



UNIVERSITY OF NAIROBI
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

SHORT-TERM LOAD FORECASTING USING A HYBRID OF GENETIC
ALGORITHM (GA) AND PARTICLE SWARM OPTIMIZATION (PSO) FOR
AN OPTIMIZED NEURAL NETWORK

By

STEPHEN TABAN INYASIO SIRI

F56/74802/2014

A thesis Submitted in Partial Fulfillment for the Degree of Master of Science in Electrical and
Information Engineering, in Department of Electrical and Information Engineering of
University of Nairobi

November, 2018

DECLARATION

I **STEPHEN TABAN INYASIO**, hereby declare that this thesis is my original work and it has not been submitted for examination, for any a degree award in this or other any university.

Signature _____

Date _____

Stephen Taban Inyasio Siri

F56/74802/2014

This Thesis has been submitted for Examination with our approval as University Supervisors.

Signature _____

Date _____

Prof. Nicodemus Abungu Odera

First Supervisor

Signature _____

Date _____

Dr. Cyrus Wekesa Wabuge

Second Supervisor

DEDICATION

This work is dedicated to my late father Inyasio may His soul rest in peace, my family, and friends.

ACKNOWLEDGEMENT

God bless those who endeavor to achieve the best of their ability. I thank to God for guiding me through this tedious task in good health. This work would have not been completed without Him.

I must in a strong term recognize the support and patient portrayed by my dear wife Josephine Iromo. She had endured much during difficult times of balancing work, studies and the crisis in the country.

I would like to appreciate my beloved mother Regina Jibrato, and all the family members for their great moral support during this research project.

My deepest gratitude goes to my supervisors Prof. Nicodemus Abungu Odera and Dr. Cyrus Wekesa for their support and patient guidance in the completion of this research work.

Finally, it is worthy to acknowledge all Staff members and supporting staffs in the Department of Electrical & Information Engineering, University of Nairobi for their tireless efforts in bringing the best out of students under their patronage.

ABSTRACT

Short-term load forecasting (STLF) has emerged as one of the most important fields of study for power system operation for system efficiency and reliability. It plays a significant role in load flow analysis, contingency analysis, planning, scheduling and maintenance of power systems facilities; therefore, the system cost-effectiveness is determined by accurate load forecast. Numerous researchers have been done to improve the accuracy of the conventional methods such as time series, regression analysis or autoregressive moving average (ARMA) and the use of Artificial Neural Networks (ANNs) in load forecasting. ANN has shown more accurate results than the others. But the training of ANNs, with a back-propagation algorithm or gradient algorithms, requires long processing time has the difficulty in selecting the optimal order of the components and trapping in local minima. This research aimed at solving this problem by proposing a hybrid method based on Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for training and optimizing the weights of ANN. The proposed hybrid method enables a reduction in the search space and the iteration time. The proposed algorithm was tested in MATLAB 2016® software using 24 hourly load data of different days (i.e. weekdays and weekends) from Juba Power Plant (JPP), South Sudan. PSO, GA and a hybrid of genetic algorithm with particle swarm optimization (HGAPSO) and ANN were studied and the resulting mean absolute percentage error (MAPE) found to be range from 1.9% to 3.40%, 2.23% to 3.65% and 1.47% to 1.98% respectively. The results obtained were compared and it was observed that HGAPSO-ANN method has a better performance in reducing and improving forecast error compared to PSO-ANN and GA-ANN methods. Therefore, a hybridized HGAPSO algorithm with ANN improves forecast accuracy.

TABLE OF CONTENTS

DECLARATION	ii
DEDICATION	iii
ACKNOWLEDGEMENT.....	iv
ABSTRACT.....	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xii
CHAPTER 1: INTRODUCTION.....	1
1.1 Background.....	1
1.1.1 Load Forecasting	1
1.1.2 Load Driving Parameters.....	1
1.2 Problem Statement.....	2
1.3 Deficiencies/ Gaps	3
1.4 Objective.....	4
1.5 Scope of work.....	4
1.7 Organization of the thesis	5
CHAPTER 2: LITERATURE REVIEW	6
2.1 Short-Term Load Forecasting.....	6
2.2 Short-Term Load Forecasting Techniques	8
2.3 Statistical Methods for Short-Term Load Forecasting	8
2.4 Computational Intelligence Techniques	10
2.4.1 Artificial Neural Network.....	10
2.4.2 Fuzzy Logic	13

2.4.3 Genetic Algorithm	16
2.4.3.1 GA Population Initialization.....	17
2.4.3.2 Selection	17
2.4.3.3 The Crossover and Mutation	18
2.4.3.4 Elitism:.....	18
2.4.3.5 The GA Sequential Execution Steps	18
2.4.4 Particle Swarm Optimization.....	20
2.4.4.1 PSO Parameters	22
2.4.4.2 PSO implementation Steps	23
CHAPTER 3 : RESEARCH METHODOLOGY	26
3.1 Optimization Techniques	26
3.2 Hybridization of GA and PSO for Training ANN.....	27
3.2.1 Proposed Method.....	27
3.2.2 ANN Representation as a Chromosome.....	28
3.2.3 Population Initialization	28
3.2.4 Roulette Wheel Selection Method.....	29
3.2.5 Crossover and Mutation.....	29
3.2.7 Mutation:	30
3.2.8 Particle Swarm Optimization Algorithm (PSO).....	31
3.3 Prediction and Data Processing	33
3.4 Correlation Analysis	33
3.5 Input and output for the hybrid HGAPSO model.....	36
CHAPTER 4 : RESULTS AND DISCUSSION	43
4.1 Load Forecast Results.....	46

CHAPTER 5 : CONCLUSION AND RECOMMENDATIONS.....	58
5.1 Conclusion	58
5.2 Beneficiaries of this work	59
5.2 Recommendations.....	59
REFERENCES.....	60
APPENDINCES	66
APPENDIX A.....	66
PSO CODE	66
APPENDIX B	71
GA CODE.....	71
APPENDIX C	77
HGAPSO CODE	77

LIST OF TABLES

table 3-1: Illustrations of Various Correlation Coefficients	34
Table 3-2: Correlation Analysis Results	34
Table 3-3: PSO and GA Optimal Parameters	36
Table 3-4: HGAPSO-ANN Model Performance	38
Table 3-5: PSO-ANN Model Performance	40
Table 3-6: HGA-ANN Model Performance	41
Table 4-1: MAPE% (05/02/2010 To 11/02/2010)	43
Table 4-2: MAPE (25/07/2010 To 31/07/2010)	44
Table 4-3: MAPE% (06/11/2010 To 13/11/2010)	44
Table 4-4: Correlation R (04-10/01/2010) And Correlation R (06-13/11/2010)	45
Table 4-5: Correlation R (06-13/11/2010) And Correlation R (25-31/07/2010)	45
Table 4-6: One Week Forecast Models Comparison	46
Table 4-7: 24-Hourly Forecast Results Date 19/07/2010	47
Table 4-8: 24-Hourly Forecast Results Date 20/07/2010	48
Table 4-9: 24-Hourly Forecast Results Date 22/07/2010	49
Table 4-10: 24-Hourly Forecast Results Date 28/07/2010	50
Table 4-11: 24-Hour Forecast Hourly Results Date 30/07/2010	51
Table 4-12: Comparison of MAPE of Different Methods	57

LIST OF FIGURES

Figure 1-1: The Driving Parameters [3].....	2
Figure 2-1: Daily Load Profile For 24 Hours [3].....	7
Figure 2-2: Structure Of Three-Layered ANN [21].....	10
Figure 2-3: Artificial Neuron And Multilayered Neural Network [57].....	10
Figure 2-4: Three-Layer Forward Neural Network.	11
Figure 2-5: Flowchart For Basic GA Algorithm.....	17
Figure 2-6: PSO Searching Point Model	20
Figure 2-7 Velocity Updating In PSO	21
Figure 2-8: Flowchart For Basic PSO Algorithm	24
Figure 3-1: Flowchart For Hybrid GA &PSO	32
Figure 3-2: Total Load And Relative Humidity	35
Figure 3-3: One Year Load And Temperature.....	35
Figure 3-4: Daily Load Curves For One Week.....	36
Figure 3-5: HGAPSO-ANN Performance Nodes Based On MAPE	39
Figure 3-6: HGAPSO-ANN Performance Nodes Based On MPE	39
Figure 3-7: PSO-ANN Model Performance Nodes Based On The MAPE	40
Figure 3-8: PSO-ANN Model Performance Nodes Based On The MSE.....	41
Figure 3-9: HGA-ANN Model Performance Based On The MAPE	42
Figure 3-10: GA-ANN Model Performance Based On The MSE.....	42
Figure 4-1: 24-Actual And Forecasted Load Profile For 8th February 2010	52
Figure 4-2: 24-Actual And Forecasted Load Profile For 9th February 2010	52
Figure 4-3: 24-Actual And Forecasted Load Profile For 10th February 2010	53
Figure 4-4: 24-Actual And Forecasted Load Profile For 11th February 2010	53
Figure 4-5: 24-Actual And Forecasted Load Profile For 12th February 2010	53
Figure 4-6: 24-Actual And Forecasted Load Profile For 13th February 2010	54
Figure 4-7: 24-Actual And Forecasted Load Profile For 14th February 2010	54
Figure 4-8: 24-Actual And Forecasted Load Profile For 19th July 2010	54
Figure 4-9: 24-Actual And Forecasted Load Profile For 20th July 2010	55
Figure 4-10: 24-Actual And Forecasted Load Profile For 21th July 2010	55

Figure 4-11: 24-Actual And Forecasted Load Profile For 22th July 2010	55
Figure 4-12: 24-Actual And Forecasted Load Profile For 23th July 2010	56
Figure 4-13: 24-Actual And Forecasted Load Profile For 24th July 2010	56
Figure 4-14: Comparison Of Average Error Of Different Methods	57

LIST OF ABBREVIATIONS

AIC	Akaike Information Criterion
AIFNN	Annealing Integrated Fuzzy Neural Network
ANFIS	Adaptive Neuro-Fuzzy Inference system
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
BNN	Back Neural Network
BPNN	Back Propagation Neural Network
CAISO	California System Operator
EP	Evolutionary Programming
ERNN	Elman Recurrent Neural Network
FCM	Fuzzy c-Mean
FL	Fuzzy Logic
FNN	Fuzzy Neural Network
GA	Genetic Algorithm
GDP	Gross Domestic Product
GM	Grey Model
GNP	Gross National Product
HGAPSO	Hybrid of Genetic Algorithm and Particle Swarm Optimization
JIAWS	Juba International Airport Weather Station
JPP	Juba Power Plant
LTLF	Long-Term Load Forecasting
MA	Moving Average
MABC	Modified Ant Bee Colony
MAPE	Mean Average Percentage Error
MTLF	Mid-Term Load Forecasting
NNWT	Neural Network Wavelet Transform
NTPS-II	Neyveli Thermal Power Station Unit II
NWP	New Weather Prediction
PSO	Particle Swarm Optimization

PSO-ANFIS	Particle Swarm Optimization- Adaptive Neuro-Fuzzy Inference
PSO-GA-RBF	Particle Swarm Optimization-Genetic Algorithm- Radial Basic Function
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SA	Simulated Annealing
SOM	Self-Organizing Map
STLF	Short-Term Load Forecasting
SVM	Support Vector Machine
SVR	Support Vector Regression
T&D	Transmission and Distribution
TS	Tabu Search
TV	Television
WT	Wavelet Transform

CHAPTER 1 INTRODUCTION

1.1 Background

Load forecasting (LF) is a prediction of load demand to help an electric utility in the areas of generations, distributions, and operators in planning, economic dispatch, and management of power systems. It plays a great role in power system planning, operation, and control[1, 2].

