Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Electrical energy demand forecasting model using artificial neural network: A case study of Lagos State Nigeria

Khadeejah Adebisi Abdulsalam[a] and Olubayo Moses Babatunde[a*]

[a]*Department of Electrical & Electronics Engineering, University Of Lagos, Akoka, Nigeria*

CHRONICLE

Article history:
Received: January 01, 2019
Received in revised format: March 6, 2019
Accepted: May 24, 2019
Available online: May 24, 2019

Keywords:
Artificial Neural Network
Electrical Energy Demand Forecasting
Recurrent Neural Network

ABSTRACT

Electrical Energy is an essential commodity which significantly contributes to the economic development of any country. Many non-linear factors contribute to the final output of electrical energy demand. In order to efficiently predict electrical energy demand, many time-series analysis and multivariate techniques have been suggested. In order for these methods to accurately work, an enormous quantity of historical dataset is essential which sometimes are not available, inadequate and inaccurate. To overcome some of these challenges, this paper presents an Artificial Neural Network based method for Electrical Energy Demand Forecasting using a case study of Lagos state, Nigeria. The predicted values are compared with actual values to estimate the performance of the proposed technique.

1. Introduction

Modern science and engineering studies use models to describe physical, biological and social systems and experimental data are used to verify and estimate such models. However, in many real life systems, the underlying systems are either unknown or sometimes the systems are too complex for concise mathematical representation. Coincidentally, increasing use of computers, continuous development in database technology and low cost sensors have made it possible to capture data generated from systems. These data can be used to derive system models by estimating relationship between the input and output components of the system. Examples of such complex systems include medical diagnosis, handwritten character recognition, and time series prediction. Developing models from data based on methodologies such as artificial neural network, fuzzy systems, genetic algorithms, expert systems and wavelet has been inspired by the learning capabilities of biological systems which learn through data-driven interaction with the environments. Specific learning tasks includes classification or estimation of class decision boundary, regression and probability density estimation from samples. Basically, there are two common types of the learning problems viz: supervised and unsupervised. In supervised learning, a set of target

[*] Corresponding author.
E-mail address: kabdulsalam@unilag.edu.ng (O. M. Babatunde)

doi: 10.5267/j.ijdns.2019.5.002

of interest is provided by an external teacher; this target may take the form of a desired input-output mapping which the network is required to approximate. Classification and regression tasks are typical examples of supervised learning. However, unsupervised learning or self organising system discovers significant patterns or features in the input data without a teacher. The learning algorithm is provided with a set of local rules which enables it to learn to compute an input-output mapping with specific desirable properties (Vladimir & Fillip, 2007; Haykin, 1999). The are two main stages of a learning system viz: learning/estimation (from training samples) and operation/prediction-when predictions are made for future or test samples (Vladimir & Fillip, 2007). Artificial Neural Network (ANN) have found great application in modelling unknown input-output relations in engineering and its use in critical mission applications is increasing by the day especially with the emergence of new network models and user friendly ANN software (Murata et al., 1994; Rodvold, 1999). There are three main activities in deploying ANN viz. ANN model proposal, training the proposed ANN model with data, and verification, validation and evaluation of the trained ANN before deployment. This paper focuses on a numerical model of utility critical ANN for predicting electrical energy demand, the data format/structure for the training and a highlight of MATLAB implementation of the proposed model.

1.1 Neural Network Development Process

Although, deploying ANN can be tricky because of the experimental nature of its construction and the black box label concept associated with it, nevertheless, research efforts have produced software development process model for ANN development. This development process model is as shown in Fig 1. These phases are iterative until the requirements are met. The focus of this work is to derive a neural network architecture model for predicting electricity consumption in Lagos state based on the available data and, also to prepare the data in the required format for training the derived network model (Fig 1).

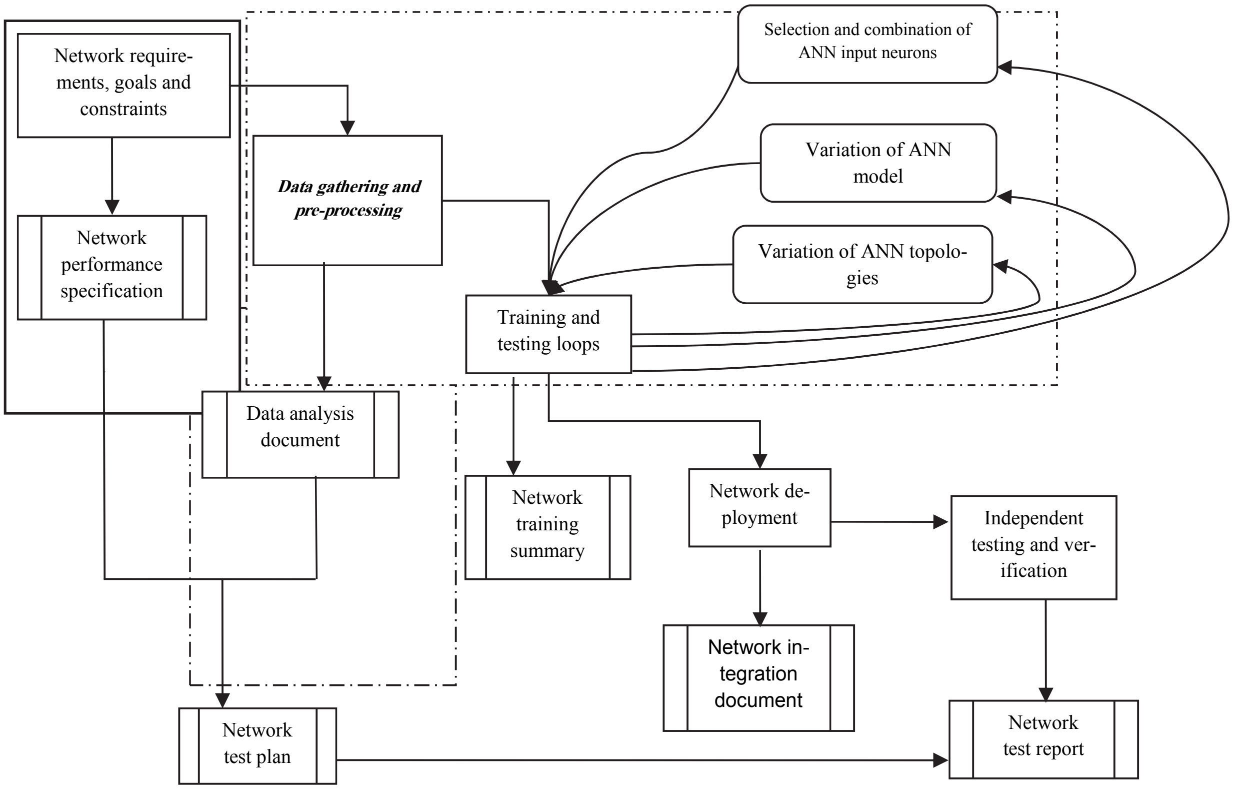
Network requirements, goals and constraints
Network performance specification
Data gathering and pre-processing
Selection and combination of ANN input neurons
Variation of ANN model
Variation of ANN topologies
Training and testing loops
Data analysis document
Network training summary
Network deployment
Independent testing and verification
Network integration document
Network test plan
Network test report

Fig. 1. Neural Network Development Process

2. Neural network model

A neural network is a collection of parallel distributed processors comprising of simple processing units that has a tendency to store experiential knowledge and making it available for use (Haykin, 1999), (Jain & Mao, 1996), (Fausett, 1994). It resembles the brain in two aspects:

- Knowledge is acquired by the network from its environment through a learning process-this is equivalent to training the network.
- Interneuron connection strengths known as synaptic weights are used to store the acquired knowledge- the acquired knowledge is used for generalization.

