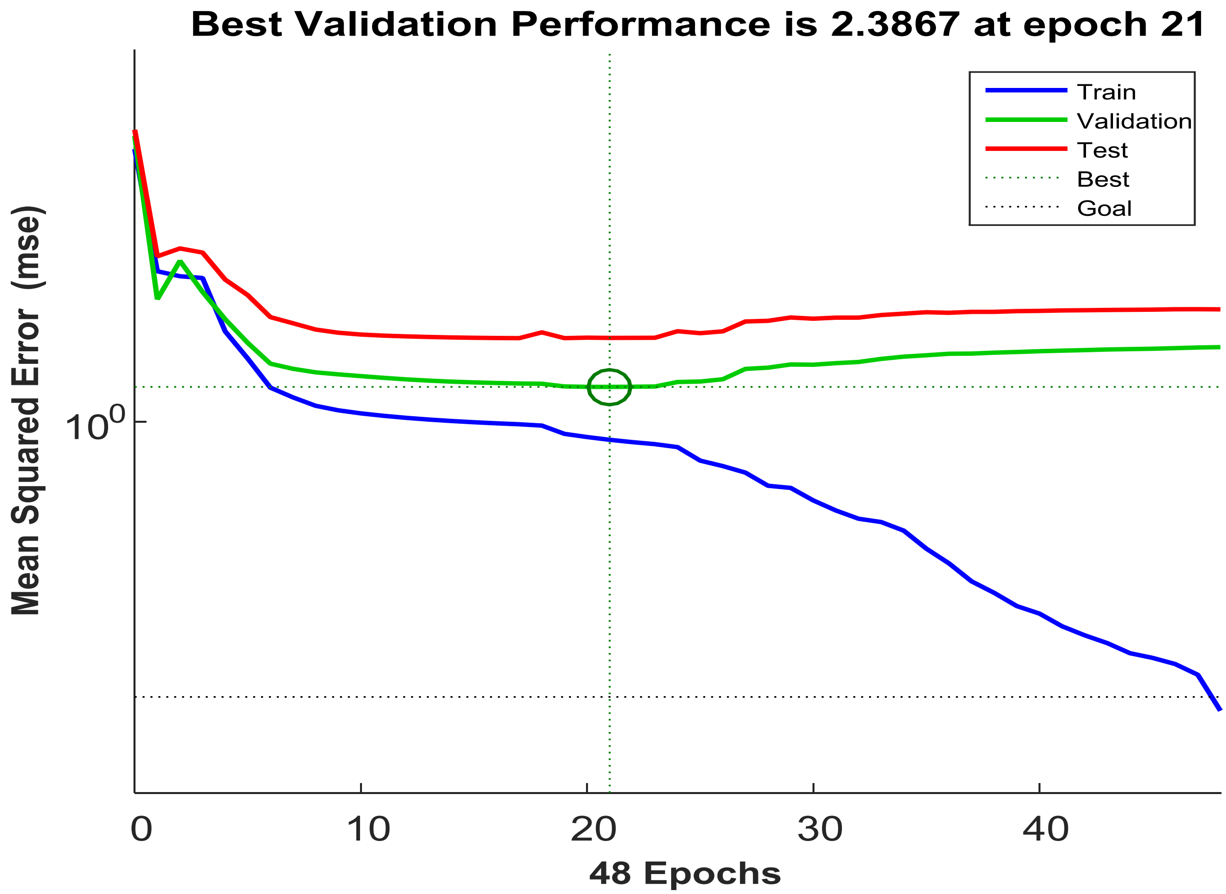


Figure 4.15 (c): Br Training Performance Plot; Best Validation Recorded was 2.3867 at Epoch 21

4.3.3 Comparison of the Models Performance Evaluation Results

The summary of the results obtained for performance evaluations of the models forecast by each of the model is showed in Table 4.8. However, the output performance evaluation results of each models were further compared by calculating their percentage improvement in term of forecasting error reduction (smallest MAPE) using equation (4.1)and their results is shown in Table 4.9 and Figure 4.16 respectively.

$$\text{Percentage Improvement} = \frac{Initial - Final}{Initial} \times 100\% \tag{4.1}$$

Table 4.8: Summary of Models Performance for SARIMA, NARX (LM) and NARX (Br)

Model Evaluation	**Performance Metrics**	
	MAPE (%)	**R (%)**
SARIMA	6.6402	67.20
SARIMA-NARX ($\boldsymbol{Br}$)	4.0420	95.98
SARIMA-NARX ($\boldsymbol{LM}$)	2.4102	96.59

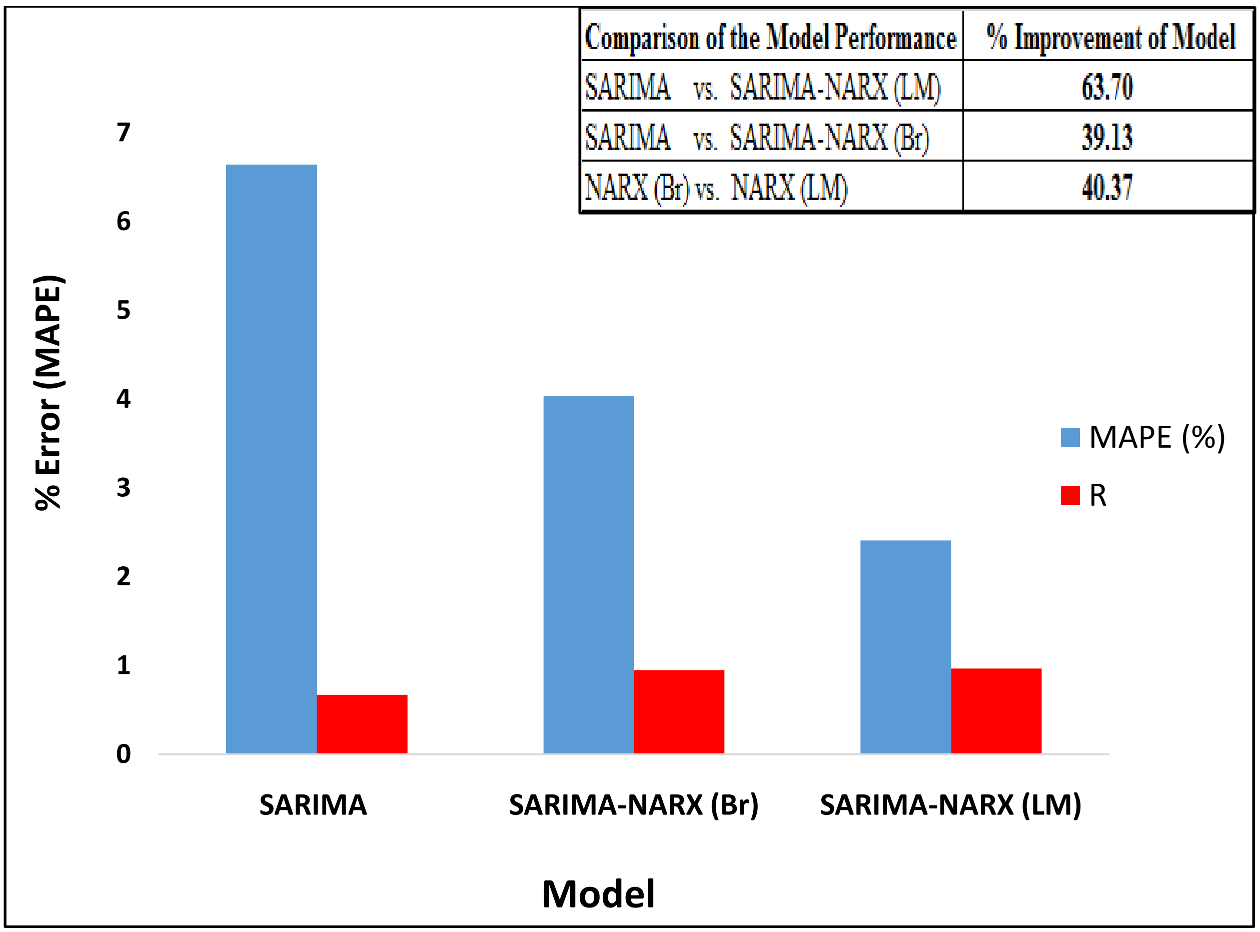
***smaller values mean higher forecast accuracy.*

Table 4.9: Model Performance Comparison and Its Percentage Improvement

Comparison of the Model Performance	**% Improvement**
SARIMA vs. SARIMA-NARX ($\boldsymbol{LM}$)	63.70
SARIMA vs. SARIMA-NARX ($\boldsymbol{Br}$)	39.13
SARIMA-NARX ($\boldsymbol{LM}$) vs. SARIMA-NARX ***(Br)***	40.37