1.1.1 Load Forecasting

Load forecasting is the basic and maybe the most importance module of power systems planning software. It helps an electric utility to make decisions in unit commitment that is which units are to be available, when and where to allocate them so as to meet demand and have acceptable reserve capacity.

This would help schedule plans for maintenance for unit to be taken offline for maintenance, while to be online, reduction of generation cost as well as improving power system reliability.

1.1.2 Load Driving Parameters

Load forecasting has numerous timeframes depending on driving factors affecting load, this include Short-Term Load Forecasting (STLF), Mid-Term Load Forecasting (MTLF), and Long-Term Load Forecasting (LTLF) [2]. These driving factors are

- Times, such as
 - Day or night hours
 - Weekday or weekend
 - Season of the year
- Weather condition, such as temperature and relative humidity
- Types of consumers such as residential areas, commercial centers, industry, agricultural farm, public, etc.
- Occasions, public holidays, TV programs, etc.

- Increase in Economy such as income per capita, Gross National Product (GNP), Gross Domestic Product (GDP), etc.
- Tendencies in upcoming technology
- Price of electricity

As cited above, load predicting approaches are classified into Short-Term, Medium-Term and Long-Term. The STLF is hourly prediction, whereas MTLF and LTLF are for daily and seasonal predictions respectively [2]. Show in Figure1-1,

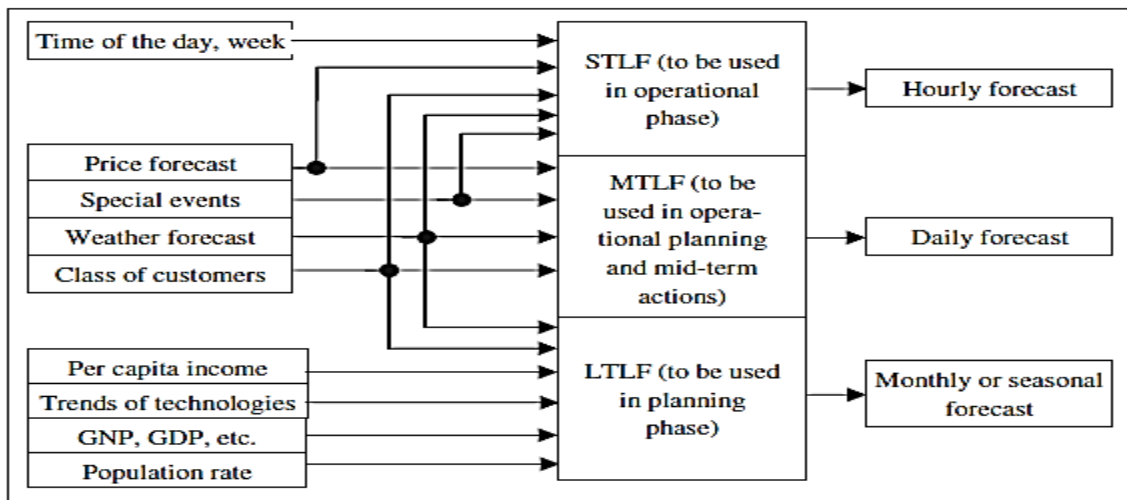


Figure 1-1: The driving parameters [3]

1.2 Problem Statement

Electric load prediction is a critical step for electric power companies in the process of planning. The purpose of load forecasting is to meet future demand, reduce unforeseen cost and provide a possible input to the decision such as systems reliability, efficiency, distribution, transmission (T&D) and the cost [3-6]. In order to plan for an effective power systems operation and control, the utility company must be able to anticipate the consumers future demand, how to deliver it, where and when [3, 7]. Since the number of customers connected to a network increases continually, demand increases, necessitating upgrade of facilities to greater capacity [7]. This requires earlier planning, hence the role of load forecasting

To come out with appropriate approach for future demand prediction, there are a lot of challenges electric companies in deregulated markets face such as weather prediction difficulty,

inability to store electricity, mass customer switching, different meter types, new or lost contracts, data problems, models, etc. On the other hand, electricity moves at the speed of light, making it faster to drop off at customer's end in a millisecond and possible to deliver in no time. Therefore, because of all this, the customers' demand load needs to forecast every hour; on a continuous basis and any forecast error in Electricity Company resulted to very high cost.

1.3 Deficiencies/ Gaps

In the effort to improve load forecasting, many researchers have utilized different methods such as statistical methods, computational methods, and artificial intelligence optimizations methods to enhance the forecast accuracy, minimize error, schedule maintenance as well as reduction of the generation cost. However, these methods have presented significant results with some limitations. Therefore, there is a need to explore new methods to come up with better solutions. Some of the deficiencies reviewed in the literature review which need to be addressed to improve the load forecast accuracy are:

Statistical Methods

1. Their performance deteriorates when a sudden change in variables that affect load patterns happens.
2. They only consider static load data
3. Have a large number of complex equations, accompanied by long computation time, and may result in numerical uncertainties.

Artificial Intelligence

1. Many researchers were proposed on Artificial Intelligence-based techniques for STLF, however, the limitation in computer memory and the long processing time cause a barrier to artificial intelligence efficient implementation.
2. Difficulty in selecting the optimum order of the components and trapping in local minima.

1.4 Objective

The Overall objective is to develop a hybrid model based on Genetic Algorithms with Particle Swarm Optimization (GAPSO) for training Feed Forward Neural Network (FNN) to forecast next 24-hourly loads.

The sub-objectives are as follows:

1. To design a hybrid model of Genetic Algorithms with Particle Swarm Optimization (GA-PSO) for optimizing Feedforward Neural Network (FNN).
2. To apply actual historical load data obtained from Juba Power Plant (JPP) to predict next 24-hour load profile using the developed models.
3. Validate the results of proposed model by comparing against GA and improved PSO algorithm.

1.5 Scope of work

Load forecasting is an important tool to the electric utility for power systems planning such as unit commitment, reduction of the reserve power, scheduling maintenance as well as reduction of the generation cost. It is a multi-tasks practice. Artificial neural network techniques are superior to statistical techniques for STLF. However, design of best possible network structures has not yet been successful. Artificial Intelligence requires optimal design of network structure and suitable training algorithm, in order to enhance the precision of the forecast as well as network performance.

1.6 Research Questions

1. How will the objectives of the research be achieved?
2. How will the new approach of the hybrid models GA-PSO be formulated to train ANN?
3. What limitations will be considered and how will they be addressed?
4. What are the advantages and the limitation of the proposed algorithms, how to improve them to achieve a better optimization model to reduce the prediction errors and better computational times?

5. Will the hybrid model produced have any challenges or weaknesses?
6. What other constraints may be added to improve the new approach?
7. What parameters will be varied in the program and how will they affect the output of the program?
8. Will the new approach be applied to real systems?
9. How significant is the research to Load prediction?

1.7 Organization of the thesis

This thesis is organized in Five Chapters, as follow: Chapter One describes research background, problem statement, deficiencies/gaps, objectives, scope of the work, and research questions. Chapter Two reviews of related studies and this include load forecasting, load forecasting methods such as statistical and computational methods. Chapter Three explains the methodology and the proposed method respectively. Chapter Four elaborate results and discussion. Finally, Chapter Five is conclusion and the recommendations.

CHAPTER 2 : LITERATURE REVIEW

2.1 Short-Term Load Forecasting

Short-Load forecasting (STFL) is a forecast of demand from a minute, an hour, a day up to a maximum of a week [1, 2]. It is required for the economic dispatch of generation to meet the system load. Since electricity has great impact on the economic development in any country, the load forecasting accuracy is of great significance for operation and load management of a utility company to produce the energy needed to meet the demand of the country as well as support its development projects [8]. Due to unknown or random factors affecting the daily energy demand, in particular, special occasion days such as holidays, days on which strikes occur or extreme weather conditions which are difficult to model in a mathematical way, Short-term load forecasting (STLF) still has challenges despite a lot of studies have been done on load predicting methods.

To develop an appropriate forecast model which incorporates all factors affecting load such is not an easy task [1,3]. Therefore, in order to develop an accurate forecasting tool, it is essential to understand the characteristics of a power system load, the factors that affect the shape of a load profile and all the parameters involved in load demand. Figure 2-1 shows the trend in the supply of electrical demand over a time period [9].

A load shape can be influenced by the following factors:

Time: The power system load behaves differently at various times in a day over a 24-hour period. The load at midnight is different from the load at peak hours of the same day as well as in the evening as shown in Figure 2-1.

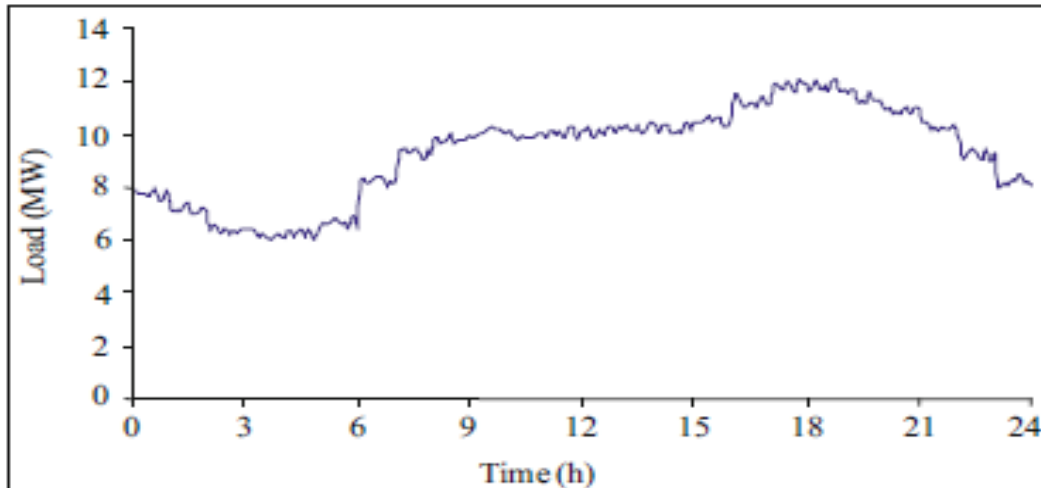


Figure 2-1: Daily load profile for 24 hours [3]

Day type: Weekday and weekend has a relatively high influence on the energy consumption as mostly a weekend load is lower than a weekday load. Therefore, the load of weekday is relatively different from weekend.

Weather Factors: Temperature, moisture, rain, wind speed, cloud cover etc., have influences the load profile. The change in weather such as a hot day or cold day will result in the use of electric machines either switching on the air conditioner on a heater.

Seasonality: The Seasons of the year have an influence on the energy consumption of a customer.

Economic: The GDP and GNP of a country influence the utilization of electricity, if the economic factors i.e. GDP, GNP shows a promising future, and the new electricity-based appliances are coming up in the market, the electricity consumption may increase as well in all classes of the consumers. In short-term load forecast the economic inductors such as GDP and GNP are ineffective; however, they may have strong effective on MTLF and LTLF.