Neural network has capability to adjust its topology, which is equivalent to the fact that the human brain cells die, and new ones are born.

2.1 Structure of Neural Network

The neuron is the fundamental information processing unit in a neural network. It has three basic elements as shown in the Fig. 2. A neuronal model includes an externally applied bias b_k used to increase or decrease the net input of the activation function (Haykin, 1999, Jain & Mao, 1996; Fausett, 1994; Mandic & Chambers, 2001).

i. *A set of synapses/connecting links* characterised by a weight/strength of its own. A signal x_j at the input of a synapse j connected to neuron k is multiplied by the synaptic weight w_{kj} which lie in a range that includes both positive and negative values.

ii. *Combinational function:* each processing unit in a neural network performs some mathematical operations on it input values via synaptic connections from other units. The result of this is known as activation potential. Commonly used combination functions are linear product combination functions used in multilayer perceptron and recurrent neural network, the Euclidean function used in radial basis function etc. Adder, multiplier and delay are commonly used to realise combination function. Eq. (1) is a mathematical representation of the combinational function.

iii. *Activationfunction:* neural network map their activation potential provided by the combination function onto the output of a neuron using a scalar function called a nonlinear activation function. The entire functional mapping performed by a neuron i.e. the combination of combination function and a nonlinear activation function is called the transfer function of a neuron : $\mathbb{R}^N \to \mathbb{R}$. Activation functions are also used to limit the amplitude of the neuron output. Non-linear activation functions with a bounded range are called squashing functions e.g. the *tanh*and logistic function. Activation function for neural network neurons must be non-linear to form universal approximator because neural networks are non-linear processors. The activation function must be centred round a certain value in the output space and in order to perform an efficient prediction, the range of the input data, mean, variance must match with the range of the chosen activation function. Activation function of a neuron may also defined by probability of the excitation of the state of the neuron. The output of the activation function is the output of a neuron and it is depicted in equation *(2).* Typical examples of activation functions include:

- The hard-limiter heaviside (step) function defined as:

$$H(x) = \begin{cases} 0, x \leq \theta, \\ 1, x > \theta, \end{cases}$$

$$where \theta is some threshold$$

- Sigmoid/Logistics function defined as: $\sigma(x) = \frac{1}{1+e^{-\beta x}}$ and $f(x) = \frac{1}{1+e^{-(x^2+bias)}}$

- Gaussian sloped activation is a difference of two sigmoid activation functions and defined as: $\sigma_1(x) = \tanh(\beta x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}$; $\sigma_2(x) = \frac{2}{\pi}\tan^{-1}\left(\frac{1}{2}\pi\beta x\right)$ *and*

$$\sigma_3(x) = \frac{x^2}{1 + x^2} sgn(x)$$

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

$$y_k = \sigma(u_k + b_k) \tag{2}$$

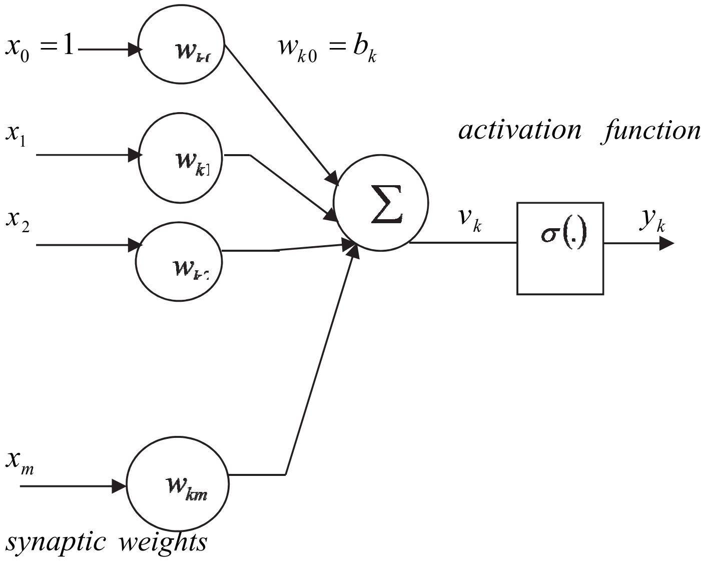

Fig. 2. Non-Linear Model of a Neuron

where x_1, x_2,...,$x_n$are the input signals, w_{k1}, w_{k2},...,w_{kn} are the synaptic weight of the neuron, u_k is the combination function due to input, b_k is the bias, $\sigma(.)$ is the activation function and y_k is the output of the neuron. The relationship between the activation function v_k and the output of the combination function $u_{k,}$ the weights due to the synapses and the bias b_k are shown in equations *(3-4).* A fully connected feed-forward network with one hidden layer is shown in fig. 3 and a recurrent network is shown in Fig. 4.

$$v_k = u_k + b_k = \sum_{j=0}^{m} w_{kj} x_j \tag{3}$$

$$y_k = \sigma(v_k) \tag{4}$$

2.2 Architecture of ANN Models for Prediction

The fundamental building blocks for linear predictors are adder, delays and multipliers while non-linear predictors also use zero memory non-linearity. An adder or a summer sums all the components at its input, a multiplier or a scaler outputs the product of its inputs while a delay acts as memory. Predictors that do not use feedback are known as moving average (MA) while those with feedback are known as autoregressive moving average structures (ARMA) (Mandic & Chambers, 2001. Haykin, Adaptive Filter Theory, 1996; Cichocki & Amari, 2002). Linear predictors defined in Eq. (5) can be used when the data source is from linear system and feedback is not required while Eq. (6) is suitable when feedback is required.

$$y(k) = \sum_{j=0}^{q} b_j\, e(k-j) \tag{5}$$

$$y(k) = \sum_{i=1}^{p} a_i y(k-i) + \sum_{j=0}^{q} b_j e(k-j) \tag{6}$$

Eq. (6) is a constant difference equation (equations that recursively define a sequence) which is a general form of (ARMA *q,p*),where *y(k)* is the output, *e(k)* is the input; $a_i|i=1,2,\dots,p$ are the (AR) feedback coefficients and $b_j|j=0,1,2,\dots,q$ are the (MA) feedforward coefficients.