*** higher % improvement means lowestforecast error*

From Table 4.8,it is observed that among all the models, the proposed SARIMA-NARX model trained with LM algorithmachieved lowest MAPE of 2.4102% and R of 96.59%. This implies thatthe model established superiority overall the other methods i.e. SARIMA of 6.6402% and SARIMA-NARX (Br) of 4.0420% in term of training time and forecast accuracy for forecasting 365-days ahead of daily peak load demand for Nigeria power system grid.Thecomparative analysis results of models' performance in term of smallesterror reduction and thecorresponding percentage improvement is presented inTable 4.9. It can be seen that the developed model trained with LM produced a lowest error reduction with an equivalent improvement of 63.70%as compared with traditional SARIMA model. In practice, it means that when comparing the power generation to the load consumption at peak period, this simulation results shows that there is an estimated forecast error reduction which can be equivalentto the cost of producing 97.769MW of powers. In addition, when compared the performancetwo-training algorithm used in this work, the proposed model $(SARIMA - NARX_{LM})$ outperformed the model trained with $Br$with an improvement of 40.33%which is equivalent to cost of producing 41.5MW power. The detailed of the results in Table 4.8 and 4.9 is depict with bar chart representation shown in Figure 4.16



Comparison of the Model Performance	% Improvement of Model
SARIMA vs. SARIMA-NARX (LM)	63.70
SARIMA vs. SARIMA-NARX (Br)	39.13
NARX (Br) vs. NARX (LM)	40.37

Figure 4.16:Comparative Analysis of the Models Performance Evaluation &Percentage Improvement.Smaller values mean higher forecast accuracy

Therefore, it can be concluded that training theproposed model using the Logistic Sigmoid (LogSig) activation functions in the hidden layer and Levenberg-Marquadt training algorithm with tents of chaos for the selection of number of neurons in the hidden layershows anenhanced the optimal structure of the NARX to produced fast convergence rate.Furthermore, it is clear to see that the developed improved SARIMAbased on NARX neural network model optimized with tenet of chaoshas great improvement on the accuracy of the forecastedoutput withdifference of 4.23% error reduction when compare with multiplicative SARIMA method. This is as a result of the non-linear nature of the improved NARX model which incorporates chaos dynamics in its architecture. It is also worthy to note that the Levenberg-Marquardt (LM) training function is faster and out-performed the Bayesian-regularization (Br) training function in this model in terms of processing speed and accuracy of results. Hence, the proposed modelis highly recommended for forecasting chaotic time series.

4.4 Results of Forecasting the Daily Peak Load Demand forYear 2017 and 2018 using the Proposed Model for Planning of Generation Scheduling and Timely Load Dispatch on Nigeria Power System Grid.

Theentire data from 2006-2015 was inputted into the model, the processes of model selection were executed and daily peak load demand forecasts for year 2016 was obtained and stored in a new text file. In a similar vein the new data set from 2006-2016 was again inputted, the procedures repeated again and the peak load demand forecast for the year 2017 obtained. The same process was also repeated to produced year 2018 forecast results. These results for year2017 and 2018 are displayed in Figure 17 (a) and (b). From the Figure, the expected load demand forecasted for year 2017 is between 4500MW to 5600MW and for year 2018 the demand will increase from about 5600MW to about 6350MW respectively. This projection can assist all stakeholders in the industry in the area of planning.See the detail of MATLAB program in Appendix B4.

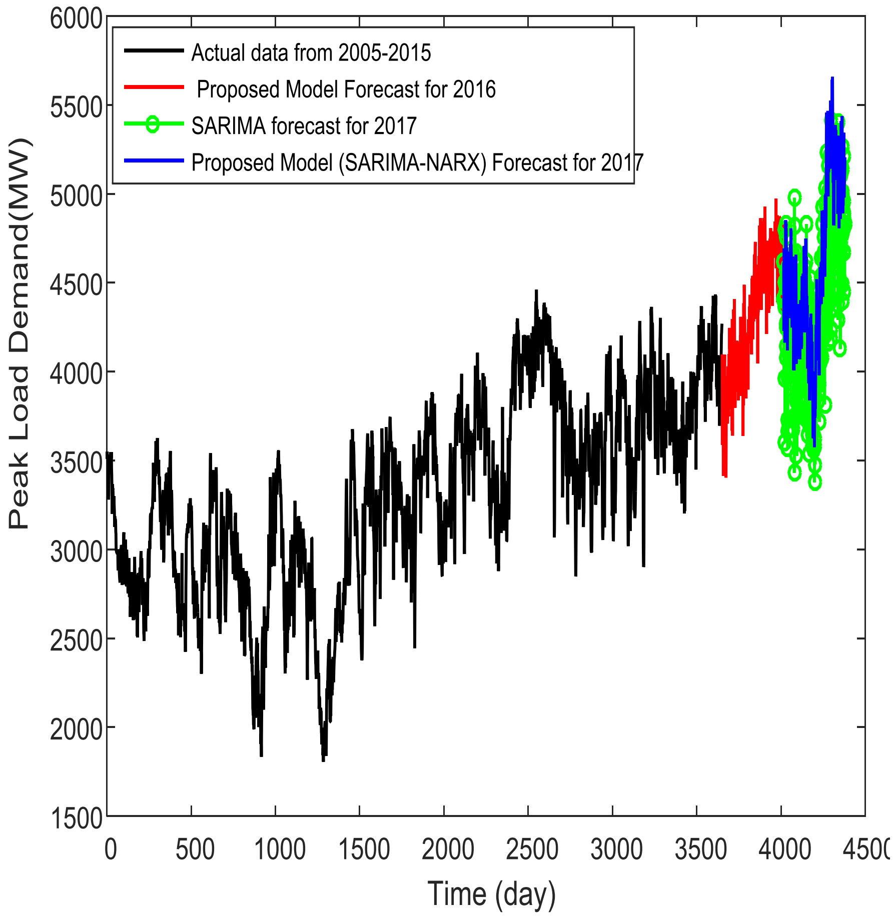


Figure 17(a): Daily Peak Load Forecasting for year 2017 with Demand of 5600MW

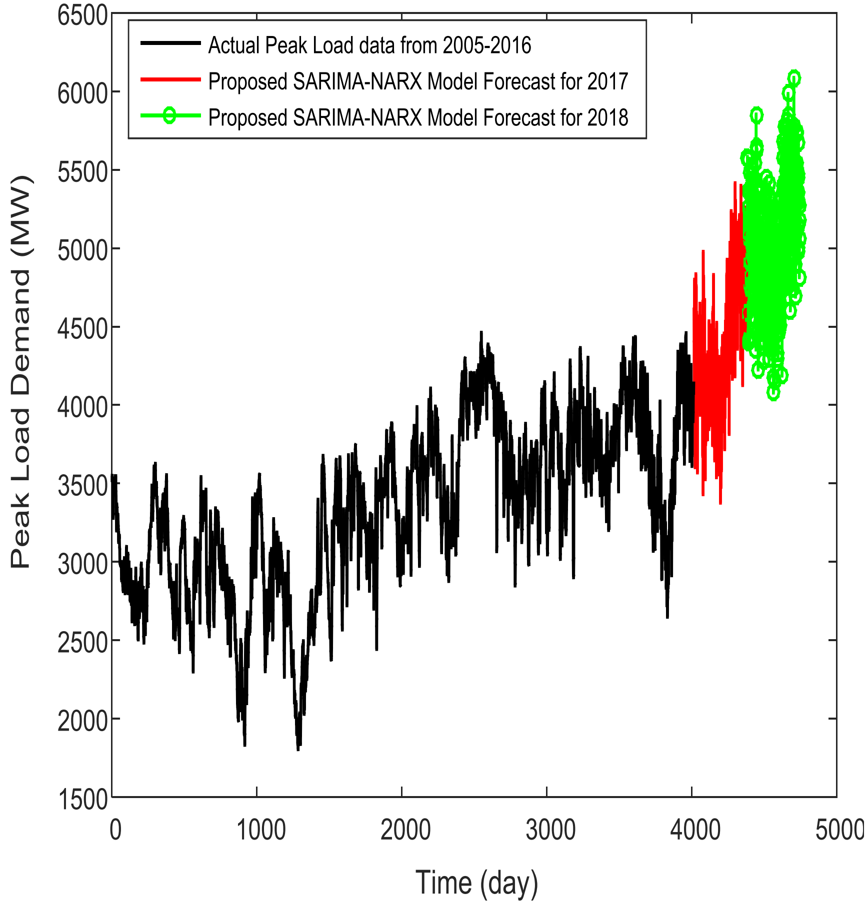


Figure 17b): Daily Peak Load Forecasting for year 2018 with Demand of 6350MW

Figure 4.17:(a) Days Ahead Peak Load Forecasting Results for year 2017; (b):Days Ahead Peak Load for 2018 using Proposed Improved SARIMA based NARX Neural Network Model for Nigeria Power System Grid Operation with anticipated High Load Demand of 5600MW in 2017 and 6350MW in year 2018