Electricity Pricing: The price of electricity has a major role in electric energy consumption by the customer. The increase in electricity prices may resulted in a reduction of forecast load, as many customers may control their electricity consumption[2]. Vice versa, if the electricity prices are predicted to be low, customers demand will increase.

Special Occasions: (TV programs, public holidays, etc.)

Holidays: Demand on public holidays is not the same as on “working days”. They have different forecast models. All the factors mentioned above can be incorporated together to design load forecaster model that based on historical data, however, irregularities do occur which could cause deterioration in the accuracy of the forecast. These are a few factors that are difficult to incorporate into short-term forecasting model, such GNP, GDP, etc. they are ineffective to be incorporate in STLF forecaster.

2.2 Short-Term Load Forecasting Techniques

A large variety of Statistical and artificial intelligence techniques have been developed for Short-Term load forecasting [10]. The statistical load forecasting methods exploit time series[11].These methods used static load data with regular distribution features to correlate the relationships of the electricity consumption and other factors. The statistical methods have disadvantages because it used the historical data for future load predation and incapability to adapt with dynamic load series. Since the load is dynamic, a difference between current load and past load data present will result to large forecasting errors [12].

The application of computational intelligence techniques has been widely studied in electric load forecasting. They are essentially circuit dynamically adaptable and have demonstrated capacity to do non-linear curve fitting and forecasting error is less as compared with statistical forecasting techniques. Computational intelligence techniques can be combined as Hybrid methods to enforce a load forecasting techniques [8,9 ,10]

2.3 Statistical Methods for Short-Term Load Forecasting

M. Ghayekloo, M.B.Manhaj, *et al* (2015) [13] proposed a hybrid of Short-Term Load Forecasting (STFL) frame-work with a new inputs selection method. BNN (Bayesian Neural Network) is used to forecast the load. The combination of the correlation analysis and 12 norms selects the appropriate inputs to the individual Bayesian Neural Network. The authors ‘compared the proposed STLF with the existing state of the-art forecasting techniques, shows a significant improvement in the forecast accuracy

A. Jain and B Satish, *et al* (2009) [14] applied fuzzy to provides load prediction data for load economic dispatch, generation scheduling at all time. A. Jain compared forecasted load with the conventional methods forecasted values. The estimated load matches the actual load within $\pm 3\%$.

A. Kumluca and I. Erkmén, *et al* (2003) [15] Used hybrid learning approach of recurrent neural network which have one or more feedback loops to forecast one day in advance. The authors expressed that the model can be applied to real-time application that has historical load data, the recent observed data can be supported by online learning phase of weights update within certain periods. The author described that, the proposed model shows considerably better results.

H. Al-Hamadi, S. Soliman, *et al* (2004)[16] applied Kalman filtering algorithm with moving window weather to predict the peak load for each day. The authors described that the approach yields better results compared to the (ARs), (MAs), (ARMAs), and ARIMA) which are non-weather-sensitive models.

J. Nowicka zagrajek, R. Weron *et al* (2002)[17] applied ARMA with hyperbolic noise to forecast real data of the California System Operator (CAISO) using new seasonality removal technique, the authors tested the effectiveness of the method by comparing the real load data with the predicted values. The approach produces a 1.2–1.25% APE difference, whereas the CAISO returned error is 1.7%.

H. NieG.Liu, X. Liu, *et al* (2012)[18] applied the combination ARIMA and SVM to predict load, where ARIMA model was applied to forecast static part of the load and the SVM was employed to handle nonlinear load, the authors' stated as a hybrid model ARIMA –SVMs, produced better results. The Error analysis for model ARIMA, SVMs, and ARIMA-SVMs MAPE are 4.50%, 4.00%, and 3.85% respectively and the RMSE (MW) 43.4, 38.77 and 35.72 respectively.

M. Felice, A. Alessandri, P. Ruti, *et al* (2013)[19] applied a numerical weather prediction model, using univariate time-series methods ARIMA/X, the study was done in Italy 2003 in the summer months of June and July. The authors stated that, the incorporation of NWP model to ARIMA leads to reasonable results, especially for hot regions where the consumption of electricity is more seriously influenced by temperature.

2.4 Computational Intelligence Techniques

2.4.1 Artificial Neural Network

Artificial Neural Network (ANN) is an Artificial Intelligence derived from biological natural systems of human brain[52], McCulloch and Pitts were the first to introduce Artificial Neural Network in (1943) [53]. The ANN processing components called input, hidden and output layers or nodes interconnected with synapses weights [54]. The ANN synapses control the inputs and nonlinear characteristic of the transfer function in hidden neurons [24]. The neural networks consist of input and a hidden layer as well as the output layer as shown in Figure 2-2.

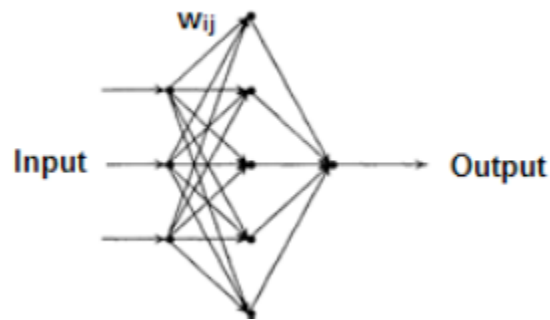


Figure 2-2: Structure of Three-Layered ANN [21]

The Figure 3-2 depicted, Three-layer Forward Neural Network to interchange data received and handle it accurately and effectively[25].

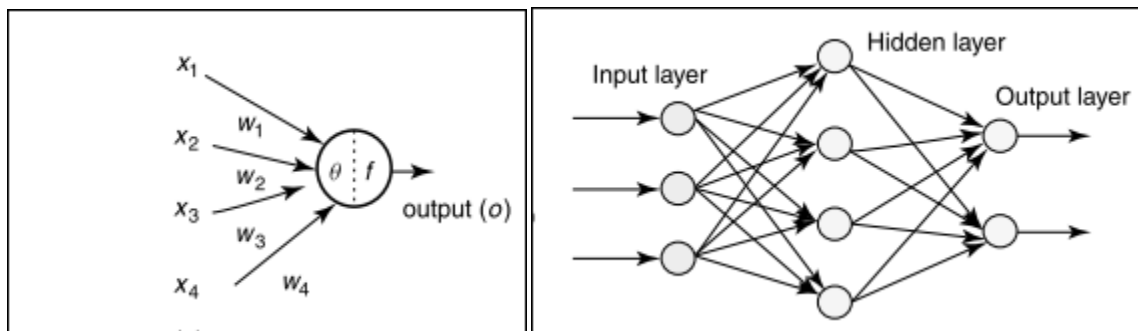


Figure 2-3: Artificial Neuron and Multilayered Neural Network [25]

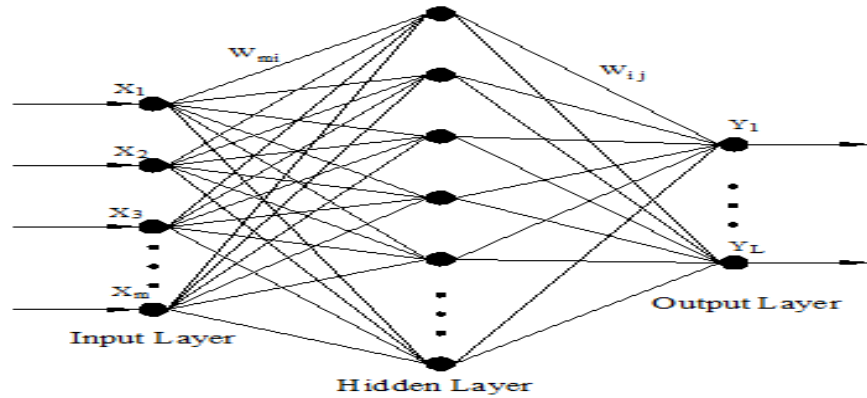


Figure 2-4: Three-layer Forward Neural Network.

From the above Figure 2-4, the n set of problems to network is expressed as linear combination n inputs. Therefore, for the n -inputs and output, signals X_1, X_2, \dots, X_n flows via neurons at hidden layers and to the output signal flow Y_o . The output equation is formulated as

$$Y_0 = f\left(\sum_{i=1}^n \omega_i x_i\right) \quad i = 1, 2, \dots, n \quad (2.1)$$

Where, $w = \{\omega_1, \omega_2, \dots, \omega_n\} \in R^n$ is called the weights vector? The weight $(\omega_i)^n_{i=1}$ assign to each input synapse. It may be positive or negative. f is transfer function and, n is the input number, ω_{ij} the weighted link to the inputs i th to j th hidden layers, and θ_j is the threshold hidden layer. Assuming that the transfer function at the hidden layer is tan sigmoid, the linear output can be formulated as follow:

$$f(H_j) = \left(\frac{1}{1 + \exp^{-(H_j)}}\right), \quad j = 1, 2, \dots, m \quad (2.2)$$

$$H_i = \sum_{i=1}^n \omega_{ij} * x_i - \theta_j \quad (2.3)$$

The output signal Y_o is given by the equation (2.4)

$$y_o = \sum_{j=1}^m (w_{kj} * f(H_j)) + \theta, k = 1, 2, \dots, 0 \quad (2.4)$$

Where, ω_{kj} is a weight link vector between the j th hidden layers and k th outputs layers. For n inputs, and n outputs of network, such as $\chi = \{x_1, x_2, \dots, x_n\}$ input vectors and the vector weights $W = \{\omega_1, \omega_2, \omega_3, \dots, \omega_n\}$, and Y_o is the output. Suppose the sets of training inputs i.e.

$$\left. \begin{array}{l} \chi^1 = (\chi_1^1, \chi_2^1, \dots, \chi_n^1) \rightarrow y^1 \\ \chi^2 = (\chi_1^2, \chi_2^2, \dots, \chi_n^2) \rightarrow y^2 \\ \vdots \\ \chi^k = (\chi_1^k, \chi_2^k, \dots, \chi_n^k) \rightarrow y^k \end{array} \right\} \quad (2.5)$$

For given input χ , the resultant learning error and mean percentage error can be describe to measure the effectiveness and response of the network system as follows

$$MSE_{Error} = \frac{1}{n} \sum_{k=1}^n \frac{E_k}{y_i} \quad (2.6)$$

The objective function is expressed with the mean absolute percentage error (MAPE) as

$$Objfunc. = \frac{1}{(1 + MAPE)} \quad (2.7)$$

$$MAPE = \frac{1}{n} \left(\sum_{i=1}^n \frac{|y_i - d_i|}{y_i} \right) * 100 \quad (2.8)$$

Where, $E_k = (y_1 - d_i)^2$, n is the number of training inputs,

S. Quaiyum, Y. Khan, S. Rahman, *et al.*(2011) [26] conducted the study on different methods of short-term hourly load prediction using different types of ANNs, such as RNN, PSO, ERNN, and PSO-ERNN. According to their discussion and the comparison, The Elman Weather sensitivity model gives a good result; however, it has longer processing time compared to ENN and PSO with recurrent neural networks which are faster but slightly more prone to errors.

I. Ibraheem, D.Ph, M. Ali *et al* (2014) [27] described ANNs as one of main tool applicable in machine learning. As the name neural suggested, the neural network is brain inspired systems which are intended to replicate the way that human learn. The learning capability of ANN is attained by regulating the weights according to the chosen learning algorithm [28].