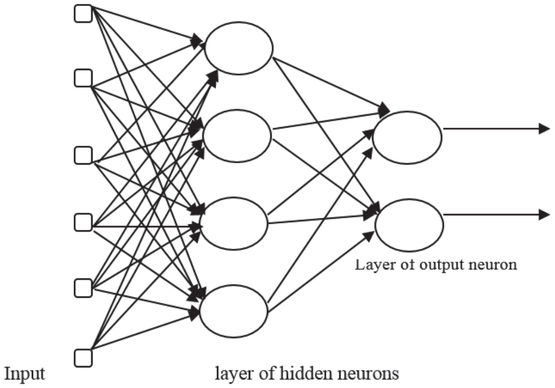

Fig. 3. A fully connected feedforward network

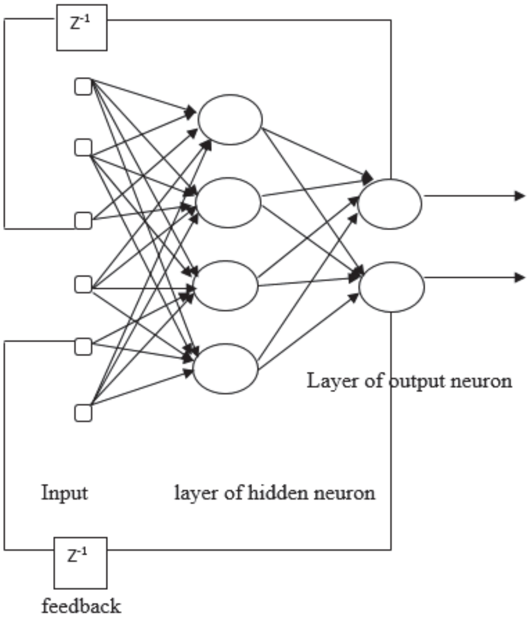

Fig. 4. A Recurrent Network with Output Feed back

If data are generated from non-linear system described by Eq. (7), the non-linear predictor can be derived from Eq. (6) to obtain Eq. (8)

$$\hat{y}(k) = E[y(k)|y(k-1), y(k-2), \dots, y(0)] \tag{7}$$

$$\hat{y}(k) = \sum_{i=1}^{p} a_i y(k-i) + \sum_{j=1}^{q} b_j \hat{e}(k-j) \tag{8}$$

where $\hat{e}(k-j) = y(k-j) - \hat{y}(k-j), j = 1,2,\dots,q$ is the residual.

Therefore, a general form of non-linear ARMA (NARMA *p,q*) can be defined as Eq. (9)

$$y(k) = \sigma(\sum_{i=1}^{p} a_i y(k-i) + \sum_{j=1}^{q} b_j e(k-j)) + e(k) \qquad 9$$

$where\ \sigma(.)$is the activation function and $e(k)$is the error due to unobservable input.

A (NARMAX *p,q,r*) models with exogenous inputs *u(k-s), s=1,2,…,r* defined by *(10)*has its associated predictor in Eq. (11).

$$y(k) = \sigma(\sum_{i=1}^{p} a_i y(k-i) + \sum_{j=1}^{q} b_j e(k-j) + \sum_{s=1}^{r} c_s u(k-s)) + e(k) \qquad (10)$$

$$\hat{y}(k) = \sigma(\sum_{i=1}^{p} a_i y(k-i) + \sum_{j=1}^{q} b_j \hat{e}(k-j) + \sum_{s=1}^{r} c_s u(k-s)) \qquad (11)$$

Application of Artificial Neural Network in forecasting, signal processing and control require analysis of dynamics associated with the input data. Feedforward networks capture the dynamic by including past inputs in the input vector. However, dynamical modelling of complex system requires feedback in the form of recurrent neural network. Four general architecture of ANN for prediction have been identified in Eqs. (12-15) (Mandic & Chambers, 2001), (Tsoi & Back, 1997).

i. The output *y(k)* is a linear function of previous outputs and a non linear function of previous inputs defined as

$$y(k) = \sum_{j=1}^{N} a_j(k) y(k-j) + F\big(u(k-1), u(k-2), \dots u(k-M)\big) \qquad (12)$$

where *F(.)* is a non-linear function.

ii. The output *y(k)* is a non-linear function of past outputs and a linear function of past inputs

$$y(k) = F\left(y(k-1), y(k-2), \dots y(k-N) + \sum_{i=1}^{M} b_i(k) u(k-i)\right) \qquad (13)$$

iii. The output *y(k)* is a non-linear function of both past inputs and outputs. The functional relationship between the past inputs and outputs can be expressed in a separate manner as

$$y(k) = F\big(y(k-1), \dots, y(k-N)\big), +G\big(u(k-1), \dots, u(k-M)\big) \qquad (14)$$

iv. The output *y(k)* is a non-linear function of past inputs and outputs and is defined by

$$y(x) = F\big(y(k-1), \dots, y(k-N), u(k-1), \dots, u(k-M)\big) \qquad (15)$$

2.3 Recurrent Neural Network Architecture for Forecasting

Recurrent Neural Network (RNN) is a multilayer artificial neural network (ANN) with feedback loop. It makes use of unit delay elements z^{-1} which results in a non-linear dynamical behaviour as shown in fig. 4. Presence of feedback loops, with delay introduces memory into the network and makes it appropriate for prediction (Mandic & Chambers, 2001; Fausett, 1994; Haykin, Neural Networks: A Comprehensive Foundation, 1999). Due to this memory, at each time instant, the network is presented with the raw possibly noisy external input data

s(k),s(k-1),…s(k-m) and

$u_i^T(k)=[s(k-1),\dots,s(k-p),1,y_1(k-1),y_2(k-1),\dots,y_N(k-1)]$ and filtered data *$y_1(k\text{-}1),\ldots y_N(k\text{-}1)$* from the network output. This filtered input history offers an improved processing performance in RNN compared with feedforward NN. RNN is appropriately suitable to forecast electricity in a developing economy because there are no existing model structures; the electricity demand data are noisy and incomplete because of shortage in supply and the future may not necessary follow the past because developing economies lack institutional structures (Haykin, Neural Networks: A Comprehensive Foundation, 1999; Mandic & Chambers, 2001; Bhattacharyya & Timilsina, 2009). In the light of the fore going, a RNN based NARMA will be used in this study.

NARMAX RNN

This is a common Williams-Zipser type RNN consisting of only two layers (output and hidden layer), the input layer of feedforward and feedback signals. The general form of this model is approximately equal to equation (16)

$$\hat{y}(k) = \sigma\Big(\big(x(k-1), x(k-2), \ldots x(k-p), y_1(k-1), \ldots, y_1(k-q)\big)\Big) \tag{16}$$

The non-linearity is dependent on both the non-linearity associated with the output neuron and non-linearity in the hidden neuron.