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

A number of researchers have mostly used single model in forecasting the load demand of Nigeria. This research presents a developed short-term load forecasting model based on concatenation of SARIMA and NARX neural network for improving the accuracy of daily peak load demand in Nigeria power system grid operation. The model was trained and validated and model forecast results were compared with the actual data of daily peak load demand of Nigeria power system grid (TCN). The dynamics of the daily peak load time series data was examined and the results showed that the data is chaotic. The intrinsic parameters of the model were optimized using the tenets of chaos theory. The results of performance evaluation show that the developed modeltrained with Levenberg-Marquardt algorithm outperforms single SARIMA model and the Bayesian regularization algorithm trained model. This produced a significant reduction in forecast error with a percentage improvement of 63.70%. However, considering the impact of error reduction on the generation scheduling for peak load demand of case study areas network (TCN), it is evidence that this can save generation cost of about 97.77MW losses per day at peak. Because as at the last data information collected the network has total available generation capacity of 7511-MW, unit on bar capability at peak 4129-MW and typical peak load demand varying from 3534.9 MW to 4056.5 MW. It means the system incurred an overshoot of 4.23% - 4.6% times of the forecasted loads, so our prediction offers a real-time opportunity for large cost savings. From economical generation point of view, research by Buitrago (2017)established that forecast error less than 5% precision present a significant reduction in the generation costs and increase large savings by avoiding commissioning of unnecessary power plants. Hence, having error below this precision is more economical and this work has achieved below 3% precision. In addition, even though the trends were accurately predicted with good precision, the relatively fair but acceptable MAPE values obtained are as a result of the weak correlation of the input and target data as shown in Table 3.1 which could be as a result of large number of missing data and white noise in the meteorological data which may have arisen from faulty equipment or human error/neglect in measurement. However, year 2017 and 2018 load projection were also obtained with peak at 5600MW in 2017 and 6350MW in year 2018.

5.2 Significant Contribution

The significant contributions of this research are as follows:

1. The main contribution of this research is that this work has successfully developed an improveshort-term load forecasting model using seasonal ARIMA based on NARX neural network and applied it to forecast daily peak load demand forNigeria power system grid.
2. Secondly, while building the proposed model used recent Nigeria deregulated market peak generation data, which reflect that the system dynamics is chaotic over time. This gives an accurate predictability of 124 days per annum for future daily peak load demand planning for generation scheduling and timely dispatch otherwise; the grid might operate at serious risk of becoming vulnerable.
3. The proposed method has achieved a significant error reduction below 3%with a performance improvement of 63.70% which equivalent to cost of producing 97.769MW of power per day when there is any interruption in load supply at load a peak generation.The model also gives the future average daily peak load demand projection of year 2017 and 2018 to be5600MW and 6350MW respectively at peak to enhance generation planning and decision making.

5.3 Limitations

This research work could have achieved more improvement if not for following limitations:

1. The limited access to more recent primary data to date (i.e. unavailability of good data dating up to 2017) form both TCN and NiMet database. The data collected have large number of missing data and white noise which may have arisen from faulty equipmentor human error/neglect in measurement. However, this has significant effect on correlation of variables. This limitation calls for a pre-interpolation, filtering and normalization/validation of the data.

5.4 Recommendations for Further work

The following further research and studies are recommended in order to be able to further improve on forecasting errors:

1. As shown by this study, training algorithm used has huge influence on the results of the forecast. However, not all training method work in all cases it is recommended that more study be carryout on impact of training method using chaotic time series data of deregulated power market so as to achieve more improve accurate results.
2. Further research can investigate the effect of includingmore load affecting variables such as price, currency exchange rate, Gross Domestic Product, and population growth amongst others on the accuracy of the model for Nigeria power market.
3. The degree of error varies from utility to utility and is generally determined bythe processes run at that utility.Further research can investigate the accuracy of the model on distribution network for consumers based load and prices forecasting. This will addresspersistent problems of consumers tariff review by the distribution utilities.
4. Further study on how to determine the optimal network architecture are require. The number of input nodes, hidden layers, hidden nodes and output nodes and the manner in which they are all interconnected are still determined by trial and error, and in spite of many efforts to find a method to determine an optimal configuration, none have proven all-encompassing. It would be necessary to develop an analytical model for network

architecture so that it would be easy to identify the type of network and configuration to use for each type of data input.

REFERENCE

Abdollah K. F.&Mohammad-Reza A.Z., (2014).A hybrid method based on wavelet, ANN and ARIMA model for short-term load forecasting, *Journal of Experimental & Theoretical Artificial Intelligence,*Volume 26, Issue 2, pp.167-182,

Adepoju, G. A. (2007). Application of Neural Network to Load Forecasting in Nigerian Electrical Power System. *Practical Journal of Science and Technology,* 8: 68-72.

Afshin M, Sadeghian A (2007) PCA-based least squares support vector machines in week-ahead load forecasting. *In: Proceedings of IEEE industrial and commercial power systems technical conference, pp 1–6*

Al Shamisi, M. H., Assi, A. H. and Hejase, H. A. (2011). Using MATLAB to Develop Artificial Neural Network Models for Predicting Global Solar Radiation in Al Ain City-UAE. In Assi, A. H. (eds). Engineering Education and Research Using MATLAB. Croatia, In-Tech, pp. 219-238.

Al-Hamadi H.M., Soliman S.A., (2005). Long-term/mid-term electric load forecasting based on short-term correlation and annual growth, *Electric Power Systems Research* 74 (3) 353–361.

Amakali S., (2008). Development of models for short-term load forecasting using artificial neural networks, Master of Technology Theses & Dissertations, Discipline: Electrical Engineering, Faculty of Engineering, Cape Peninsula University of Technology (CPUT),Bellville Campus Paper 32. Pp. 44-53 (Published). *http://dk.cput.ac.za/td_cput/32*

Amjady N., (2006). Day Ahead Price Forecasting of Electricity Markets by a New Fuzzy Neural Network. *IEEE Trans Power System, 21(2):887–96*.

Anctil, F. and Rat, A. (2005). Evaluation of neural network streamflow forecasting on 47 watersheds. *Journal of Hydrologic Engineering*, 10, 85–88.

Aqeel A. B., Meysam D., and Gholamreza Z., (2012). Electricity Demand Estimation using an Adaptive Neuro-Fuzzy Network: A case study from the state of Johor, Malaysia. *Proceeding of 1st Congress on Applied Chemical Science, International Conference on Energy and Environmental-ICEE 2012.*

Arash A., Farid A., (2009).Multi-step ahead forecasts for electricity prices using NARX*: A new approach, a critical analysis of one-step ahead forecastsEnergy Conversion and Management, An International Journal, Elsevier Ltd. 50 (2009) 739–747,*

As'ad M., (2012). Finding the Best ARIMA Model to Forecast Daily Peak Electricity Demand, *Proceedings of the Fifth Applied Statistics Education and Research Collaboration (ASEARC)* - Looking to the future - Programme, 2 - 3 February, University of Wollongong.

Banda E., Folly K.A., (2015). *Short Term Load Forecasting Based on Hybrid ANN and PSO. Springer International Publishing Switzerland, Y. Tan et al. (Eds.): ICSI-CCI 2015, Part III, LNCS 9142, pp. 98–106, 2015. DOI: 10.1007/978-3-319-20469-7_12*

Beale, M. H., Hagan, M.T. and Demuth, B.H. (2014). *Neural Network ToolboxTM . User's Guide (R2014b). USA, The MathWorks, Inc., pp. 23-34.*

Bedri A. ÖZER*, Erhan AKIN*, (2005). Tools for Detecting Chaos, SAÜ Fen Bilimleri Enstitüsü Dergisi 9.Cilt, 1.Sayõ 2005.

Ben K., & Patrick V. S., (1996). An Introduction to Neural Networks, 8th Edition, The University of Amsterdam.

Box G.E., Jenkins G.M., (1976). Times series analysis: forecasting and control, San Francisco, USA: Holden- Day, 1976.

Bozkurt ÖÖ., Biricik G, Tayşi Z. C., (2017). *Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market. PLoS ONE 12(4): e0175915. https://doi.org/10.1371/journal.pone.0175915*

Buhari M., & Adamu S. S., (2012): Short term load forecasting using Artifical neural network, Proceedings of International MultiConference of Engineers and Computer Scientists (IMECS 2012), Vol. I, March 14-16, Hong Kong.

Cai, Y.,Wang, J.-Z., Tang, Y. and Yang, Y.-C. (2011). An efficient approach for electric load forecasting using distributed ART (adaptive resonance theory) & HSARTMAP (Hyper-spherical ARTMAP network) neural network. *Energy*. 36, 1340-1350.

Caston S., and Delson C., (2010). Daily peak electricity load forecasting in South Africa using a multivariate non-parametric regression approach, *ORiON Journal,* Volume 26 (2), pp. 97-111, ISSN 0529-191-X, *http://www.orssa.org.za*

Chaoming H., Chi-Jin H., Mingli W., (2005). A Particle Swarm Optimization to Identifying the ARMAX Model for Short-Time Load Forecasting, *IEEE Transactions on PowerSystems*, Vol.20, No.2, 2005, pp.1126-1133.