I. Drezga, *et al* (1998)[29] introduced hybrid novel method of extremely learning machine (ELN) and modified artificial bee colony (MABC) for short-term load forecasting. The MABC was developed as global search techniques to find the weights of the inputs and the biases of the hidden neuron of ELM. Authors tested the ELM-MABC on two datasets of ISO New England data and North America electricity utility data, the ELM-MABC produced good results relatively to the standard and state of art methods.

Li, Song Wang, Peng Goel, Lalit *et al* (2015)[30] developed hybrid model based on Wavelet Transform Extreme Learning Machine (ELM) and partial least squares regression. The individual forecasts models were derived from combinations of WT with different levels of decomposition, for each sub-section from WT decomposition consist of 24 parallel ELM invoked to forecast hourly load of the next-day. The numerical result shows the proposed method significantly improved forecasting performance.

2.4.2 Fuzzy Logic

Fuzzy Logic concept started in the mid-1960 initiated by Dr. L.A. Zadeh. It is based on computational set of rules and thinking. It helps in solving logical problems and gives decisions[31].

Assembly of the Fuzzy Logic Inference comprises of three theoretical parts, i.e.

- Rule part, where ‘if conditions’ are set to carry out logical statements
- Databank or database defines the relationship of the functions.
- Intellectual tool: it does the inference process depend on the rules set for it and derives a rational output and conclusions.

I. Harrison, *et al* (2014)[32] conducted a study on an hour ahead load for the company located in the north eastern of Nigeria based on adaptive neuro-fuzzy inference system (ANFIS). From the

results obtained by ANFIS model, the author concluded that the ANFIS is a better model for one hour ahead load forecast.

M. López, S. Valero, C. Senabre, *et al* (2012)[33] presented a Kohonen's Self Organizing Maps (SOM) Neural Network model for a STLF, the model applied to the analysis the predicted load and real load data. The authors described that the model is flexible in the inputs, frequency and the prediction period specifically, in specific group of customers in small area.

B. Wang, N. Tai, H. Zhai *et al* (2008)[34] presented a new Auto-Regressive Integrated Moving Average model with an Evolutionary Algorithm (EA), and Particle Swarm Optimization (PSO). The (EA) and (PSO) handle nonlinear part of the load of power system and categorize the parameters of ARIMAX model for Prediction. The authors described that a hybrid method has advantage in speeding up the convergence of PSO, which enhance the global search ability, so more capable to achieving good accuracy than PSO.

S. Pandian, S. Duraiswamy, K Asir, C Christopher. *et al.* (2006)[35] applied Fuzzy Inference and the fuzzy rules for STLF, the method was simulated in MATLAB and using data obtained from Neyveli Thermal Power Station in India. According to the authors, the results show the error $\pm 3\%$.

A.Khwaja, M.Naeem, A. Anpolaga, *et al.* (2015)[36] Presented a Bagged Neural Network (BNN) for STLF. The authors compared the performance of BNNs technique with the ANN artificial neural network, Bagged Regression Trees, and Auto Regressive Moving Average supervised and unsupervised ANN. The analysis result obtained by BNNs has lower error compared to some ANN and Bagged Regression Trees.

A. Abdoos, M. Hemmati, *et al.* (2014)[37] applied knowledge-Based Systems of a hybrid intelligent based for STLF. The authors found that the hybrid method is more stable in convergence and very efficient, flexibility in modeling and its computational time is good but can be improved

A. Ghanbari, N. Kandil. M. Saad *et al.* (2010)[38] conducted a study to compare the Artificial Intelligence techniques for short-term load prediction, the AIs such as (ANN), (ANFIS) and Genetic Algorithm (GA). The results of approaches were compared using Means Absolute

Percentage Error (MAPE). The authors described that the ANFIS output result is approximate to real load than ANN and GA. ANFIS are an appropriate tool for STLF problems.

M. Othman, I. Musin, Q. Raza *et al.* (2012)[39] tested the application of ANNs to STLF. The author's observation of the pure ANN models that were constructed and tested was that there was room for improved forecasting which lead to the introduction of an expert system. The resulting forecasting errors ranged from 0.5% to 2.5%.

A. Badri, Z. Ameli, Z. A. Birjandi *et al.* (2012)[40] proposed fuzzy logic system with a network structure and learning steps similar to neural network thus giving it the name Fuzzy Neural Network (FNN). The authors discovered that the FNN could forecast the future load with accuracy similar to that of neural network. The error ranged from 2.43% to 3.06% for the FNN while their ANN error ranged between 2.3% and 3.14%

S. Pappas, L. Ekonomou, P. Karampelas *et al.* (2010)[41] compared multi-model partitioning theory with three-time series analysis techniques such as (AIC) and (BIC) to forecast load and price, the data from Hellenic power systems were used, the method demonstrated good result and its use in the studies electricity consumption and prices forecasts.

M. Buhari, S. Adamu, *et al.* (2010)[42] applied ANN to forecast the daily load of the 132/33kV sub- Station, Kano, Nigeria, the Levenberg-Marquardt optimization technique as (BPA) was applied to learned ANN. The authors reported that the method is recommended for future load demand prediction.

G. Liao, T. Tsao, *et al.* (2004) [43] discussed on integration of the fuzzy neural network, with the simulated annealing and evolutionary programming (EP) to forecast load, FHRCNNs was applied for searching optimal parameter of EP. The authors described evolutionary programming is capable of getting global best value, however, inability to get the local best point search and the simulated annealing helps in local optimal search.

P. Li, Y. Li, Q. Xiong *et al.* (2014) [44] introduced genetic algorithm to obtain suboptimal structure of hybrid Quantized Elam Neural Network (HQENN) to forecast hourly load, using the inputs data such as historical data, predicted temperature and time index. The authors applied quantum law to describe the interface of qubit neurons and the classic neurons.

2.4.3 Genetic Algorithm

Genetic Algorithm (GA) is heuristic search computation based techniques inspired from natural biological selection by Holland, 1975 [44], which based on mechanism of the biological organism that has been adopted and prospered in the highly competitive changing environment, according to theory of survival of the fittest by Darwinian [45], or simulates the process of natural evolution [46]. Genetic algorithm (GA) optimization is applicable in any control processes for parameters optimization, via the mutation and crossover operators to select the optimized values[47]. The proper selection of the crossover and mutation values depend on the problem requirements and the encoding methods[38].

The genetic algorithm looks for solutions in large spaces using the operators by crossing the parents, mutation, and selection, whereby the probability of global increasing as well as the convergence [34]. The results of the crossovers are the offspring. In mutation, GA randomly changes some of the genes values of the parents [7, 38]. In general, this work presents genetic algorithm as optimization technique which has ability to search a solution in a vast region to come up with optimal results.

GA search starts with the set of solutions representing the chromosomes called population; the solutions are taken from one generation to form new generation with the motivation that the possibility of the new generation will be better than previous generation. Further solutions were selected based on their fitness $f(x)$ to formed a new solution called offspring.

Where, $\mathcal{X} = \mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n$, represent the vectors of the optimal parameters. The building block of the Genetic Algorithm is formed of chromosomes where genes are concatenation in the form of binary strings $Chrom = [S_1^1 S_2^1 \dots, S_{\ell_1}^1, S_1^2 S_2^2 \dots S_{\ell_2}^2, \dots, S_1^n S_2^n \dots S_{\ell_n}^n] = [\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n]$,

Where S_i^i is gene string, ℓ_i the code length of gene string, i^{th} optimization parameters & $\mathcal{X}_k = [S_1^k S_2^k \dots, S_{\ell_k}^k]$?

Figure 2-5 shows the flowchart representation for a basic GA algorithm.

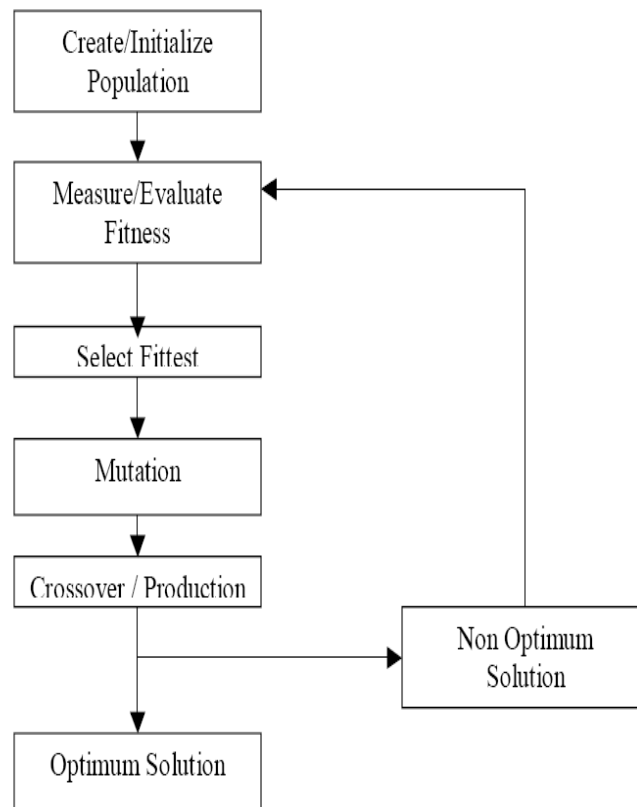


Figure 2-5: Flowchart for basic GA Algorithm

2.4.3.1 GA Population Initialization

The n -chromosomes is initially created randomly as vectors n -gene, each vector represent a point search solution in the search space of the problem.

2.4.3.2 Selection

The selection of best genes according to their objective functions and generates temporary population, is done by selection operator, such as tournament, random and roulette wheel selection. In the roulette wheel, each solution candidates are sort according to fitness of the function which reflects the function of the previous solutions candidates.

2.4.3.3 The Crossover and Mutation

These operators are a basic component of genetic algorithm; crossover may be of single, multi and uniform point crossover.

2.4.3.4 Elitism:

Elitism is the process of keeping the best genes or chromosome from generation to generation and replaced worst gene, or chromosome from the old generation with the newly created from the old generation. Therefore, the process of conserving the elite parent is called elitism.

The fundamental components for genetic algorithm implementation that should be considered are the population, parameters and the GA operators. The following procedures for implementing genetic algorithm are specified below:

2.4.3.5 The GA Sequential Execution Steps

The sequential implementation of GA, the n chromosomes represents a set of m possible solution. Where m vector number of GA parameters, each m -parameter is a search point to the dimension of problem

Step by step of GA implementation

1. Initialization: Create the initial population from gene strings randomly.
2. Fitness Function: Each individual (gene) in the population is evaluated using the fitness $f(\chi)$.
3. The selection: The selection of best two individual parents from the population to mate, depending on individual fitness, and check the stopping criteria.
4. Crossover and mutation. The probability of the crossover, the two-parent crossover to produced new offspring's called children. If the crossover is not happened, the offspring's is copy of the parents, while in the mutation mutate a new offspring at each point.
5. Accepting. Place new offspring in the new population to next generation.