2.3.1 The RNNTopology

There are two ways to implement recurrent connections in ANN viz. activation feedback and output feedback as shown in Fig. 5 a and b and described in Eqs. (17-20).

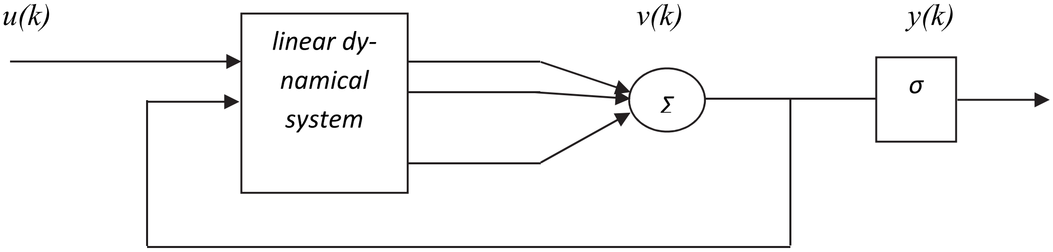

Fig. 5a. activation feedback scheme

The output of a neuron for activation feedback scheme is

$$v(k) = \sum_{i=0}^{M} w_{u,i}(k)u(k-i) + \sum_{j=1}^{N} w_{v,j}(k)v(k-j) \tag{17}$$

$$y(k) = \sigma\big(v(k)\big) \qquad 18$$

where $w_{u,i}$ and $w_{v,j}$ are the weights associated with *u* and *v*.

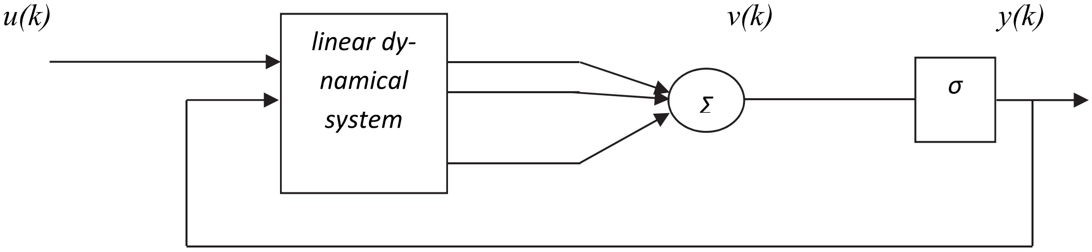

Fig. 5b. Output feedback scheme

The output feedback scheme can be defined as

$$v(k) = \sum_{i=0}^{M} w_{u,i}(k)u(k-i) + \sum_{j=1}^{N} w_{y,j}(k)y(k-j) \tag{19}$$

$$y(k) = \sigma\big(v(k)\big) \tag{20}$$

where $w_{y,j}$ is the weight associated with the delayed output.

Other available RNN topologies in literature include modular and hybrid architectures, pipelined RNN, Elman RNN, and Jordan RNN (Mandic & Chambers, 2001), (Tsoi & Back, 1997).

2.4 Size of ANN

Determining the size of Neural Network entails finding the number of hidden units in the network, optimal weight vector for the neurons and optimum network configuration required to estimate a function that best describes the available data to a high accuracy. The object is the size of network required for valid generalisation when m example of training data is provided and accuracy parameter ε is expected such that a feed forward neural network can predict correctly at least fraction $(1-\varepsilon)$ of future examples drawn from the same distribution (Lappas, 2007; Baum & Haussler, 1989; Murata et al., 1994). The concept of capacity, measured as the maximum number of dichotomies that can be induced on m inputs has been shown to hold for the problem of valid generalisation for arbitrary learning problem. It is related to the number of training sample and the formula for the upper bound size *S* which is the computational units is defined in Eq. (21) as

$$S = 8 \times \sqrt{\frac{2^n}{n}} \tag{21}$$

This *S* has been found to be sufficient for small error rates where n is the number of bits required to enumerate all existing training data and is defined as Eq. (22)

$$n = \log|S_L| \tag{22}$$

and S_L is the number of the existing training data.

Theoretically, the lower and upper bound of Vapnik-Chervonenkis (VC) dimension, which is related to the number of weights (W), and computational unit *(N)* in a feed forward neural network architecture has been performed on some networks. The VC is bounded by Eq. (23)

$$W \log_2 W \leq VC \leq 2 * W * \log_2(e * N) \tag{23}$$

The problem of estimating the smallest network in terms of hardware that can describe an arbitrary function given a set of m vectors in n dimension is the circuit complexity problem. The circuit complexity theory is used to classify Boolean function according to the amount of computational resources required to compute them in terms of size and depth where the size of a circuit is the number of non-input gate it has and the depth is the length of the longest path from an input gate to the output gate.

3. Data structure

Data structure is the format of representing data in a data set used for neural networks training (Beale et al., 2010). There are two basic types: Concurrent and Sequential data vectors. Concurrent data vectors are data that have no particular time sequence and they are used in static network. They are usually presented to the network as a single matrix and the network produces a single matrix of concurrent vectors as output. The output results of these networks are equivalent to the output results of parallel networks with each network receiving only one of the input vectors. Sequential vectors however represent data whose time of occurrence is important when used in network training. Sequential inputs are therefore, presented to the network as elements of a cell array and the output is elements of cell array. When neural networks deal with data sets that contains several different sequences at the same time, a concurrent set of sequential vectors are used in which the data are presented as a cell of array where each element of the array contains elements of the sequence that occur at the same time.

3.1 Data Description, Preparation and Pre-processing

As shown in some earlier works (Vladimir & Fillip, 2007; Haykin, Neural Networks: A Comprehensive Foundation, 1999; Konstantinos, 2002; Kevin, 2001), certain transformation or operations need to be carried out on the data to condition it for training a NN. The applicable processes are:

3.1.1 Data Description

Data collection includes assembling all the data that will be used in training the neural network and these data collected need to be analyzed before any training method is employed. Data analysis aims at answering important questions about the process under investigation with regards to the statistical parameters (mean, variance, standard deviation); nature of the process (random, chaotic, periodic, stable, linear or non-linear); data distribution in the problem space (clustered, sparse, uniformly distributed); and the problem of missing data etc. Data, which is also known as variable in a data set, is considered as an individual entity and the data set is considered as a whole noting the interactions and the interrelationships between individual data/variables. The data for this study is a time series data (a sequence of observations which are ordered in time or space) for the period 1971 to 2009 obtained from three sources – National Population Commission (Lagos Office); the World Bank Database[1] and the Global Temperature Database[2]. The data consists of the energy audit variable represented by the Electric power consumption (kWh), Socio-economic variables (Gross Domestic Product GDP measured in local currency, GDP growth rate, Inflation rate as a measure of the rate of change of the purchasing power of the income), Demographic factors (total population of Lagos, population annual growth rate) and measure of the impact of climate or seasonality as represented by the daily temperature readings of Lagos.