Di Piazza A., Di Piazza M. C., and Vitale G., (2014). Estimation and Forecast of Wind Power Generation by FTDNN and NARX-net based models for Energy Management Purpose Smart Grids. *International Conference on Renewable Energies and Power Quality (ICREPQ'14) Cordoba (Spain), 8th to 10th April, 2014, ISSN 2172-038 X, No.12, April 2014DOI: 10.1080/0740817X.2014.999180*

Diaconescu E., (2008).The use of NARX Neural Networks to predict Chaotic Time Series, WSEAS Transactions on Computer Research, Issue 3, Volume 3, pp. 182 – 191, March 2008

Diaz-Robles, L. A., Ortega, J. C., Fu, J. S., Reed, G. D., Chow, J. C. Watson, J. G. and Moncada-Herrera, J, A. (2008). A Hybrid ARIMA and Artificial Neural Networks Model to Forecast Particulate Matter in Urban Areas: The case of Temuco, Chile. *Elsevier*, *Atmosphere environment*, 42(2008), 8331-8340.

Di-Piazza A., Maria Carmela Di Piazza, and Gianpaolo Vitale (2016). *Solar and wind forecasting by NARX neural networks, Renew. Energy Environ. Sustain. Vol.1, No.39 published by EDP Sciences, 2016, DOI: 10.1051/rees/2016047*

Dwijayanti S., (2013). Short Term Load Forecasting Using Artificial Neural Network Based on Time Series Approach, Master of Science in Electrical Engineering Thesis, Faculty of the Graduate College of the Oklahoma State University Indonesian May, 2013 (Published).

Echi I.M., E.V. Tikyaa, Isikwue B.C., (2015). Dynamics of Daily Rainfall and Temperature in Makurdi. International Journal of Science and Research (IJSR), Volume 4 Issue 7, pp.493-499.

Ellen Shezi., (2015). Short term Load Forecasting based on Hybrid ANN and PSO, Thesis presented for the degree of Master of Science in Electrical Engineering in the department of Electrical Engineering, University of Cape Town, 16 February 2015 (Publised).

Feilat E.A., Bouzguenda M., (2011). Medium-term load forecasting using neural network approach. *IEEE 2011 PES Conference on Innovative Smart Grid Technologies*-Middle East (ISGT Middle East), pp. 1-5, 17-20 December 2011, DOI: 10.1109/ISGT-MidEast.2011.6220810.

Hagan, M. T., Demuth, H. B., Beale, M. H. and Jesus, O. D. (1996). Neural Network design (2nd ed.). Oklahoma, Amazon press, 1012pp.

Hakki, E. (2006) On the Predictability of Time Series by Metric Entropy. Unpublished M.Sc. Thesis, Graduate School of Engineering and Sciences, Izmir Institute of Technology.

Harun, M. H. H., Othman M. M., and I. Musirin, (2009). Short-Term Load Forecasting (STLF) using artificial neural network based multiple lags of time series. Lecture Note Computer Science, 5507: pp. 445-452.

Hasselblatt, B. and Anatole, K. (2003). A First Course in Dynamics: With a Panorama of Recent Developments. Cambridge University Press, London. P.56.

Haykin, S. (2009). Neural Networks and Learning Machines, 3rd ed. New Jersey, Pearson Education, Inc., p. 37.

Heiko H., Silja M., Stefan P., (2009). Electric load forecasting methods: Tools for decision making, *European Journal of Operational Research (Science Direct) vol. 199 p.902–907.*

Hitech K. and Sanju S. (2016). Chaotic Characterization of Electric Load Demand Time Series & Load forecasting by using GA trained Artificial Neural Network. International conference on signal processing, communication, power and embedded system (SCOPES)-2016

Hongzhan N., Guohui L., Xiaoman L., Yong W. (2012). Hybrid of ARIMA and SVMs for Short-Term Load Forecasting, *Elsevier energy Procedia, 16 (2012) 1455 – 1460*

Hui L., Hong-qi T., Yan-fei L., (2012). "Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction, *Applied Energy Elsevier energy Procedia, Vol.98, pages 415–424*

Ina Khandelwal, Ratnadip A., Ghanshyam V., (2014).Time Series Forecasting using Hybrid ARIMA and ANN Models based on DWT Decomposition, *Proceeding of International Conference on Intelligent Computing, Communication & Convergence (ICCC), At Bhubaneswar, India, Volume: Computer Science, Elsevier December 2014*

Jaime & Shihab, (2017). Short term Forecasting of Electric Load Using Nonlinear Autoregressive Artificial Neural Networks with Exogenous Vector Inputs, Journal of Energies MDPI, doi:10. 3390/en10010040.

Jason G., Moataz E., and Shihab A., (2014). Short-Term Electrical Peak Demand Forecasting in a Large Government Building Using Artificial Neural Networks, *Energies Open Access Journal, Vol.7*, Pp. 1935-1953; doi:10.3390/en7041935 ISSN 1996-1073,

Jayawardena, A., Achela, D. and Fernando, K. (1998). Use of Radial Basis Function Type Artificial Neural Networks for Runoff Simulation. *Computer-Aided Civil and Infrastructure Engineering*, 13, 91–99.

Jeffrey, A. (2002). Advanced Engineering Mathematics. USA, Harcourt/Academy Press, pp. 1062-1064.

Buitrago, J. H., (2017) "Short-Term Forecasting of Electric Loads Using Nonlinear Autoregressive Artificial Neural Networks with Exogenous Multivariable Inputs" (2017). Open Access Dissertations. 1908. http://scholarlyrepository.miami.edu/oa_dissertations/1908

Jingmin W., Guoqiao R., (2006), Load Forecasting Based on Chaotic Support Vector Machine with Incorporated Intelligence Algorithm, *IEEE First International Conference on Innovative*

Computing, Information and Control, (ICICIC '06). Beijing China, Vol. 3, Aug. 30 -Sept. 1, 2006, pp 435 – 439.

Kandananond, K., (2011). A Forecasting Electricity Demand in Thailand with an Artificial Neural Network Approach, *Energies Journal, vol. 4, pp.1246- 1247.*

Karytinos, A. (1999). *A Chaos Theory and Nonlinear Dynamics Approach to the Analysis of Financial Series a Comparative Study of Athens and London Stock Markets*. A dissertation presented for the Doctor of Philosophy Degree Warwick Business School, University of Warwick, UK. P. 4-19, 20-21.

Kennel, M.B., Brown, R., Abarbanel H.D. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction." *Phys. Rev. A*, **45**(6), 3403-3411.

Khashei M., M. Bijari (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Elsevier Journal of Applied Soft Computing, 11 (2011) 2664–2675*

Kiusalaas, J. (2010). Numerical Methods in Engineering with MATLAB. UK, Cambridge University Press, Pp. 116-119.

Kwiatkowski, D., Phillips, P. C., Schmidt, P. and Shin, Y. (1992). Testing the Null Hypothesis of Stationarity against the Alternative of a Unit Root. Journal of Econometrics, 54, 159–178.

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.b7, No. 1, pp.124 – 132.

Lei D., Shuang G., Xiaozhong L., Yang G., (2012). Short-Term Wind Power Forecasting with Combined Prediction Based on Chaotic Analysis, *PRZEGLĄD ELEKTROTECHNICZNY (Electrical Review), pp. 35 -39, ISSN 0033-2097, R. 88 NR 5b/2012.*

Letellier, C. (2013). Estimating the minimum embedding dimension," ATOMOSYD Algorithms, 619, 234-236.

Li Li., and Liu C., (2011*).* Application of Chaos and Neural Network in Power Load Forecasting. *Journal of Discrete Dynamics in Nature and Society,* ***2011****, pp.1-12. doi:10.1155/2011/597634.*

Lin T., B. G. Horne, and C. L. Giles, (1998). "How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies," *Neural Networks, vol. 11, no. 5, pp. 861–868.*

Lisheng Y., Yigang H., Xueping D., Zhaoquan Lu., (2013). Adaptive Chaotic Prediction Algorithm of RBF Neural Network Filtering Model based on Phase Space Reconstruction, *Journal of Computers, Vol. 8, No. 6, Pp.1449 – 1455*.

Luiz F., and Afshin A. (2015).Short-term forecasting of the Abu Dhabi electricity load usingmultiple weather variables*. The 7th International Conference on Applied Energy – ICAE2015, elsevier energy procedia 75 (2015) 3014 – 3026.*

Maged M. E., & Elsayed E. H., (2017). *Enhancing Electric Load Forecasting of ARIMA and ANN Using Adaptive Fourier Series, IEEE power sys. trans 2011ISBN 978-3-642-17988-4, DOI 10.1007/978-3-642-17989-*

Manoj K., (2009). Short-Term Load Forecasting Using Artificial Neural Network Techniques, Bachelor of Technology (B.Tech.) Degree Thesis,Department of Electrical Engineering, National Institute of Technology, Rourkela Indian, Pp. 8-12

Mansour S., Najmeh M., (2011). *Neural-based electricity load forecasting using hybrid of GA and ACO for feature selection. Spring Journal of Neural Computation & Application, Pp.1-10. DOI 10.1007/s00521-011-0599-1*

MathBits.com (2015). Statistics 2: Correlation Coefficient and Coefficient of Determination. Retrieved from *http://mathbits.com/MathBits/TISection /Statistics2/Correlation.html.*

MathWorks (2015). Information Criterion: Bayesian information criterion. www.mathworks.com

MathWorks(2016). Divide Data for Optimal Neural Network Training. *http://www.mathworks.com/help/nnet/ug/divide-data-for-optimal-neural-network-training.html. Retrieved on 16-06-2016.*

Menezes -Jnr, J. M. and Barreto, G. A. (2006). A New Look at Nonlinear Time Series Prediction with NARX Recurrent. IX Brazilian Neural Networks Symposium, October 23-27, 2006, Department of Teleinformatics Engineering, Av. Mister Hull, S/N, CP 6005, CEP 60455-760, Fortaleza, Cear´a, Brazil.