6. New population: Discard the non-elite population members and replace with the new children
 7. Repeating steps 4 and 5 number of times to generate number of candidates at mating pool
 8. Elitism: keep the best for next generation and discard worst.
 9. Check the convergence criterion at certain iteration; if the end condition is satisfied stop and return the best solution in current population, keep the solution as final result. Or else return to step 2
-

Z. Honghui, L. Yongqiang, *et al.*(2012) [49] Applied GA to enhanced adaptive fuzzy inference system by incorporating Artificial Neural Networks with Fuzzy Inference System for STLF. The authors used fuzzy inference system to identify the structure of ANFIS, while backpropagation and genetic algorithm was applied for learn neural network respectively. The authors describe that ANFIS produced better result, precision and less training time compared to Artificial Neural Network.

B. Islam, Z. Baharudin, Q. Raza *et al.* (2013)[50] integrated Genetic Algorithm with Artificial Neural Networks for STLF, the Genetic Algorithm was used for initial weight selection and structure optimization of FFNN. The authors expressed that the forecast precision of the approach is enhanced.

F. Yu, X. Xu, *et al.* (2014) [51] Proposed a combination of Back-Propagation Neural Network with Genetic Algorithm for STLF. GA was used to train and decide the initial weight of BP neural network, such as not to stuck in local minima.

S. Yu, K. Wang, Y. Wei, *et al.* (2015)[54] applied hybrid of artificial intelligent algorithm such as genetic algorithm and particle swarm to improve the Radial Basis Function, the model was used to forecast yearly energy consumption in Wuhan. Authors describe the model as unsupervised prediction model. The combined PSO-GA gives an advantage in searchability both at global and local search.

2.4.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is evolution algorithms developed by Eberhart and Kennedy which based on optimization theory of swarm intelligence algorithm [55]. The particles in the swarm compete amongst themselves to created intelligence of particle [55]. In the Swarm, each particle fly search solution in space with velocity and position [56]. At a time of convergence, every particle converges to the best position attained by previous particles, and the global best positionof swarm. The particles movement and locations experienced is calculated iteratively [45,46]. The process involves the updating of a particle velocity and position with time until the best solution is obtained. The velocity of a particle is update depend to three factors, such as the velocity, the best position moved or experienced and the best position of the whole swarm has experienced as depict in the Figure 2-6 and 2-7.

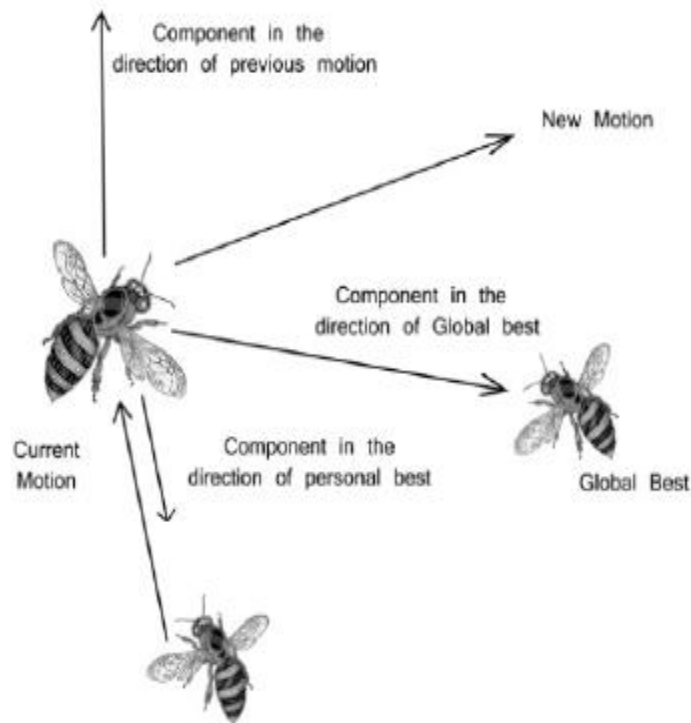


Figure2-6: PSO searching point model

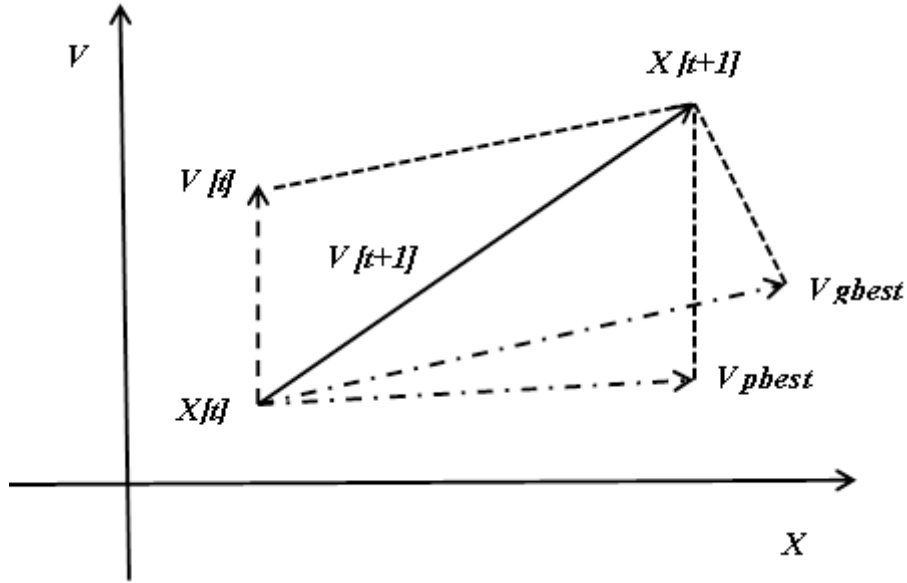


Figure2-7 Velocity updating in PSO

In the process of "particle" finding the optimal solution, its velocity $V_i(t)$, direction, and position $\chi_i(t)$ coordinate is changing, and the formula applied is shown as follows

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 [pbest_i(t) - \chi_i(t)] + c_2 r_2 [gbest(t) - \chi_i(t)] \quad (2.9)$$

$$\chi_i(t+1) = \chi_i(t) + V_i(t+1) \quad (2.10)$$

Where $V(t+1)$ is updated velocity,

ω is the inertia weight for learning equilibrium between local and global search.

C_1 and C_2 are weight coefficients for each term respectively

V_i Particle is current velocity

r_1 and r_2 are the values within the range (0,1).

$pbest_i(t)$ is the particles best position and

$gbest(t)$ is the groups best position.

2.4.4.1 PSO Parameters

In the equation (2.9) and (2.10), the position of a particle at time (t) is $\chi_i(t)$ and $V_i(t)$ the velocity. $p_{best}(t)$, a particle best position attained, and $g_{best}(t)$ is the global best position experienced by a particle in a swarm. C_1 and C_2 are initial values of learning coefficients factors influencing the $p_{best}(t)$, and $g_{best}(t)$ position of the particle, r_1 and r_2 are the random values within the range (0,1). ω , is the inertia weight which provides learning equilibrium between local and global search, formulated as follows:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{It_{\max}} * It \quad (2.11)$$

Where, ω_{\max} and ω_{\min} are the maximum and minimum inertia weight respectively, It iteration at time $t=0$, and It_{\max} the maximum of allowable iteration. The equation (2.12) indicated the particle best position of the particle i , which is the best position that the particle has visited and equation (2.13) is a single best solution found called the global best particle, in the entire particle in the swarm.

Therefore, $i \in (1, \dots, s)$ if,

$$f[pbest_i(t)] \leq f[\chi_i(t+1)] \Rightarrow pbest_i(t+1) = pbest_i(t) \quad (2.12)$$

$$f[pbest_i(t)] > f[\chi_i(t+1)] \Rightarrow pbest_i(t+1) = \chi_i(t+1)$$

$$gbest(t+1) = \min \{f(g), f[gbest(t+1)]\} \quad (2.13)$$

Where, $g \in \{pbest_0(t), pbest_1(t), \dots, pbest_2(t)\}$

2.4.4.2 PSO implementation Steps

In the PSO algorithms, the population particles in the swarm represent sets m possible solution.

Particle Swarm Optimization Algorithm sequential steps,

1. Initialization

Initialize the population of particles with random position and the velocity, inertia weight, acceleration constants in the swarm

2. Fitness

For n number of the particles, evaluate the objective function and the fitness of particles using the Equation (3.5).

3. Compare the fitness value of each particle (3.11) using objective function and save as *best* for each particle, choose the *best* as the value obtained.

4. Velocity and location

The position update and the velocity of the particles is obtained using the equations (3.7) and (3.8)

5. Updating the *gbest* and *pbest*

Evaluating the fitness values of the particles and updating *gbest* and *pbest* values using equations (3.11) and (3.12)

6. Check the stop condition.

One the stop conditions are met proceed to step 7, otherwise return to step 2.

7. End simulation.

Figure 2-8 depicted the flow chart of a basic Particle Swarm Optimization algorithm.

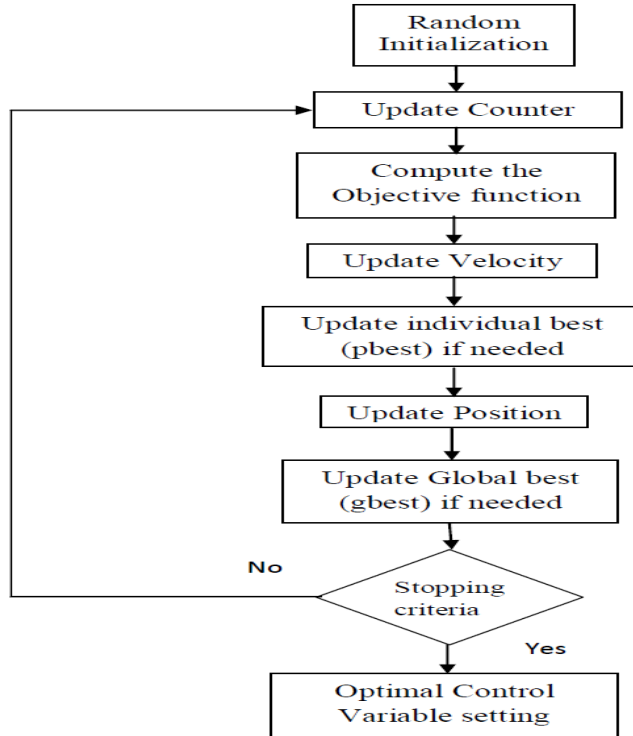


Figure 2-8: Flowchart for basic PSO Algorithm

P. Duan, K. Xie, T. Guo, *et al.* (2011)[56] applied a combined fuzzy logic FCM, artificial intelligence PSO with time series SVR techniques in short-term load forecast, The authors firstly used Support Vector Regression to forecast the load separately, while the same data were applied to PSO-SVR, the results found for PSO-SVR and FCM are 1.443% and 1.066% respectively.

Z. Bashir, *et al.* (2009) [58] trained Adaptive Artificial Neural Network (ANN) with Particle Swarm Optimization (PSO) and BP to adjust weights ANN. The author described that method has a higher load forecasting accuracy compared to BP.

H. Pousinho, V. Mendes *et al.* (2011)[59] presented a hybrid artificial intelligence (AI) of PSO with ANFIS, for daily wind energy prediction. Authors outline the proposed method is innovative and effective, with MAPE of %.41% and less computational time compared to ARIMA, NN, NNWT, WNF, and HPA.

Despite of various methods which have been studied to improving the precision of the load forecasting methods, the design the optimal ANN topology, as well as the number of neuron, long processing time and stuck at local minima, [23,26] still challengeable. The factors that affect

the demand are either unknown or random; in particular the forecasting of a load for special occasion days such as holidays, days on which strikes occur or extreme weather conditions,[60]. Hence the current research is interested at improving the error accuracy in STLF using hybrid of GA-PSO model.