3.1.2 Data Preparation: the data preparatory steps include

Outlier Removal: 95% of normally distributed data set lies within two standard deviations of mean. Discarding values outside the range is a simple method for removing outliers, which can have an effect on the network. Removing outliers can produce a network with smoother learning curve. Quantity Checks: the more variable a model contains, the more training data points are required. Problems of quantity check may be overcome by either enlarging the data set or reducing its dimensionality. In addition, missing data will be regressed or a moving average value substituted for it as the case may be. Quality Check: even distribution of training samples must be ensured in order to build a well-balanced model.

3.1.3 Data Pre-Processing

Data Normalization: is a type of data scaling in which data is transformed to an index of the range 0 to 1 by following the steps below:

[1] http://data.worldbank.org/country/nigeria

[2] http://gcmd.nasa.gov/KeywordSearch/Metadata.do?Portal=GCMD&KeywordPath=&NumericId=20062&MetadataView=Data&MetadataType=0&lbnode=mdlb1

I. For any series X_{it} such that each element $x_{it} \geq 0 \forall x_{it}$, let x_{it}^{min} denote the minimum value in X_{it}, and $x_{it}^{max}$the maximum value in X_{it}. Therefore, the resultant indexδ, is calculated as

$$\delta = \left(x_{it} - x_{it}^{\min}\right)\left(x_{it}^{max} - x_{it}^{min}\right)^{-1}$$

and

II. For any series X_{it} such that $0 \leq x_{it} < 1$,

let $\left[x_{it} + \left|x_{it}^{min}\right|\right] = \chi_{it}$

$$\therefore \delta = \chi\left(\chi_{it}^{max} - \chi_{it}^{min}\right)^{-1}$$

where χ_{it}^{max} and χ_{it}^{min} are defined as the maximum and minimum value respectively.

Dimensionality Reduction: Dimensionality Reduction is a process whereby a data space is transformed into a feature space that in theory has exactly the same dimension as the original data space such that the input data falls within the range of the activation function that will be employed before it is used for training. This transformation is such that the data set may be represented by a reduced number of effective features, which may be binary, categorical or continuous, however, it is domain specific and usually related to the available measurements. It may not alter the space dimensionality and others may enlarge it, however, it retains most of the intrinsic information content of the data. Dimensionality reduction enhances the accuracy of the data, speed up computation, reduces the problem of over fitting, enhances the understanding of data and reduces measurement and storage requirement. Dimensionality reduction techniques can be categorized into two classes: feature extraction and feature selection (Haykin, Neural Networks: A Comprehensive Foundation, 1999; Mandic & Chambers, 2001; Zaman & Fakhri, 2009; Guyon et al., 2006; Dy, 2008).

Feature Extraction is the production of a new set of feature from the original features in the data through the application of some mapping techniques. The dominant feature extraction techniques are the principal component analysis (PCA) and the linear discriminant analysis (LDA).

Feature Selection selects the best subset of the original features. It reduces the number of features and removes irrelevant, redundant, or noisy data. Wrapper and filter technique are the most important feature selection schemes.

4. Model development

4.1 Data Preparation and Processing

The data for training the network is a data set of six variables containing population *P*, temperature *T*, energy consumption E_c, economy variable E_v as represented by the GDP and estimated energy demand E_d value. The sixth is the random variable R_v, which will be the error term found from the regression of *P, T,* E_c*,* E_v over E_d. Data for training is divided to three in ratio 14:3:3 for training, testing and validating.

The pre-processing and post processing stages are an integral part of any RNN architectural model (Tsoi & Back, 1997), however data required for training must be tested for non-linearity and probably transform it to non-linear form. Theoretically, population models are represented by non-linear equations, weather are defined by basic hydrodynamic and thermodynamic non-linear equations representing the behaviour of the atmosphere. Economic value is dependent on many variables, which do not have linear relationships (Robert, 2006; Jose & Abraham, 1992).

Outlier Removal: Only the temperature data contains outlier (-99) which are mainly because the data were not captured (a case of missing data). For this, the average of previous three years for the particular month and three years forward where applicable are used.

Missing data: the data available for electricity consumption applies to the whole country, but during the course of this work, data for Lagos state were obtained from Ikeja and Eko distribution companies. In addition, data were obtained from PHCN HQ in Abuja and National Control Centre (NCC) Oshogbo. These institutional data only cover from 1999-2008 in case of the PHCN HQ and NCC; data from Ikeja and Eko distribution are for only 2 years (2007-2008). Interpolating these data shows that Lagos accounts for consumption of 22.15% of the total consumption and this was subsequently used to approximate energy consumption in Lagos from the country consumption during the period.

According to Okoye (2007), less than 40% of Nigerians are connected to the grid for electricity supply; and for the connected few, power supply is usually for less than 60% on the average. Also, the highest energy per capital of Nigeria in the last ten years is 27.88W/hr in 1999 against 297W/hr for the World in 2005[3]. The total energy that should be demanded was calculated by simple proportion.

4.2 Data structure: MATLAB implementation

Each data set will be presented to the network as a cell array of sequential concurrent data and the output as sequential data such that training is carried out on a set at an instance. Dummy variables D_v, are introduced because MATLAB requires ten time step values for each data set.

Input={$[P][T][E_c][E_v][R_v][D_v][D_v][D_v][D_v][D_v]$}

Output= {$[E_d][D_v][D_v][D_v][D_v][D_v][D_v][D_v][D_v][D_v]$}

4.3 MATLAB Implementation for Testing for Non-Linearity of Data

The data to be used for training was tested for non-linearity and non-linearly transform the data using the curve fitting curves toolbox in MATLAB.

Temperature: The temperature data was fitted with non-parametric model whose objective is to draw a curve through the data using the smoothing spline that minimizes equation *(24)* as shown in fig. 6.

$$p\sum_i w_i\big(y_i - s(x_i)\big)^2 + (1+p)\int\left(\frac{d^2s}{dx^2}\right)^2 dx \tag{24}$$