Menezes -Jnr, J. M. and Barreto, G. A. (2007a). Long-Term Time Series Prediction with the NARX Network: An Empirical Evaluation. Preprint submitted to Elsevier Science, Department of Teleinformatics Engineering, Av. Mister Hull, S/N, CP 6005, CEP 60455-760, Fortaleza, Cear´a, Brazil.

Mohammad S.,Hatim Y., Zuyi Li (2002). Market Operations in Electric Power Systems Forecasting, Scheduling, and Risk Management. A John Wiley & Sons, Inc., Publication.

Mohammed A. M., Wassee H. A., Amin T. A., (2016). *Short term Forecasting Based on NARX and Radial Biasis Neural Networks Approaches for Jordanian Power Grid, Jordan Journal of Electrical Engineering, Vol. 2, No. 1, ISSN 2409-9600*

Muhammed B., Sanusi S. A., (2012). Short Term Load Forecasting Using Artificial Neural Network, *Proceedings of the International MultiConference of Engineers and Computer Scientist (IMEC 2012), Vol. I, March 14 – 16, Hong Kong.*

Musa M., (2017). Real Time Measurement and Prediction of Nigeria Power System Frequency in FINET Using Artificial Neural Network, M.Sc. Thesis, Electrical Engineering (Power system), Ahmadu Bello University Zaria, Nigeria (Unpublished).

Musa S. Y., Mbaga E. V., (2014). Daily Nigerian Peak Load Forecasting using Artificial Neural Network with Seasonal Indices, *Nigerian Journal of Technology(NIJOTECH), Faculty of Engineering, University of Nigeria, Nsukka, Vol. 33. No. 1, Pp. 114 –118, www.Nijotech.Com, Http://Dx.Doi.Org/10.4314/Njt.V33i1.15*

Narendra B., Pallaviram S., (2016). Partitioning and interpolation based hybrid ARIMA–ANN model for time series forecasting, *Sādhanā academy proceedings in engineering science, Springer India, pp 1-12, First online: 12 July, 2016.*

Nau, R. (2014). *Introduction to ARIMA models. Lecture notes on forecasting, Fuqua school of Business, Duke University. http://people.duke.edu/ ~rnau/forecasting.html*

Nitin Singh, S. R. Mohanty (2015). A Review of Price Forecasting Problem and Techniques in Deregulated Electricity Markets. *Journal of Power and Energy Engineering, Scientific Research Publication, Vol. 3, pp. 1-19*

Niu D., Liu, D., & Wu, D.D. (2010). A soft computing system for day-ahead electricity price forecasting. *Journal of Applied Soft Computing, Vol.10, 868–875*

Nwohu, M. N. (2009), *"Voltage Stability Enhancement of the Nigerian grid system using flexible alternating current transmission system (FACTS) devices", Ph.D. thesis. Abubakar Tafawa Balewa University, Bauchi, Nigeria, pp. 7-12 (Upublished).*

Ozer, B.A. and Akin, E. (2005) "Tools for Detecting Chaos." *Sau Fen Bilimleri EnstitusuDergisi.* 9(1):60-66.

Ozozen A., Kayakutlu G., Marcel K., Ozgur K., (2016). *A combine Seasonal ARIMA and ANN Model to Improved Results in Electricity Spot Price Forecasting: A Case Study in Turkey. IEEE proceeding of PICMET' 16 Tchnology Management and Scaial innovation, Pp. 2681*-2690

Packard, N., Crutchfield, J., Farmer, J. and Shaw, R. (1980) "Geometry from a Time Series." *Phys. Rev. Lett.*, 45, 712.

Pillai G.G., Ghanim A. Putrus, Nicola M. Pearsall. (2014). *Generation of synthetic benchmark electrical load profiles using available load and weather data. Elsevier Journal of Electrical Power and Energy Systems, 61(2014), 1-10*

Rafal Weron (2006) Modeling and forecasting electricity loads and prices: A statistical approach Published by John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England.

Rasoul, J., Mohammad, G. and Abolfaz, S. (2014) Dynamics of Rainfall in Ramsar. *Journal of Applied Science and Agriculture*, 9(4), 1371-1378.

Rob J Hyndman and Shu Fan (2008). *Density forecasting for long-termpeak electricity demandDepartment of Econometrics and Business Statistics, working paper 06/08, Monash University, http://www.buseco.monash.edu.au/depts/ebs/pubs/wpapers/*

Rosenstein, M., Collins, J., De Luca, C. (1992). A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D*,65, 117-134.

Sanjib M., (2008) "*Short Term Load Forecasting Using Computational Intelligence Methods" Master of Technology degree thesis in Electronics & Communication Engineering (Specialization in Telematics and Signal Processing), Department of Electronics and Communication Engineering National Institute of Technology, Rourkela India*.

Santos, P.J.,Martins, A. G.and Pires, A. J., (2007). *Designing the input vector to ANN-based models for short-term load forecast in electricity distribution systems. Electr. PowerEnergy Syst. 29, 338–47.*

Seifi H., and M. S. Sepasian, (2011). Electric Power System Planning, Power Systems Issues, Algorithms and Solutions, DOI: 10.1007/978-3-642-17989-1-1, Springer-Verlag Berlin Heidelberg. 2011

Shannon, H. (2013). Power Spectrum in MATLAB. BitWeenie: Digital Signal Processing. Retrieved from *http://www.BitWeenie.html.* Accessed on 4-8-2013.

Soliman S.A., and Al-Kandari A. M., (2010). Electrical Load Forecasting – Modeling ad Model Construction, (ed), Butterworth–Heinemann publications, British Library Cataloguing-in-Publication Data. Elsevier 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA, pp. 80-81

Stylianos I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, A. G. Bakirtzis (2016). Comparison of SARIMAX, SARIMA, Modified SARIMA and ANN-based Models for Short-Term PV Generation Forecasting. *IEEE Journals of Power Systems,* Takens F., (1980). Detecting Strange Attractors in Turbulence. *Lecture Notes in Mathematics,*898, 366.the Foreign Exchange Market, *LSE Financial Markets Group, Discussion Paper No 86*, London School of Economics.

Telgarsky, R. (2013). Dominant Frequency Extraction. arXiv: 1306.0103(1), 1-12.

Thays, A., Carlos R S. J., Anna P. L., Mara L. M. L., (2016).*Electrical Demand Load Forecasting by ARIMA Regression and Artificial Neural Networks.International Journal of Computer and Information Technology, Volume 05 – Issue 03, May 2016, ISSN: 2279 – 0764*

Wei-Chiang Hong (2009). Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm, *Energy Conversion and Management on Science Direct,Volume 50, Issue 1, January 2009, Pages 105–117*.

Xiping W., & Ming M., (2012). A Hybrid Neural Network and ARIMA Model for Energy Consumption Forecasting, Journal of Computers, Vol. 7, No. 5, pp. 1184 – 1190.

Yusri S. A., Hajime M., (2013). Seasonal peak electricity demand characteristics: Japan case study International *Journal of Energy and Power Engineering 2(3): pp. 136-142,*

Zeng, X. (1992). Chaos Theory and its Application in the Atmosphere. *Atmospheric Science,* 504, 5-44.

Zhang G. P., (2003). Time series forecasting using a hybrid ARIMA and neural network model, *Elsevier Journal of Neurocomputing,vol. 50, pp.159 – 175.*

Zhang Q., Zhang Li (2008). Short Term Load Forecasting Using Chaotic Analysis Algorithm for Neural Network, pp366 – 370.