CHAPTER 3 : RESEARCH METHODOLOGY

3.1 Optimization Techniques

The current trend in the technologies with advances in computing, the system complexity became more complex especially in electric power systems where the generations, distribution and customers system growing rapidly. To handle these very large scale problems it requires different optimization techniques to be incorporated [28], these optimization techniques are Artificial Intelligence and Evolutionary Algorithms. They are often hybridized with several heuristic approaches to facilitate and solve the complex optimization task [28]. The studies have been carried out on applications of these Artificial Intelligence and Evolutionary Algorithms techniques. However, there are different optimizations techniques which can be used in enhancing the forecaster precision so as to improve the accuracy. Hence, a lot of optimization techniques need to be explored.

In the past decade witnesses the exciting advance in applications of artificial network for solving many optimization problems in power systems such as load forecasting [19, 20], fault diagnosis [61], construction cost estimations [62], there are many algorithms used for learned artificial neural network, such as Back Propagation [33][38][43] Genetic Algorithm [39], Simulating Annealing [21], Particle Swarm Optimization [54][58].

So far, the most functional learning algorithm for Artificial Neural Network is the gradient-based backpropagation (BP). Although back propagation has been used for learned ANN to solve many problems, it has some limitations, firstly BP algorithms easily get stuck at local minima, secondly, slow in convergence as compare to conventional computation system, which may be good at accurate and exact computation, however, weak in operations. Evolutionary computation provides a more robust and efficient approach for solving complex problems [63].

To improve performance of BPNN, the introduction of hybridization of Artificial Intelligence algorithm is necessary, such as AI algorithm, Particle Swarm Optimization which works on social interchange and the behavior and Genetic Algorithm explore search space. In this research studied, we proposed new algorithm that combined both GA and PSO for learned ANN, the performances of proposed method was compared to GA and PSO

Although, researchers revealed that evolutionary algorithms, such as GA and PSO have been proposed for learned the ANN. Meanwhile, the GA and PSO are the heuristic and Stochastic search algorithms, they have less tendency to get stuck in local minima, and both are population based of group of individuals particles with a specified similar to biological phenomenon. These similarity phenomena let to increasingly development of EA computation very interesting field. Furthermore, a possible disadvantage of PSO is premature convergence, because it requires an additional momentum. By incorporate of the genetic operators in the PSO may provide the balance between the global search of GA and social thinking ability in PSO will improve the ability of hybrid algorithm.

The Genetic Algorithm search end solution negatively since they are discarded progressively during iterations. In other way, the strength of Particle Swarm Optimization and its flexibility to absorb other parameters are presented. Therefore, genetic algorithm was adopted be used in initial stages for exploration and then a Particle Swarm Optimization improved by incorporating the GA's operators was selected to be used later for exploitation determination. The hybridization of two optimization techniques strength was believed to over excellent results.

3.2 Hybridization of GA and PSO for training ANN

There are several ways to incorporate the advantage of GA and PSO to come out with a better technique for learned ANN. Recently, hybridization of GA and PSO is become popular dur to their abilities to handle several-real world problem which are more complex, uncertain and imprecision. Figure 3-7 depicted the flow chart of proposed algorithm.

3.2.1 Proposed Method

The concept behind these hybrid algorithms is to combine the search abilities of the algorithms to optimizing the weights and the bias of ANN. The hybridization of GA and PSO optimization technique strength provides exploration for global search and the exploitation of local search in different undiscovered regions.

The proposed hybrid method works as follows. First, initialization of population n -pop of n candidate solutions is generated randomly within the interval $[x_{Max}, x_{Min}]$, for each iteration (It) of algorithm. Using GA selection and recombination operations were applied in n -pop to

produce new solutions in current population. The current population $n\text{-pop}_{t-1}$ was enhanced and evaluated according to the objective function and best solutions are recorded.

After the new generation is created, particle swarm optimizations enhance individuals' particles by interchanging the social information among them and the learned knowledge. The improved ones are reproduced and selected for crossover. Offspring's reproduced from improved ones are expected to have better performance than the old population, hence the weak performance individuals will be discarded from generation to generations. This hybrid iterative search process continues until specified stopping criteria was satisfied.

3.2.2 ANN Representation as a Chromosome

The total number of the weight of ANN depends on input nodes, hidden nodes and output nodes, which are formulated below,

$$\text{T.N.Ws} = [(In_n + 1) * H_n + (H_n + 1) * O_n] \quad (3.1)$$

Whereas In_n are the input nodes, H_n , the number hidden nodes which correspond to the weight bias and O_n is number of output nodes.

The initial weights were randomly initialized within the interval [xMax, xMin], and each weight are weighted link between the neurons of the layers to another. In the research, we have represented the individuals such that each individuals chromosome contain a number of gene representing the weights of ANN

The fundamental components for genetic algorithm implementation that should be considered were the population, parameters and the GA operators. The following procedures for implementing GA are specified below:

3.2.3 Population Initialization

The n -chromosomes GA was initialized randomly as vectors n -genes within the interval [xMax, xMin], these-vectors represent a possible solutions of the problem.

3.2.4 Roulette Wheel Selection Method

In the roulette wheel approach, a probability of selection p_i assigns to each individual q . Each individual is selected according to their fitness which reflects the fitness of the previous individual chromosome. So, a series of N -random numbers is generated and compared against the cumulative probability $cp_i = \sum_{q=1}^n p_i$ of the population. Roulette wheel selection method was

develop by Holland *et al*[64], state that if the fitness of individual i in the population is $f_i(x)$, its chance of being selected to next generation is

$$P_i = \frac{f_i(\chi)}{\sum_{q=1}^n f_q(\chi)} \quad (3.2)$$

Whereas, the n is the total number of the individual in a population and $f(\chi)$, is objective function of the individual i . Thus, each individual has a chance to become a parent in next generation according to its fitness. In other selection methods, the individuals with better fitness have highest chances of selection which is biased. It may neglect the best individuals of a population, therefore, there is no assurance that, the best one will pass to succeeding generation.

The roulette wheel selection method, the parent is selected according to the fitness. Each individual is designated to a slice of roulette wheel, the slices sizes are proportion to the individual's fitness, therefore, the bigger the value the larger the slice size. However, individual with best or worst fitness has chances to be selected for next generation. This is a merit, however, the solution may be having weak results, but it could be useful for following regeneration process.

In this research work, the algorithm employed GA to do exploration while roulette wheel selection techniques were used for selection, thus making the process the most complimentary.

3.2.5 Crossover and Mutation

Early convergence is the most critical problem in EA optimization techniques involving populations, which happens when the parent highly fit in a population reproduce many offsprings

of the same similarities in early stage of evolution[64]. However, crossover operation of the parents could not generate different offspring, because they acquired the same information which is applied to crossover the chromosomes. Whereas an alternate operator, called mutation can help in exploration new areas compared to the crossover. Therefore, the crossover is used for exploitation while the mutation is applied for exploration new one.

In this research crossover and mutation operators has been used to both Genetic Algorithm and Particle Swarm Optimization of the hybridized algorithms. The operators help to avoiding early convergence, partial optimism and thus enhance the performances of the algorithms. Boden hofer *et al* 2004 [65] assumed that handling n -dimensional optimal problems, i.e. $\chi = \mathbb{R}^n$. Then, n -dimensional vector of individual is denoted as real numbers.

$$a = (\chi_1, \chi_2, \dots, \chi_n)$$

3.2.6 Flat Crossover:

Given two parents $a_1 = (\chi_1^1, \dots, \chi_n^1)$ and $a_2 = (\chi_1^2, \dots, \chi_n^2)$

To reproduce the new offsprings $a^\varphi = (\chi_1^\varphi, \dots, \chi_n^\varphi)$ for all $i=1, 2, \dots, n$. Arithmetic crossover applied to calculate mean of the parents.

$$X_{\text{childA}} = \beta * x_{\text{childA}} + [1 - \beta] * x_{\text{childB}} \quad (3.3)$$

$$X_{\text{childB}} = \beta * x_{\text{childB}} + [1 - \beta] * x_{\text{childA}} \quad (3.4)$$

Where β_i a uniformly distributed random value from the unit interval is used to compute the offspring,

3.2.7 Mutation:

If the element X is selected for mutation, the resulting offspring is given by;

$$x^k = x^k + \Delta x^k \quad (3.5)$$

$$\Delta x^k = +(xMax - x^k)(1 - rand^{(1 - It/MaxIt)^\psi}) \quad (3.6)$$

The xMax is a maximum limit of search space, xMin is the minimum limit of search space, It is initial iteration at time $t=0$, MaxIt is a maximum limit of the iteration, ψ is a parameter determining the degree of iterations.

In proposed GA algorithm flat crossover and mutation was employed to avoid the process of encoding and decoding of the chromosomes, thus, speed-up the algorithm and facilitated in less computational time.

Siriwardere *et al.* (2006) [66] presented the variation and probability selection of crossover and mutation for urban drainage model optimization. The authors describe that 80% probability work best for crossover and 1% is the best figures to work for mutation. Therefore, 80% crossover and 1% mutation probabilities have been used in this research work.

3.2.8 Particle Swarm Optimization Algorithm (PSO)

The PSO developed to enhance individuals reproduced from GA by both sharing information between each other and their individually learned knowledge. Then, these enhanced individuals are reproduced and selected as parents for crossover operation. The process involves adjustment of velocity and position of particles involved towards the best particles position, the global best position experienced by particle, hence the updating the velocity and position involves the equation (3.7) and (3.8) as illustrated below.

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 [pbest_i(t) - \chi_i(t)] + c_2 r_2 [gbest(t) - \chi_i(t)] \quad (3.7)$$

$$\chi_i(t+1) = \chi_i(t) + V_i(t+1) \quad (3.8)$$

This research work proposes the use of a hybrid of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for optimizing Artificial Neural Network (ANN). The ANN is considered to forecast next day load profile with the aim of reducing and improving forecast error.