[3] http://en.wikipedia.org/wiki/List_of_countries_by_electricity_consumption

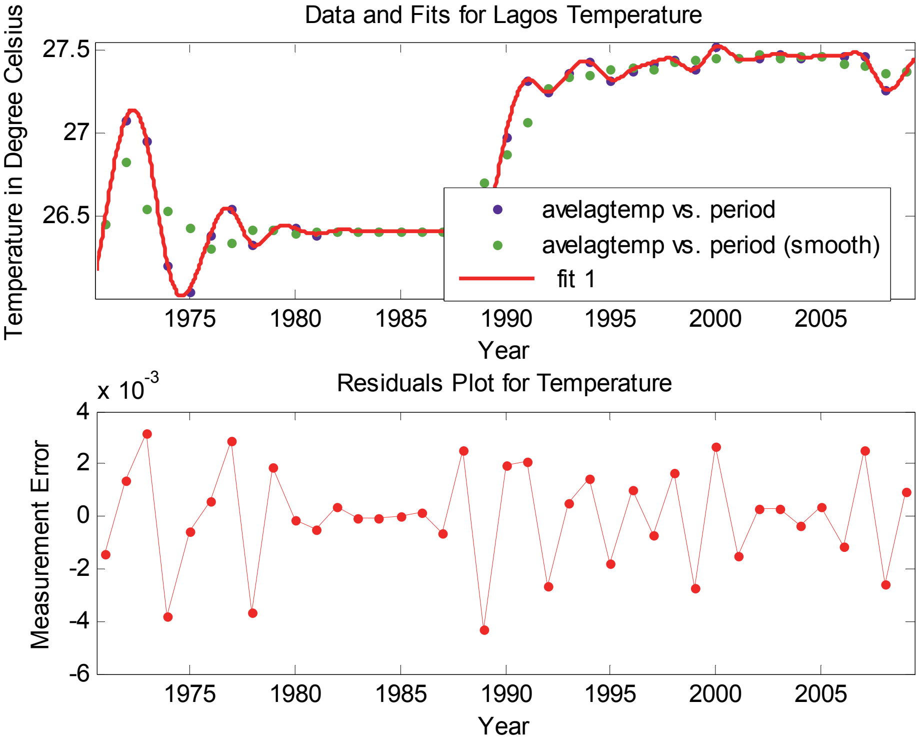

Fig. 6. Graph showing data and fit for Lagos Temperature

Fit analysis: Smoothing spline: *f(x)* = piecewise polynomial computed from *p*

Smoothing parameter: *p = 0.9984;SSE: 0.0001383;R-square: 1;Adjusted R-square: 0.9993; RMSE: 0.01315*

The residual also shows a good fit for the data because it displays a randomly scattered point evenly distributed around zero and equation *(24)* is a representative model for the data

Lagos Population:

The exponential model fitting function was employed because the rate of change of biological population is a function of its initial value and the two-term exponential model of *(25)* was used. This is shown in Fig. 7.

$$f(x) = ae^{bx} + ce^{dx} \tag{25}$$

where the standard deviation of *x* is 11.4 with coefficients (with 95% confidence bounds):

a =	*4058(-4.881e+004, 5.693e+004);*	*b =*	*-1.967(-9.675, 5.742)*
c =	*5.773e+006(5.68e+006, 5.867e+006);*	*d =*	*0.3232(0.3097, 0.3367)*

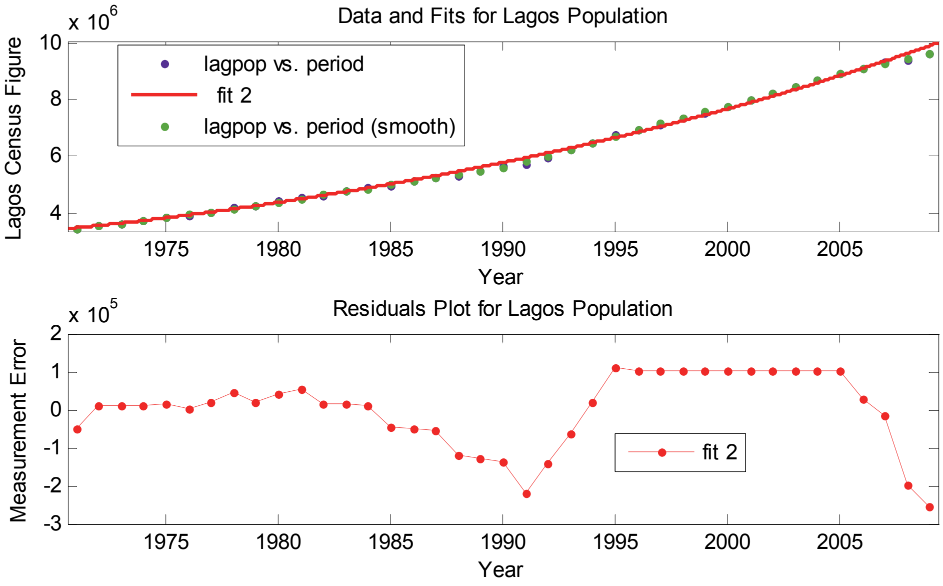

Fig. 7. Graph Showing Smoothed Population Figure From 1970-2009: this curve fitting shows the non-linearity of the data and equation *(25)* is a mathematical representation of the data.

Fit Analysis

SSE: 3.639e+011, R-square: 0.9974, Adjusted R-square: 0.9972, RMSE: 1.02e+005

Electricity Consumption: The actual electricity consumed from 1970-2009 is fitted using the Gaussian model of Eq. (26) as shown in Fig. 8, because electricity consumption peaks.

$$f(x) = a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2} + a_3 e^{-\left(\frac{x-b_3}{c_3}\right)^2} \tag{26}$$

with coefficients (with 95% confidence bounds):

a_1 = 3.115e+006(-3.789e+009, 3.796e+009); b_1 = 1999(903.5, 3095)

c_1 = 0.2874(-365.5, 366.1); a_2 = 2.958e+006(2.292e+006, 3.623e+006)

b_2 = 2006(2006, 2007); c_2 = 5.997(4.766, 7.227)

a_3 = 1.808e+00 (1.64e+006, 1.976e+006); b_3 = 1994(1990, 1998)

c3 = 17.05(13.11, 20.99)

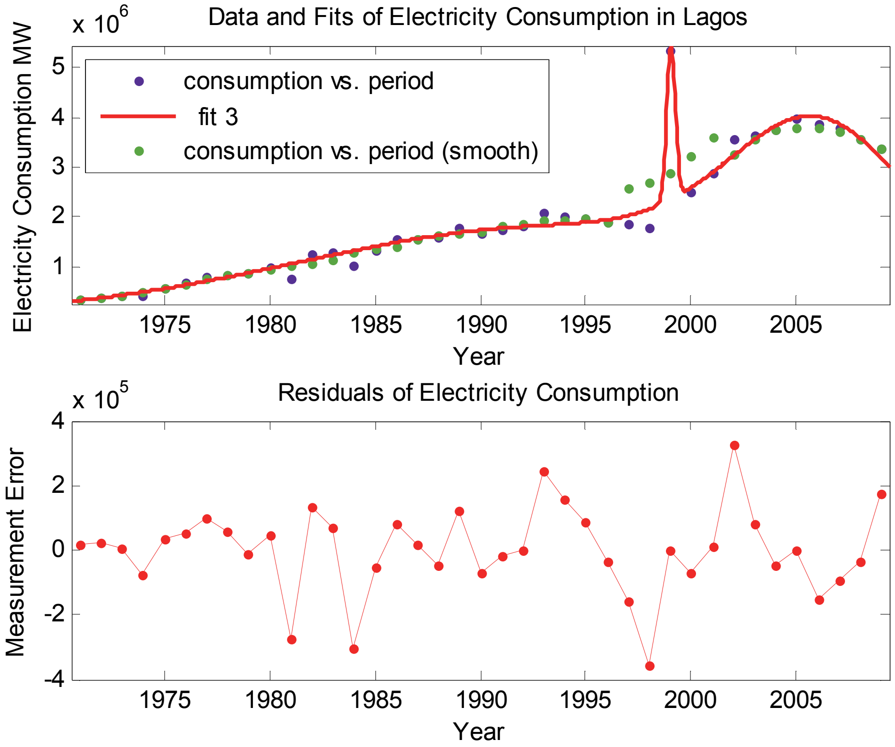

Fig. 8. A curve-fitting graph showing the non-linearity of electricity consumption data