APPENDIX A

M-file MATLAB Codes for the Proposed Model

Appendix A1: Data smoothening using cubic spline interpolation, power spectrum and time series plot

```
% Input the data from the excel file 'TCN & NiMet DATA'
fid=fopen('exp.txt','w+');
yData = xlsread('c:\Users\Abdul\Documents\TCN DATA.xls',-1);
n=length(yData);
xData = transp(1:n);
A=0;
for x = 1:0.25:n;      % the new sampling rate
    y = spline(xData,yData,x);
    if y<0              % set any negative values equal to zero
       y=A;
    end
    count = fprintf(fid,'%10.5f',x,y);
    fprintf(fid,'\n');
end
               % Load data and check if the Statistics are correct
t=load('exp.txt');
y=t(:,2);
ybar=mean(y)
sbar=std(y)
variance=var(y)
coeffvar=sbar/ybar
snr=1/coeffvar
mx=max(y)
mn=min(y)
Rng = range(y)
N=length(y)
k=kurtosis(y)
s=skewness(y)
x=1:N;
for i=1:N
    if y(i)==0
        cnt(i)=1;
    else cnt(i)=0;
    end
end
 Nzeros=sum(cnt)
       % Plot new and old data to visualize the form of each signal
figure(1);
plot(x,y,'b');
xlabel('time(hrs)','fontsize',16);ylabel('peak load(MW)','fontsize',16)
figure(2);
plot (xData,yData,'r');
xlabel('time(days)','fontsize',16);ylabel('peak load(MW)','fontsize',16)
figure(3);
window=[];
nfft=[];
[Pxx,w]=periodogram(y,window,nfft);
semilogy(w,Pxx);
```

```
xlabel('freq.(cycles/hr)','fontsize',16);ylabel('PSD(dB/hr)','fontsize',16)
```

Appendix A2:Correlation Analysis of the Input Variables

```
%%% This program computes the weather variable that affect the peak load
demand
format short;
rh=load('exprh.txt');
hum=rh(:,2);                %% humidity
sr=load('expsr.txt');
solar=sr(:,2);              %% solar radiation
w=load('expwnd.txt');
wind=w(:,2);                %% wind speed
mnt=load('expmint.txt');
mint=mnt(:,2);              %% Minimum temperature
maxt=load('expmaxt.txt');
maxt=maxt(:,2);             %% Maximum temperature
pp=load('exppp.txt');
power=pp(:,2);              %% peak load demand
data=[mint maxt solar power];
rho=corr(data)
```

Appendix A3: Determination of time delay using the method of average mutual information (AMI)

```
%%% This program computes the delay time (tau)
%%% using the method of average mutual information

% Open file and input data
fid=fopen('avminfo.txt','w+');
y=load('exp.txt');
x=y(:,2);
lag=50;       % maximum time lag (hrs)
% Collect avmis at each lag i
for i = 1:lag
    v(i)= mai(x,i);
    count=fprintf(fid,'%10.5f',i,v(i));
    fprintf(fid,'\n');
end
% plot average mutual information
g=load('avminfo.txt');
plot(g(:,1),g(:,2),'ko-','markersize',4)
grid on;
xlabel('lag length','fontsize',16)
ylabel('average mutual information','fontsize',16)
title('average mutual info plot,tau=value of lag @ 1st local min of v')
```

```
function v=mai(x,lag)
%Syntax: v=mai(x,lag)
%____________________
%
% Calculates the mutual average information of a time series x for
% some time lag.
%
```

```
% v is the the value of the mutual average information.
% x is the time series.
% lag is the time lag.
% Alexandros Leontitsis
% Institute of Mathematics and Statistics
% University of Kent at Canterbury
% Canterbury
% Kent, CT2 7NF
% U.K.
% University e-mail: al10@ukc.ac.uk
% Lifetime e-mail: leoaleq@yahoo.com
% Homepage: http://www.geocities.com/CapeCanaveral/Lab/1421
%
% May 25, 2001.

if nargin<1 | isempty(x)==1
error('You should provide a time series.');
else
   % x must be a vector
   if min(size(x))>1
error('Invalid time series.');
   end
   x=x(:);
   % n is the time series length
   n=length(x);
end

if nargin<2 | isempty(lag)==1
   lag=0:min(n/2-1,20);
else
   % lag must be a vector
   if min(size(lag))>1
error('The time lag must be a scalar or a vector.');
   end
   % lag must contain integers
   lag=round(lag);
   % lag values must be between 0 and n/2-1
   lag=lag(find(lag>=0 & lag<n/2));
   % lag must not be empty
   if isempty(lag)==1
error('You must give another set of values for lag.')
   end
end

% The mutual average information
x=x-min(x);
x=x/max(x);
for i=1:length(lag)

   % Define the number of bins
   k=floor(1+log2(n-lag(i))+0.5);

   % If the time series has no variance then the MAI is 0
   if var(x,1)==0
      v(i)=0;
   else
```

```
      v(i)=0;
      for k1=1:k
         for k2=1:k
            ppp=find((k1-1)/k<x(1:n-lag(i)) & x(1:n-lag(i))<=k1/k ...
            ppp=length(ppp);
            px1=find((k1-1)/k<x(1:n-lag(i)) & x(1:n-lag(i))<=k1/k);
            px2=find((k2-1)/k<x(1+lag(i):n) & x(1+lag(i):n)<=k2/k);
            if ppp>0
               ppp=ppp/(n-lag(i));
               px1=length(px1)/(n-lag(i));
               px2=length(px2)/(n-lag(i));
               v(i)=v(i)+ppp*log2(ppp/px1/px2);
            end
         end
      end
   end
```

Appendix A4: Determination of optimal embedding dimension using the method of false nearest neighbors (FNN)

```
%%% This program determines the embedding dimension of a time
%%% serie using the method of false nearest neighbors (fnn)
% INPUT PARAMETERS:
% RTOL:
% ATOL:
% x:time series (signal)
% tao:time delay
% mmax:maximum embedding dimension

% Input the time series and parameters
y=load('exp.txt');
x=y(:,2);
rtol=15; atol=2; mmax=30;
tao=input('delay time = ');
% Begin computation using FNN or CAO method
tic
[FNN] = knn_deneme(x,tao,mmax,rtol,atol);
%[E1 E2] = cao_deneme(x,tao,mmax);
toc
```

```
function [FNN] = knn_deneme(x,tao,mmax,rtol,atol)
%x : time series
%tao : time delay
%mmax : maximum embedding dimension
% Author:"Merve Kizilkaya"
%rtol=15
%atol=2;
N=length(x);
Ra=std(x,1);

for m=1:mmax
    M=N-m*tao;
    Y=psr_deneme(x,m,tao,M);
    FNN(m,1)=0;
```

```
    for n=1:M
        y0=ones(M,1)*Y(n,:);
        distance=sqrt(sum((Y-y0).^2,2));
        [neardis nearpos]=sort(distance);

        D=abs(x(n+m*tao)-x(nearpos(2)+m*tao));
        R=sqrt(D.^2+neardis(2).^2);
        if D/neardis(2) > rtol || R/Ra > atol
             FNN(m,1)=FNN(m,1)+1;
        end
    end
end
FNN=(FNN./FNN(1,1))*100;
figure
plot(1:length(FNN),FNN,'-o')
grid on;
title('Minimum embedding dimension with false nearest neighbors')
xlabel('Embedding dimension','fontsize',16)
ylabel('The percentage of false nearest neighbours','fontsize',16)
----------------------------------------------------------------------------
function Y=psr_deneme(x,m,tao,npoint)
%Phase space reconstruction
%x : time series
%m : embedding dimension
%tao : time delay
%npoint : total number of reconstructed vectors
%Y : M x m matrix
% Author:"Merve Kizilkaya"
N=length(x);
if nargin == 4
    M=npoint;
else
    M=N-(m-1)*tao;
end

Y=zeros(M,m);

for i=1:m
    Y(:,i)=x((1:M)+(i-1)*tao)';
end
```

Appendix A5: Computation of Largest Lyapunov exponents using Rosenstein's algorithm

```
%%% This program computes the Largest Lyapunov exponent et. al.
%%% for a time series using Rosenstein's method and Pesin's identity.

% Input the time series and other parameters
close all, clear all clc,
embedd=input('embedding dimension = ');
tao=input('delay time = ');
dt=0.25;   % the sampling rate
t=load('exp.txt');
x=t(:,2);
maxiter=100;
meanperiod=embedd*tao*dt;
```

```
% Begin the computation
format long
tic
for l=1:embedd
    m=l;
    LLE = lyarosenstein(x,m,tao,maxiter);  % k from 2:15
    lle(l)=LLE;
end
LE = lle'
for s=1:m                   %Applying Pesin's Identity to get K-S
    if LE(s)>=0             %entropy as sum of positive lyapunov exponents
       K(s)=LE(s);
    else K(s)=0;
    end
end
LMAX = max(LE)   % largest Lyapunov exponent
KSE = sum(K)     % K-s entropy
T = 1/LMAX       % predictability
toc
-----------------------------------------------------------------------------
function  [LLE] = lyarosenstein(x,m,tao,maxiter)
% d:divergence of nearest trajectories
% x:signal
% tao:time delay
% m:embedding dimension

N=length(x);
M=N-(m-1)*tao;
Y=psr_deneme(x,m,tao);
for i=1:M
    i
    x0=ones(M,1)*Y(i,:);
    distance=sqrt(sum((Y-x0).^2,2));
    for j=1:M
        if abs(j-i)<=meanperiod
            distance(j)=1e10;
        end
    end
    [neardis(i) nearpos(i)]=min(distance);
end

for k=1:maxiter
    k
    maxind=M-k;
    evolve=0;
    pnt=0;
    for j=1:M
        if j<=maxind && nearpos(j)<=maxind
                       if dist_k~=0
                  evolve=evolve+log(dist_k);
                  pnt=pnt+1;
               end
        end
    end
    if pnt > 0
        d(k)=evolve/pnt;
```

```
    else
        d(k)=0;
    end
end
figure,
plot(d,'b.-');
hold on

%% LLE Calculation
fs=1;%sampling frequency
tlinear=1:90;
F = polyfit(tlinear,d(tlinear),1);  % linear regression and slope evaluation
R=polyval(F,tlinear);
plot(tlinear,R,'r.-');hold off
xlabel('Time(i*dt)');ylabel('Average exponential divergence[lnd(k)]');
LLE = F(1)*fs    % largest Lyapunov exponent value
```