The Figure 3-7 show flow chart of a hybrid Genetic Algorithm (GA) and Particle Swarm Optimization algorithm (PSO)

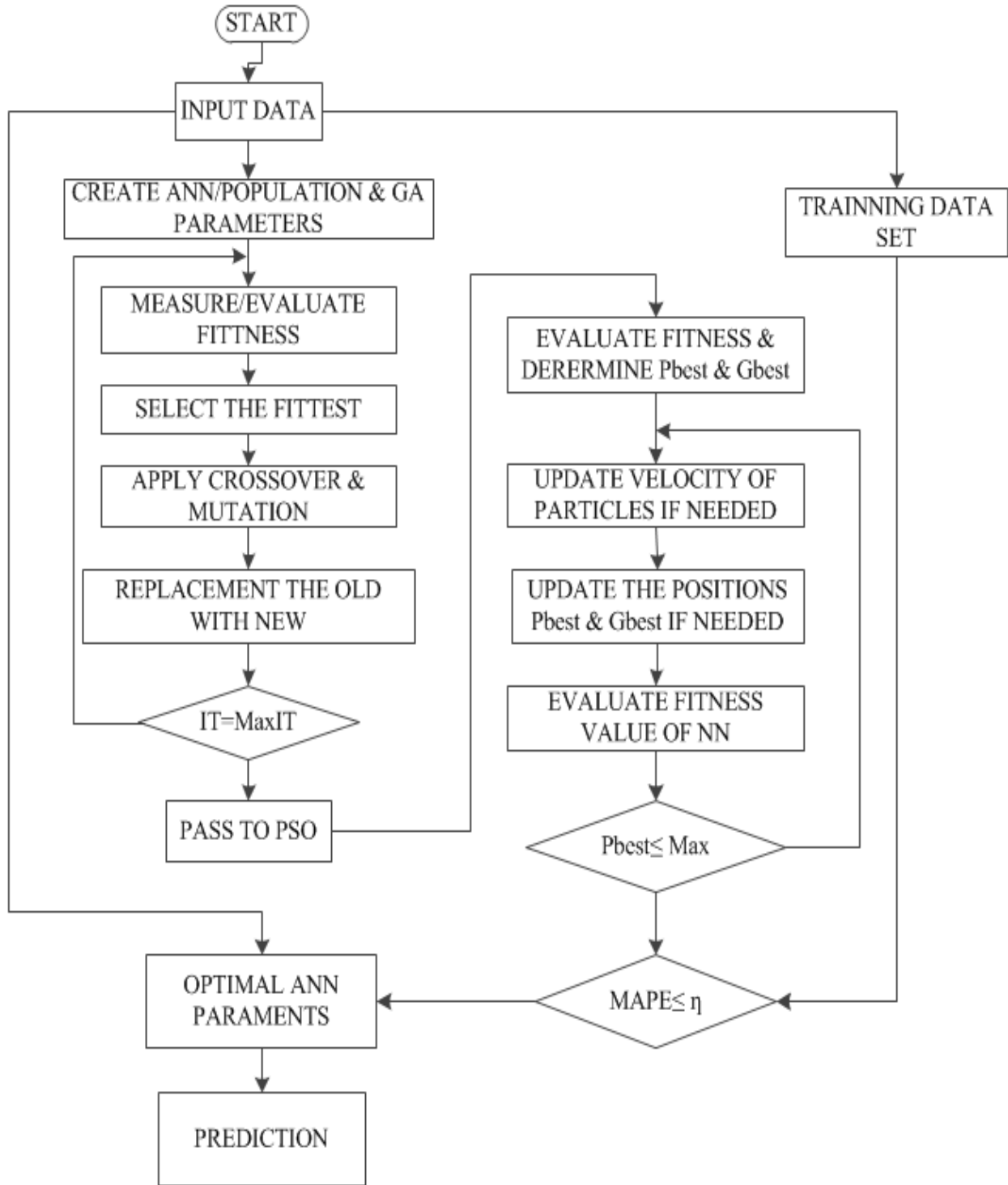


Figure 3-1: Flowchart for Hybrid GAPS0

3.3 Prediction and Data Processing

The ANNs input data fitting requires a proper selection to minimize the variation of sampled input and output data to improve the accuracy, first, the original data set needs to be normalized. In this research, the linear transformation technique is used to normalize the data, shown as follows:

$$\chi_n = \frac{\chi_i - \min \chi}{\max \chi - \min \chi}, i = (1, 2, \dots, n) \quad (3.9)$$

Where $\max \chi$ is a maximum and $\min \chi$ is minimum value, χ_n is the matrix data vector converted within [0, 1] after the normalization process.

In this project, the data sets, historical load data and weather data for analysis were obtained from Juba Power Station (JPS) and Juba International Air Weather Station (JIAWS) for the year 2010 are used to build the GA-PSO with ANN to predict daily electric demand. The data from January to April was used for validation of the proposed method. The data was grouped into weekday load data and weekend load data because the weekday and weekend have different load activity.

3.4 Correlation Analysis

DeCoursey *at el.* 2003[67] conducted studies on correlation analysis on two dependent variables of X and Y to determine the relationships. Peter X-K, Song *atel.* 2007[68], both the authors expressed that if the variables are dependent, they are linearly correlated. The measure of correlation coefficient relationship of two variables can be calculated as follows;

$$S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 \quad (3.10)$$

$$S_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (3.11)$$

$$S_{xy} = \sum_{i=1}^n \left\{ (x_i - \bar{x})(y_i - \bar{y}) \right\} \quad (3.12)$$

$$r_{xy} = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}} \quad (3.13)$$

Where S_{xy} is co-variance of x and y. S_{xx} and S_{yy} is the mean deviation of x and y, r_{xy} is correlation coefficient.

Table 3-1: Illustrations of various correlation coefficients

Correlation	(X, Y)
$r_{xy} = +1$	Positive correlation
$r_{xy} = -1$	Negative correlation
$r_{xy} = 1$	Perfect correlation
$r_{xy} = 0$	No correlation between the variables' and y

The samples of load data and weather data were used to get information about the correlation between load data and the weather variables. The regression analysis presumes that the independent variable has no error and the dependent variable has a random error.

Correlation analysis was done to correlate the weather variables and the load, and determine which weather variables have significant effect on loading substation. The weather variables which were taken into account are daily observed Temperature and daily observed Relative Humidity; other variables are not included because of data inadequacy. The Table 3.2 shows the correlation results.

Table 3-2: Correlation analysis results

Variable	r_{xy} = correlation coefficient
Relative Humidity, Load	0.3265
Temperature, Load	0.6917

From the Table 3.2 of correlation analysis above, shows the temperature has significant effect on the load contrast to relative humidity, hence, temperature will be include in load forecasting.

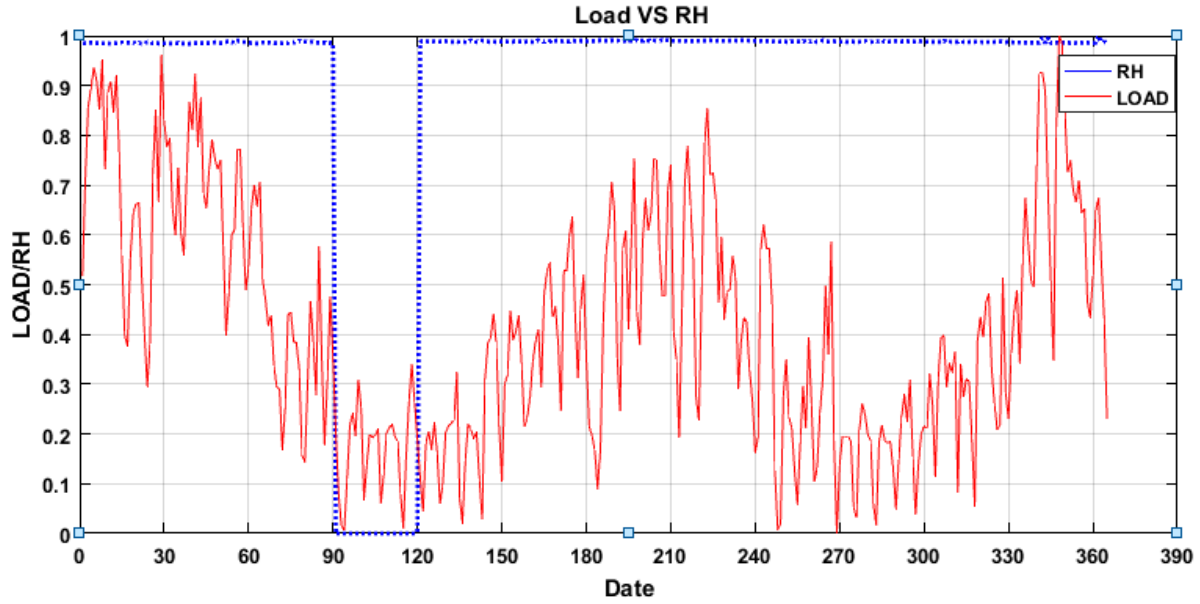


Figure 3-2: Total Load and Relative Humidity

From the Figure 3-2 depicted, load varies, while the relative humidity seems constant.

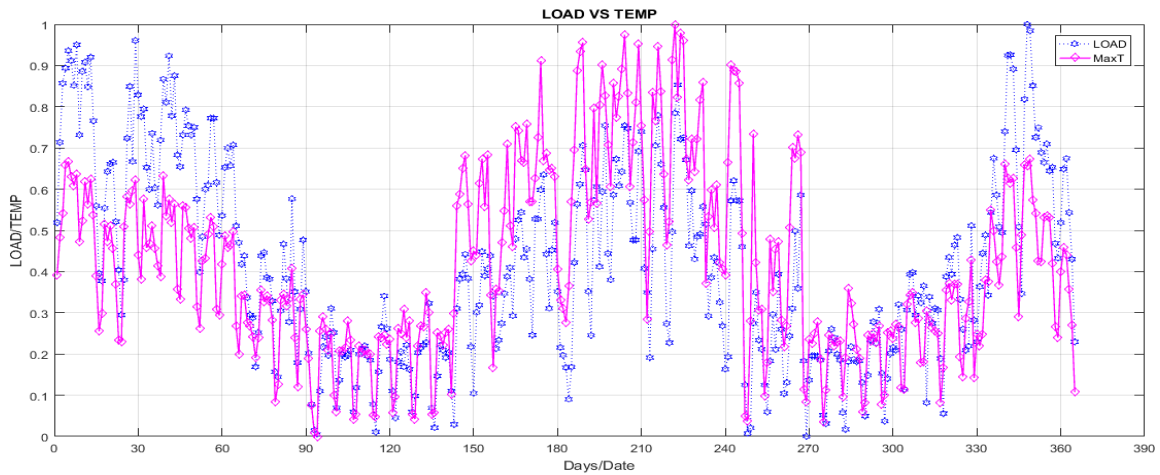


Figure 3-3: one year load and Temperature

From the Figure 3-3 depicted, the total load varies relatively with increase in temperature, thus indication that temperature play significant role on load profile, therefore, temperature has impact on load forecasting.

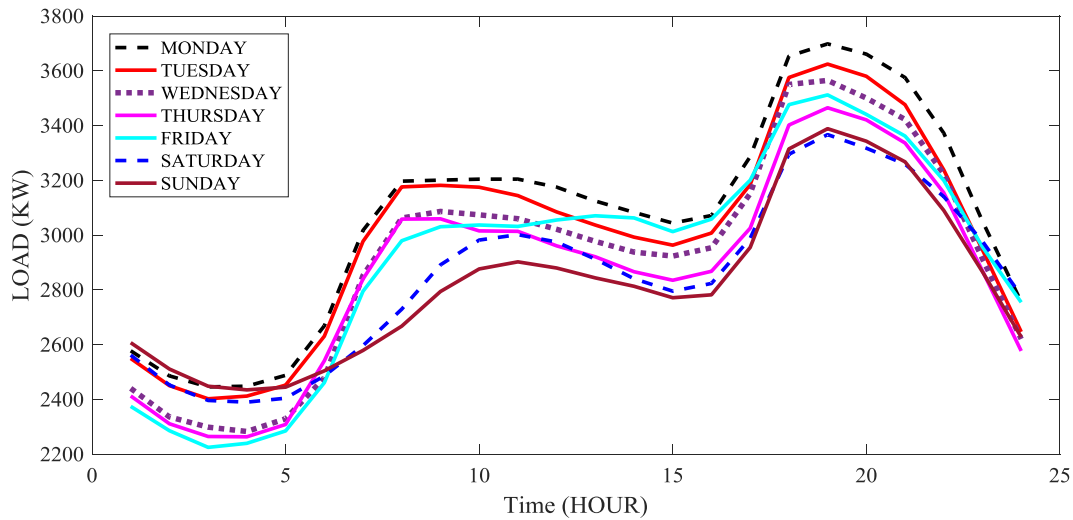


Figure 3-4: Daily load curves for one week.