Fit Analysis

SSE: 6.79e+011; R-square: 0.9886; Adjusted R-square: 0.9856; RMSE: 1.504e+005

Estimated Electricity Demand is necessary to determine the characteristic of expected output data. A Gaussian model described by Eq. (27) and shown in Fig. 9 was employed with 95% confidence bounds and coefficients

$$f(x) = a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2} + a_3 e^{-\left(\frac{x-b_3}{c_3}\right)^2} + a_4 e^{-\left(\frac{x-b_4}{c_4}\right)^2} + a_5 e^{-\left(\frac{x-b_5}{c_5}\right)^2} + a_6 e^{-\left(\frac{x-b_6}{c_6}\right)^2} \quad (27)$$

a_1= 2.934e+007; b_1= 1999; c_1= 0.3273; a_2= -3.444e+006; b2= 1999; c2= 2.488; a3= 1.665e+008; b3= 1993; c3= 0.2029; a4= 1.182e+007; b4= 2006; c4= 7.805; a5= 7.298e+006; b5= 1994; c5= 17.54; a6= 0; b6= 1989; c6= 0.03352

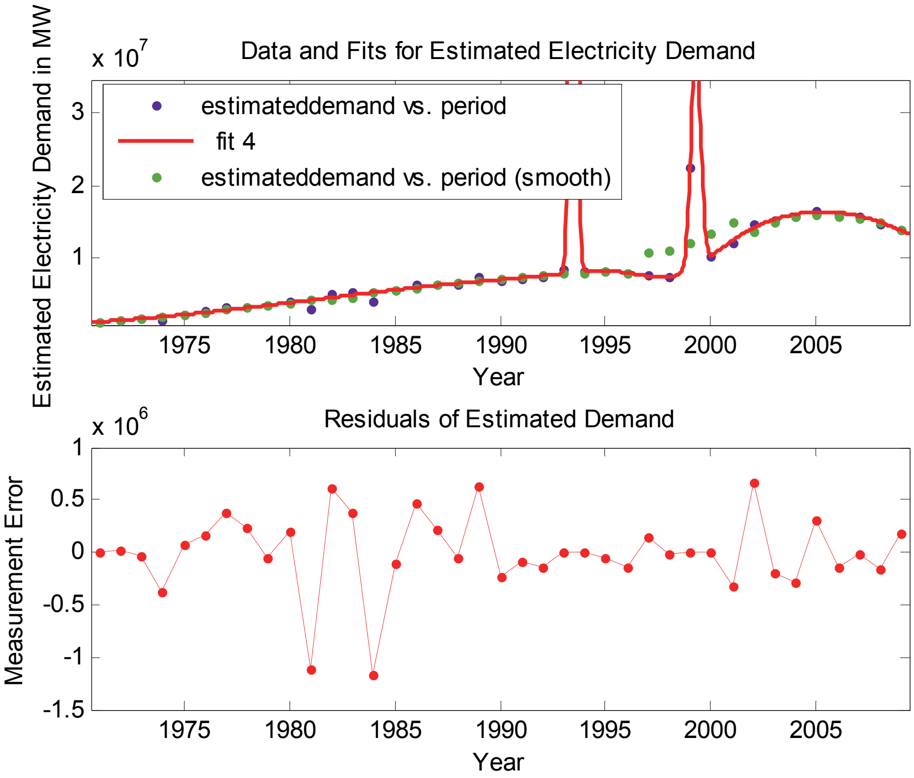

Fig. 9. A Curve-fitting graph showing the Non-linearity of estimated electricity consumption data: this is to demonstrate the non-linearity of the expected output results

Fit Analysis

SSE: 5.144e+012; R-square: 0.995; Adjusted R-square: 0.991; RMSE: 4.949e+005

4.4 Recurrent Neural Network Architecture Model for Prediction

There are five set of inputs data (*P, T,* E_c, E_v and R_v) for the ANN and an estimated output (E_d) and four of these data have been shown to have non linear property using the curve fitting functions in MATLAB. The proposed network model is a multiple- input single- output (MISO) non-linear mapping using NARMAX RNN with an output feedback scheme(a non- linear auto regressive moving average with exogenous inputs recurrent neural network model). The choice of NARMAX is because of the presence of random variables in the input data set. The RNN is required to introduce feedback (of other input and output of order higher than those presented to the network in order to avoid under modelling) in the network because the present demand for electricity will depend on the past availability, which can be taken as a measure of previous demand. This is necessary because shortfall in electrical energy supply in the country has resulted in relocation of industries from Nigeria to other neighbouring countries with stable supply of electricity. In addition, economy analysts have commented that improvement in power infrastructure will lead to industrialisation because Nigeria has the market potential. The output *y(k)* of the RNN will be a non-linear combination of non-linear functions of previous output, previous input and present input as stated in the MISO model equation of *(28)* . It is a form of global/output feedback structure in which the output is fed back into the network as represented in equations *(19-20)*.

$$y(k) = \alpha\left(\beta\left(y_{(k-1)}, y_{(k-2)}, \dots, y_{(k-N)}\right), \mu\left(x_{(k-1),}x_{(k-2)}, \dots, x_{(k-N)}\right), \gamma\left(x_0\, , x_1, x_2 \dots x_m\right)\right) + e(k) \qquad (28)$$

where $x(k)$is the input α is the zero memory non linearity function; $\beta, \mu, \ \gamma$ are non-linear activation functions and *e(k)* is the random and unobservable input variables.