Appendix B1: SARIMA model Identification selection

A. Trend and stationarity test

```
close all, clear all  clc,
D = load('exp.txt');   % inputs the training data set from a text file
d=D(:,2);
N=length(d);
%d=diff(d);
[h,pValue] = kpsstest(d,'lags',7:15)
plot(d)
```

B. BIC model evaluation method

```
close all, clear all  clc,
format long;
Y = load('exp.txt');
% x=xlsread('C:\Users\Abdul\Documents\data',-1); % inputs the time     %
series data set from an excel file
x=Y(:,2);
N=length(x);
D=input('Differencing factor = ');
%% fit ARMA(p,D,q) models
LOGL=zeros(4,4); % initialization
PQ=zeros(4,4);
for p=1:4
    for q=1:4
        mod=arima(p,D,q);
        [fit,~,logL]=estimate(mod,x,'print',false);
        LOGL(p,q)=logL;
        PQ(p,q)=p+q;
    end
end
%% Calculate the BIC
LOGL=reshape(LOGL,16,1);
PQ=reshape(PQ,16,1);
[~,bic]=aicbic(LOGL,PQ+1,100);
reshape(bic,4,4)
```

Appendix B2: Multiplicative seasonal ARIMA model for the forecast of daily peak load demand.

```
close all, clear all  clc, format compact
%% load the input data
y=load('exppp.txt');
Y=y(:,2);
N = length(Y);
Y1 = Y(1:N-365);
Y2 = Y(N-364:N);
%% specify multiplicative seasonal ARIMA model & estimate parameters
model = arima('Constant',0,'ARLags',2,'SARLags',2,'D',0,...
            'Seasonality',365,'MALags',2,'SMALags',2);
fit = estimate(model,Y1);

%% check the residuals for normality 'SARLags',2,'SMALags',2
res = infer(fit,Y1);
stres = res/sqrt(fit.Variance)

figure(1)
subplot(1,2,1)
qqplot(stres)

ksdensity(stres x = -4:.05:4;
[f,xi] =);
subplot(1,2,2)
plot(xi,f,'k','LineWidth',2);
hold on
plot(x,normpdf(x),'r--','LineWidth',2)
legend('Standardized Residuals','Standard Normal')
hold off
%% check the residuals for autocorrelation
figure(2)
subplot(2,1,1)
autocorr(stres)
subplot(2,1,2)
parcorr(stres)

[h,p] = lbqtest(stres,'lags',[5,10,15],'dof',[3,8,13])
%% check the predictive performance
Yf1 = forecast(fit,365,'Y0');
MSE = 1/N*sum((Y2-Yf1).^2)
RMSE = sqrt(MSE)
R=corr(Y2,Yf1)
D=R^2
figure(3)
plot(Y2,'ro-','LineWidth',2)
hold on
plot(Yf1,'k-','LineWidth',1.5)
xlim([0,365])
title('ARIMA Prediction')
legend('Observed','Forecast','Location','NorthWest')
hold off
```

Appendix B3: Algorithm for STLF of Daily Peak Load Demand of Nigeria Power System Grid based on an improved SARIMA and NARX Neural Network model

```
close all, clear all  clc, format compact
%% load the input and target data
close all, clear all  clc, format compact
%% load the input data
p=load('exp.txt');
pp=p(:,2);
s=load('expsr.txt');
sr=s(:,2);
dw=load('exp.txt');
dw=dw(:,2);
t1=load('exp.txt');
mnt=t1(:,2);
t2=load('expmaxt.txt');
mxt=t2(:,2);
%% Forecast peak power using ARIMA
Ypp=pp;
N = length(Ypp);
Y1pp = Ypp(1:N-365);
Y2pp = Ypp(N-364:N);
modelpp = arima('Constant',0,'ARLags',2,'SARLags',2,'D',0,...
            'Seasonality',365,'MALags',2,'SMALags',2);
fitpp = estimate(modelpp,'print',false);
Yfpp = forecast(fitpp,365,'Y0');
%% Forecast solar radiation using ARIMA
Ysr=sr;
N = length(Ysr);
Y1sr = Ysr(1:N-365);
modelsr = arima('Constant',0,'ARLags',2,'SARLags',2,'D',0,...
            'Seasonality',365,'MALags',1,'SMALags',1);
fitsr = estimate(modelsr,'print',false);
Yfsr = forecast(fitsr,365,'Y0');
%% Forecast wind speed using ARIMA
Ydw=dw;
N = length(Ydw);
Y1dw = Ydw(1:N-365);
modeldw = arima('Constant',0,'ARLags',1,'SARLags',1,'D',1,...
            'Seasonality',365,'MALags',2,'SMALags',2);
fitdw = estimate(modeldw,'print',false);
Yfdw = forecast(fitdw,365,'Y0);
%% Forecast minimum temperature using ARIMA
Ymt=mnt;
N = length(Ymt);
Y1mt = Ymt(1:N-365);
modelmt = arima('Constant',0,'ARLags',3,'SARLags',3,'D',0,...
            'Seasonality',365,'MALags',3,'SMALags',3);
fitmt = estimate(modelmt,Y1mt,'print',false);
Yfmt = forecast(fitmt,365,'Y0');
%% forecast maximum temperature using ARIMA
Ymx=mxt;
N = length(Ymx);
Y1mx = Ymx(1:N-365);
modelmx = arima('Constant',0,'ARLags',3,'SARLags',3,'D',0,...
            'Seasonality',365,'MALags',3,'SMALags',3);
```

```
fitmx = estimate(modelmx,Y1mx,'print',false);
Yfmx = forecast(fitmx,365,'Y0');
%% Evaluate the arima output
MSE = 1/N*sum((Y2pp-Yf1pp).^2) % calculate Root mean square (MSE)
RMSE = sqrt(MSE)                % calculate RMSE
MAPE = 1/N*sum(abs((Y2pp-Yf1pp)./Y2pp))*100  % calculate MAPE
AR=corr(Y2pp,Yf1pp)             % calculate Regression

%% Create a Nonlinear Autoregressive Network with External Input
mnt1=[Y1mt;Yfmt];
mxt1=[Y1mx;Yfmx];
dw1=[Y1dw;Yfdw];
sr1=[Y1sr;Yfsr];
pp1=[Y1pp;Yfpp];
inputSeries = [sr1 mxt1 mnt1]';
targetSeries = pp1';
inputSeries = con2seq(inputSeries);
targetSeries = con2seq(targetSeries);
inputDelays = 1:27;
feedbackDelays = 1:27;
hiddenLayerSize = [17 4];
net = narxnet(inputDelays,hiddenLayerSize);

% Preparing the Data for Training and Simulation
% using the PREPARETS function, which prepares time series data
% for a particular network, shifting time by the minimum
% amount to fill input states and layer states.

[inputs,inputStates,layerStates,targets] =
                              preparets(net,inputSeries,{},targetSeries);

% Set up Division of Data for Training, Validation, Testing
net.divideFcn = 'divideblock';
net.divideMode = 'sampletime';
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
net.trainFcn='trainlm';
net.trainParam.goal = 1e-3;
net.trainParam.min_grad = 0.001;
net.trainParam.epochs = 1000;    % denotes the maximum number of epochs to
trainan be set to the n
net.trainParam.lr = 0.01;      % denotes the learning rate
net.trainParam.max_fail =100;     % denotes the maximum validation failures

%% Create and train an ensemble of ten neural networks.
nets = cell(1,numNN);
for i=1:numNN
disp(['Training ' num2str(i) '/' num2str(numNN)])
    nets{i} = train(net,inputs,targets,inputStates,layerStates);
end
%% Next, each network is tested on the second dataset with both individual
performances and the performance for the average output calculated.
perfs = zeros(1,numNN);
outputTotal = 0;
for i=1:numNN
```

```
  neti = nets{i};
  outputs = neti(inputs,inputStates,layerStates);
  perfs(i) = perform(neti,targets,outputs);
  outputTotal = outputTotal + cell2mat(outputs);
end
%% Test the averaged network performance
AverageOutput = outputTotal/numNN
outputs = con2seq(AverageOutput);
errors=gsubtract(targets,outputs);
perfAveragedOutputs = perform(nets{1},targets,outputs)
RMSEperfavgout=sqrt(perfAveragedOutputs)
% Plot the network characteristics
figure, plotregression(targets,outputs)
figure, plotresponse(targets,outputs)
figure, ploterrcorr(errors)
figure, plotinerrcorr(inputs,errors)
Ts=Y2pp';                  % the validation data
forecastvalue=cell2mat(outputs);
N=length(forecastvalue);
Ys=forecastvalue(N-364:N);    % one year (365 days) forecast
AMSE = 1/N*sum((Ts-Ys).^2)  % calculate MSE of 365 day forecast
ARMSE = sqrt(MSE)             % calculate RMSE
AnMSE = MSE/var(Ts)           % calculate normalize MSE
MAPE = 1/N*sum(abs((Ts-Ys)./Ys))*100  % calculate MAPE
error=Ts-Ys;                  % error values

figure;    % plot the three outputs: ACTUAL, SARIMA & CONCATENATED MODEL
plot(1:length(Ts),Ts,'bs-',1:length(Yfpp),Yfpp,'ro-',1:length(Ys),Ys,'k.-
',1:length(error),error,'g-');
title('Nine years data used for training 1st Jan.2006 - 31st Dec.2014 ');
xlabel('days');
ylabel('Daily peak load demand (MW) in 2015');
legend('measured peak load','SARIMA prediction','NARX prediction','error',
'Location','Northwest');
```