The mathematical model using the *NARMAX(p,q)* equation of *(10)* is the equation *(29)*

$$\hat{y}_i(k) = \sigma(\beta \sum_{i=1}^{p} a_i y_{i-1}(k-i), \mu \sum_{i=1}^{p} b_i \, x_{i-1}(k-i), \gamma \sum_{i=1}^{p} b_i \, x_i \ + \sum_{j=1}^{q} c_j \hat{e}(k-j) + \sum_{s=1}^{r} d_s u(k-s)) \qquad (29)$$

When present output is y_i , input is x_i, random variable is c_j and error is d_s. Using the approximated NARMAX RNN Eq. (16) combined with Eq. (29) and substituting Eqs. (24-27), the proposed NARMAX RNN model is as stated in Eq. (30)

$$\begin{aligned} y_i(k) = \delta \Bigg(& \sigma(\beta \sum_{i=1}^{p} a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2} + a_3 e^{-\left(\frac{x-b_3}{c_3}\right)^2} + a_4 e^{-\left(\frac{x-b_4}{c_4}\right)^2} + a_5 e^{-\left(\frac{x-b_5}{c_5}\right)^2} \\ & + a_6 e^{-\left(\frac{x-b_6}{c_6}\right)^2}, \mu \sum_{i=1}^{p} b_i \, x_{i-1}(k \\ & - i), \gamma \sum_{i=1}^{p} b_i \, (a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2} + a_3 e^{-\left(\frac{x-b_3}{c_3}\right)^2}, a e^{bx} \qquad (30) \\ & + c e^{dx}, p \sum_{i} w_i \left(y_i - s(x_i)\right)^2 + (1+p) \int \left(\frac{d^2 s}{dx^2}\right)^2 dx \\ & + \sum_{j=1}^{q} c_j \hat{e}(k-j) + \sum_{s=1}^{r} d_s u(k-s)) \Bigg) i = 1,2, \ldots, N \end{aligned}$$

All coefficients of Eqs. (24-27) apply. $\delta, \sigma, \beta, \mu, \gamma$ are Gaussian activation functions of the form $\frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}$

4.5 Approximation of Optimal Network Size

The number of neurons *S* to approximate an ANN with 40*6 training data according to Eqs. (21-22) is

$$n = \log|240|$$

Then

$$S = 8 * \sqrt{\frac{2^{\log|240|}}{\log|240|}} \cong 12$$

And the lower and upper bound of Vapnik-Chervonenkis (VC) according to equation *(23)* is related to

$$W \log_2 W \leq VC \leq 2 * W * \log_2 (e * 8 * \sqrt{\frac{2^{\log|240|}}{\log|240|}})$$

$$W \log_2 W \leq VC \leq 10W)$$

4.6 MATLAB Implementation of Neural Network Architecture

A dynamic network with the following network object is employed in MATLAB where the key objects of a network are defined in Table 1.

The key objects of a network are:
Name: Dynamic Recurrent Neural Network for Prediction
Dimensions: shows the overall structure of a network like the input, number of layers, number of feedback layers, number of output, number of weight elements, bias.

The number of input =5 and Number of output = 1. The input is a cell array of sequential concurrent data with dummy variables and the output is a sequential data such that training is carried out on a set at an instance. Dummy variables are introduced because MATLAB requires ten time step values for each data set.

Number of feedback delays =1: an output feedback scheme according to equations *(19-20)*.

Number of weight elements is related to the number of neurons according to equation *(23)*. The proposed activation function is the Gaussian activation function because the output is a Gaussian distribution.

This is implemented in MATLAB using the NARX (Non Linear Autoregressive with Exogenous Input).

After an ANN has been constructed, it must be setup such that network input and output sizes match the available data. Also, appropriate setting must be selected for the processing inputs and outputs in order to obtain optimum network performance. The setup may be done automatically or manually.

Table 1
Elements of the Proposed RNN Model

Type of network	Multi input-single output NARMAX RNN with output feed back
Number of Layers	2
Number of Neurons	12
Number of inputs	5
Number of neurons in the output layer	1
Proposed activation function	Gaussian
Data ratio division for training, testing and validation	14:3:3
Data structure	Cell array

5. Conclusion

This paper presents an Artificial Neural Network based method for Electrical Energy Demand Forecasting using a case study of Lagos state, Nigeria.The preliminary test on available data for NN training using the curve fitting toolbox shows that the processes generating the data are non-linear and this shows that the choice of ANN for this work is justified.

References

Baum, E. B., & Haussler, D. (1989). What size net gives valid generalization?. *Advances in Neural Information Processing Systems, 1,* 81-90.

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2010). Neural Network Toolbox User's Guide. The MathWorks. *Inc., Natick, MA*.

Bhattacharyya, S. C., & Timilsina, G. R. (2009). *Energy demand models for policy formulation: a comparative study of energy demand models*. The World Bank.

Cichocki, A., & Amari, S. I. (2002). *Adaptive blind signal and image processing: learning algorithms and applications* (Vol. 1). John Wiley & Sons.

Dy, J. G. (2008). *Unsupervised Feature Selection*. In J. G. Dy, L. Huan, & M. Hiroshi (Eds.), Computational Methods of Feature Selection. US: Taylor and Francis Group.

Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures*. Algorithms and Application. NJ: Englewood Cliffs.

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (Eds.). (2008). *Feature extraction: foundations and applications* (Vol. 207). Springer.

Haykin, S. (1996). *Adaptive Filter Theory.* 3rd ed. London: Prentice Hall International.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation.* 2nd ed., New Jersey: Prentice Hall Inc.

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, (3), 31-44.

Jose, P. P., & Abraham, H. O. (1992). *Physics of Climate*. NY: Springer Verlag.

Kevin, S. (2001). *Applying Neural Networks - a Practical Guide* (3rd ed.). England: Academic Press.

Konstantinos, I. D. (2002). *Neural Networks and Principal Component Analysis*. In I. D. Konstantinos, & Y. H. Hu (Ed.), Handbook of Neural Network Signal Processing (pp. 211-248). USA: CRC Press.

Lappas, G. (2007). Estimating the Size of Neural Network from the Number of Available Training Data. In M. d. Joaquim, A. A. Luís, D. Włodzisław, & P. M. Danilo (Ed.), ANN ICANN, *17th International Conference (pp. 68-77). Portugal: Springer Berlin Heidelberg*.

Mandic, D. P., & Chambers, J. A. (2001). *Recurrent Neural Networks for Prediction*. England: John Wiley & sons.

Murata, N., Yoshizawa, S., & Amari, S. I. (1994). Network information criterion-determining the number of hidden units for an artificial neural network model. *IEEE transactions on neural networks*, *5*(6), 865-872.

Okoye, J.K. (2007). Background Study on Water and Energy Issues in Nigeria. The National Consultative Conference on Dams and Development. Nigeria.

Robert, S. (2006). *Dynamic Population Models*. Netherlands: Springer .

Rodvold, D. M. (1999). A software development process model for artificial neural networks in critical applications. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)* (Vol. 5, pp. 3317-3322). IEEE.

Tsoi, A. C., & Back, A. (1997). Discrete time recurrent neural network architectures: A unifying review. *Neurocomputing*, *15*(3-4), 183-223.

Zaman, S., & Karray, F. (2009, January). Features selection using fuzzy ESVDF for data dimensionality reduction. In *2009 International Conference on Computer Engineering and Technology* (Vol. 1, pp. 81-87). IEEE.