Appendix B4: Forecasting peak load demand for Nigeria power system grid in 2017.

```
close all, clear all  clc, format compact
%% load the input data
p=load('exppp.txt');
pp1=p(:,2);
%p1=load('farmcastpp.txt');
p1=load('Hfarmcastx.txt');
pp2=p1(:,2);
pp=[pp1;pp2];
s=load('expsr.txt');
sr1=s(:,2);
s1=load('farmcastsr.txt');
sr2=s1(:,2);
sr=[sr1;sr2];
tm1=load('expmint.txt');
t1=tm1(:,2);
tm2=load('farmcastmn.txt');
t2=tm2(:,2);
mnt=[t1;t2];
x1=load('expmaxt.txt');
```

```
tx1=x1(:,2);
x2=load('farmcastmx.txt');
tx2=x2(:,2);
mxt=[tx1;tx2];
%% Forecast Peak Load using ARIMA
Ypp=pp;
Npp = length(Ypp);
modelpp = arima('Constant',0,'ARLags',2,'SARLags',2,'D',1,...
            'Seasonality',365,'MALags',4,'SMALags',4);
fitpp = estimate(modelpp,Ypp,'print',false);
repp = infer(fitpp,Ypp);
Yfpp = simulate(fitpp,365,'NumPaths',5,'Y0',Ypp,'E0',repp);
Yfpp = mean(Yfpp,2);

%% Store the forecasted peak load values
fids=fopen('farmcastpp1.txt','w+');
yDatas = Yfpp;
ns=length(yDatas);
xDatas = transp(1:ns);
for xs = 1:ns              % Input the Julian day as a vector
    ys = yDatas(xs);
    count = fprintf(fids,'%15.5f',xs,ys);
    fprintf(fids,'\n');
end

%% Forecast solar radiation using ARIMA
Ysr=sr;
Nsr = length(Ysr);
modelsr = arima('Constant',0,'ARLags',2,'SARLags',2,'D',0,...
            'Seasonality',365,'MALags',2,'SMALags',2);
fitsr = estimate(modelsr,Ysr,'print',false);
resr = infer(fitsr,Ysr);
Yfsr = simulate(fitsr,365,'NumPaths',5,'Y0',Ysr,'E0',resr);
Yfsr = mean(Yfsr,2);

%% Store the forecasted solar radiation values
fids=fopen('farmcastsr1.txt','w+');
yDatas = Yfsr;
ns=length(yDatas);
xDatas = transp(1:ns);
for xs = 1:ns              % Input the Julian day as a vector
    ys = yDatas(xs);
    count = fprintf(fids,'%15.5f',xs,ys);
    fprintf(fids,'\n');
end
%% Forecast minimum temperature using ARIMA
Ymt=mnt;
Nmt = size(Ymt)
modelmt = arima('Constant',0,'ARLags',3,'SARLags',3,'D',0,...
            'Seasonality',365,'MALags',1,'SMALags',1);
fitmt = estimate(modelmt,Ymt,'print',false);
resmt = infer(fitmt,Ymt);
Yfmt = simulate(fitmt,365,'NumPaths',5,'Y0',Ymt,'E0',resmt);
Yfmt = mean(Yfmt,2);

%% Store the forecasted minimum temp. values
fidm=fopen('farmcastmn1.txt','w+');
```

```
yDatam = Yfmt;
nm=length(yDatam);
xDatam = transp(1:nm);
for xm = 1:nm              % Input the Julian day as a vector
    ym = yDatam(xm);
    count = fprintf(fidm,'%15.5f',xm,ym);
    fprintf(fidm,'\n');
end

%% forecast maximum temperature using ARIMA
Ymx=mxt;
Nmx = size(Ymx)
modelmx = arima('Constant',0,'ARLags',3,'SARLags',3,'D',0,...
            'Seasonality',365,'MALags',1,'SMALags',1);
fitmx = estimate(modelmx,Ymx,'print',false);
resmx = infer(fitmx,Ymx);
Yfmx = simulate(fitmx,365,'NumPaths',5,'Y0',Ymx,'E0',resmx);
Yfmx = mean(Yfmx,2);

%% Store the forecasted maximum temp. values
fidx=fopen('farmcastmx1.txt','w+');
yDatax = Yfmx;
nx=length(yDatax);
xDatax = transp(1:nx);
for xx = 1:nx              % Input the Julian day as a vector
    yx = yDatax(xx);
    count = fprintf(fidx,'%15.5f',xx,yx);
    fprintf(fidx,'\n');
end
%% Create a Nonlinear Autoregressive Network with External Input
mnt1=[Ymt;Yfmt];
mxt1=[Ymx;Yfmx];
sr1=[Ysr;Yfsr];
pp1=[Ypp;Yfpp];
Nh = length(Ypp);
pp2=Ypp(1:Nh-365);
pp3=Ypp(Nh-364:Nh);
inputSeries = [mnt1 mxt1 sr1]';
targetSeries = pp1';
inputSeries = con2seq(inputSeries);
targetSeries = con2seq(targetSeries);
inputDelays = 1:27;
feedbackDelays = 1:27;
hiddenLayerSize = [15 4];
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize);

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares time series data
% for a particular network, shifting time by the minimum
% amount to fill input states and layer states.
% Using PREPARETS allows you to keep your original
% time series data unchanged, while easily customizing it
% for networks with differing numbers of delays, with
% open loop or closed loop feedback modes.
[inputs,inputStates,layerStates,targets] =
preparets(net,inputSeries,{},targetSeries);
```

```
% Set up Division of Data for Training, Validation, Testing
net.divideFcn = 'divideblock';
net.divideMode = 'sampletime';
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
net.trainFcn='trainlm';
net.trainParam.goal = 1e-3;
net.trainParam.min_grad = 0.001;
net.trainParam.epochs = 1000;    % denotes the maximum number of epochs to
trainan be set to the n
net.trainParam.lr = 0.01;      % denotes the learning rate
net.trainParam.max_fail =100;     % denotes the maximum validation failures

%% Train the Network, Then, ten neural networks are trained.
numNN = 2;
nets = cell(1,numNN);
for i=1:numNN
disp(['Training ' num2str(i) '/' num2str(numNN)])
    nets{i} = train(net,inputs,targets,inputStates,layerStates);
end
%% Next, each network is tested on the second dataset with both individual
performances
% and the performance for the average output calculated.
perfs = zeros(1,numNN);
outputTotal = 0;
for i=1:numNN
  neti = nets{i};
  outputs = neti(inputs,inputStates,layerStates);
  perfs(i) = perform(neti,targets,outputs);
  outputTotal = outputTotal + cell2mat(outputs);
end
%% Test the network perfs
AverageOutput = outputTotal/numNN;
outputs = con2seq(AverageOutput);
errors=gsubtract(targets,outputs);
perfAveragedOutputs = perform(nets{1},targets,outputs)
RMSEperfavgout=sqrt(perfAveragedOutputs)
figure, plotregression(targets,outputs)
figure, plotresponse(targets,outputs)
figure, ploterrcorr(errors)
figure, plotinerrcorr(inputs,errors)
forecastvalue=cell2mat(outputs);
N=length(forecastvalue);
Ys=forecastvalue(N-364:N);
MSEhybrid=mean((Yfpp-Ys').^2)
RMSEhybrid = sqrt(MSEhybrid)
figure;
plot(1:Nh-365,pp2.*100,'k-','LineWidth',1.5)
hold on
plot(Nh-364:Nh,pp3.*100,'r-','LineWidth',1.5)
hold on
plot(Nh+1:Nh+365,Ys.*100,'go-','LineWidth',1.5)
hold on
plot(Nh+1:Nh+365,Yfpp.*100,'b-','LineWidth',1.5)
hold off
xlabel('days');
```

```
ylabel('Daily Peak Generation (MW)');
legend('Actual data from 2005-2015',' Cancatenation Model Forecast for
2016',...
        'SARIMA forecast for 2017','Conatenation Model Forecast for
2017','Location',...
        'SouthWest')
xlim('auto');
%% Store the forecasted values
fid=fopen('Hfarmcast1.txt','w+');
yData = Ys;
n=length(yData);
xData = transp(1:n);
for x = 1:n              % Input the Julian day as a vector
    y = yData(x);
    count = fprintf(fid,'%15.5f',x,y);
    fprintf(fid,'\n');
end
```