Table 3-3: PSO and GA optimal parameters

Parameter	PSO	GA	Description
Number of particles n	50	50	Population Size (Swarm Size)
Max Iteration	1000	1000	Maximum Iteration
C1	1.5	-	Cognitive coefficient
C2	2.5	-	Social coefficient
Inertial weight ω	$0.1 + \text{rand} * 0.4$	-	Determine the influence of the current velocity
Maximum weight	0.9	-	
Minimum weight	0.4	-	
Vmax	+5	-	Maximum velocity
Vmin	-5	-	Minimum velocity
xMax	1	-	Upper bound of swarm
xMin	0	-	Lower bound of swarm
Beta	-	8	Selection Pressure
Pc	-	0.8	Percentage Crossover
Pm	-	0.01	Percentage Mutation
Mu	-	0.1	Mutation Rate

3.5 Input and output for the hybrid HGAPSO model

The historical load data and weather data for analysis were obtained of Juba Power Station (JPS) and Juba International Airport Weather Station (JIAWS) for the year 2010. The data were clustered as weekdays from Monday through Friday, Saturday and Sunday as weekend due to the

different load profile of weekday and weekend, the weekdays load curves have relatively similar shape for different weeks. Figure 3-4 show daily load curves for one week from, Monday to Sunday. The inputs to hybrid model used 24 hours load of day, the previous day, 168-hour load for the week and the previous week. One layer for output representing time ahead 24 hours load forecast for next day.

The idea behind clustering the data and taking specific inputs was to consider the number of hours of the day, the effect of the temperature variation on the load, weekday and weekend and how they reflect on load pattern.

3.6 Simulation Results

The simulation process for getting the forecast results, which is the network learning process, can be summarized below:

- a) The load historical data was read and loaded from excel data sheet ('Datafile.xlsx') and it's then normalized to the range of 0-1.
- b) The data is loaded to Matlab workspace from excel file 'Datafile.xlsx'.
- c) The 'Datafile.xlsx' file is read using 'xlsread' function load inputs and target.
- d) The network is then created 'fit net', with number of inputs and output. Since there are no rules set for determining the structures of the artificial neural network, therefore, in this proposed model, the numbers of the neurons at the hidden nodes were determined by trial error. The appropriate numbers of neurons for the proposed hybrid model are 30 neurons used in the hidden layers which produced better MAPE and APE, as shown in Figure 3-5.
- e) The data from May-July 2010 was applied for learning the neural network and from the July 2010 data were used for neural network validation.
- f) The output results of the ANN were renormalized and a compared with the input data of neural network.
- g) Network performance was evaluated by using (MAPE) and APE.

In this research, the designs of ANN, GA, PSO, and HGAPSO were simulated and the sequence results are shown in tables, Table 3-4, Table 3-5 and Table 3-6. The values in the Table 3-3 were adopted by Mishra *et al* 2008[69] and validated by experimentation with other values.

The ANN with 24 inputs, 5 nodes at hidden layer and output layer i.e. N=30 are all connected, the hidden nodes were increased to obtain the minimum neurons at hidden nodes that would produce the lowest MSE and MAPE. The network design configuration was taken based on the results

The network topology is taken from the best performing approach as it is illustrated in Table 3-4, Table 3-5 and Table 3-6 and performances were compared.

Table 3-4: HGAPSO-ANN Model Performance

Day	Network	inputs	hidden node	MSE	MAPE	R
Weekday	HGAPSO-ANN	24	5	0.042	0.259	0.992
			10	0.032	0.222	0.834
			15	0.021	0.175	0.942
			20	0.024	0.205	0.992
			25	0.015	0.059	0.982
			30	0.029	0.256	0.997
			35	0.025	0.209	0.992
			40	0.049	0.338	0.937
			45	0.038	0.329	0.937
			50	0.022	0.173	0.985
			55	0.013	0.104	0.995
			60	0.026	0.186	0.961
			65	0.028	0.248	0.978
			70	0.021	0.183	0.988
			75	0.014	0.114	0.984
			80	0.038	0.292	0.916
			85	0.001	0.033	0.953
90	0.009	0.076	0.959			
95	0.033	0.252	0.997			
100	0.009	0.066	0.890			

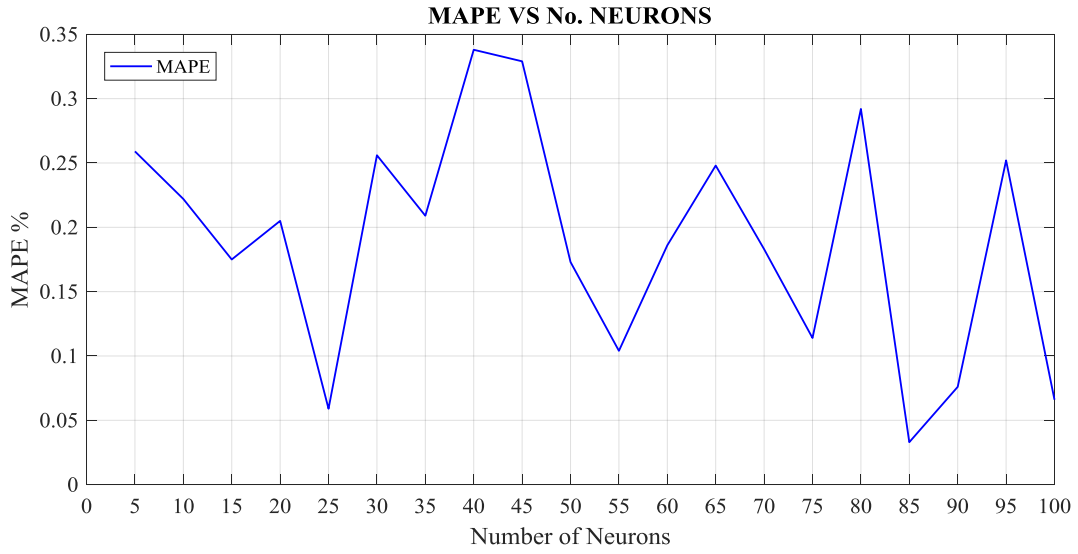


Figure 3-5: HGAPSO-ANN Performance Nodes Based on MAPE

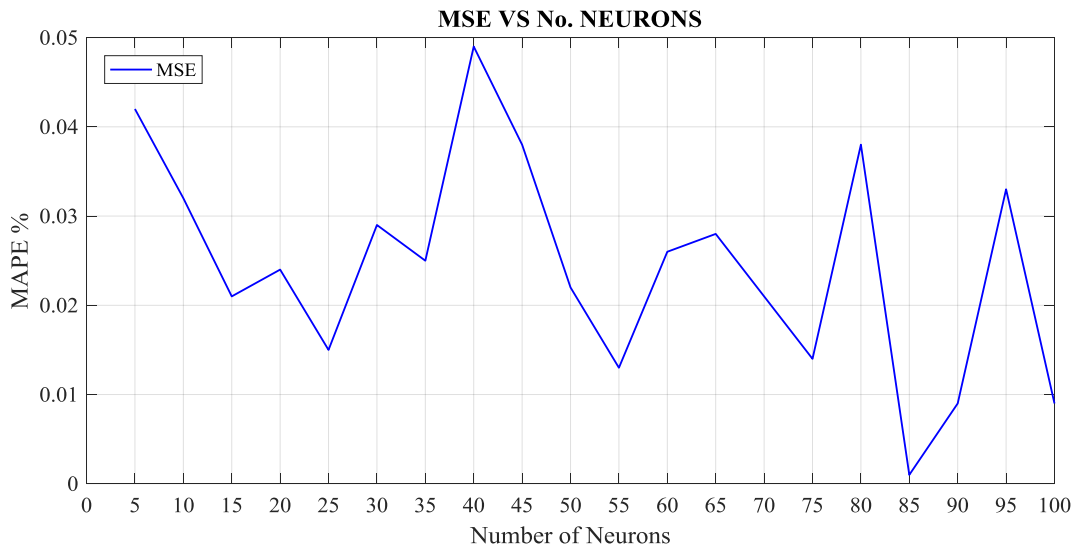


Figure 3-6: HGAPSO-ANN Performance Nodes Based on MPE

Table 3-5: PSO-ANN Model Performance

Day	Network	inputs	hidden neuron	MSE	MAPE	R
Weekday	PSO-ANN	24	5	0.0169	0.197	0.963
			10	0.0181	0.175	0.997
			15	0.0207	0.175	0.996
			20	0.0202	0.169	0.998
			25	0.0140	0.114	0.997
			30	0.0182	0.147	0.999
			35	0.0209	0.174	0.996
			40	0.0163	0.123	0.995
			45	0.0182	0.123	0.999
			50	0.0165	0.108	0.995
			55	0.0122	0.072	0.999
			60	0.0206	0.159	0.998
			65	0.0229	0.201	0.999
			70	0.0204	0.178	0.979
			75	0.0141	0.114	0.998
			80	0.0150	0.049	0.995
			85	0.0260	0.223	0.999
			90	0.0161	0.101	0.991
			95	0.0304	0.253	0.998
			100	0.0177	0.154	0.998

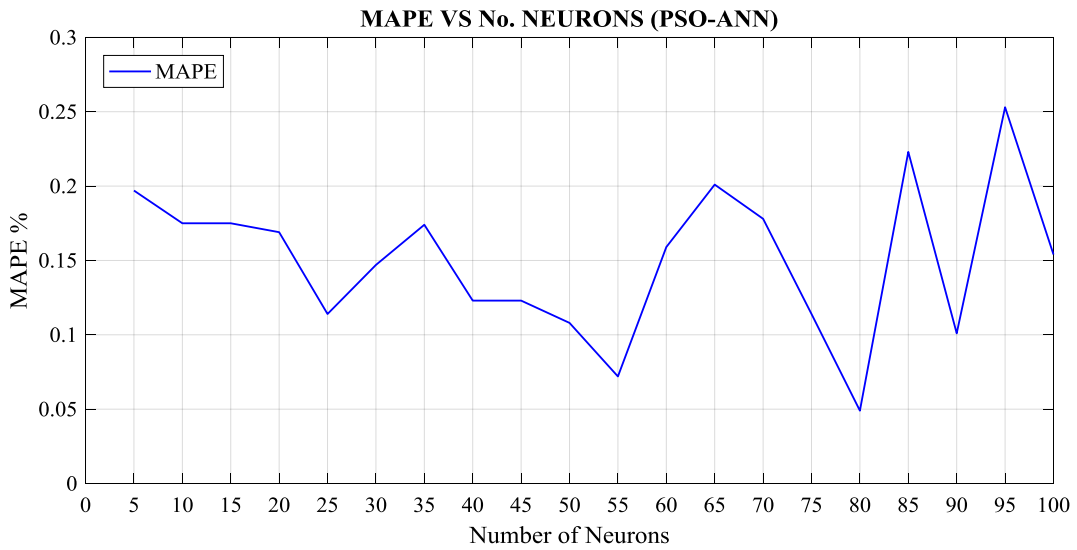


Figure 3-7: PSO-ANN Model Performance Nodes Based on the MAPE

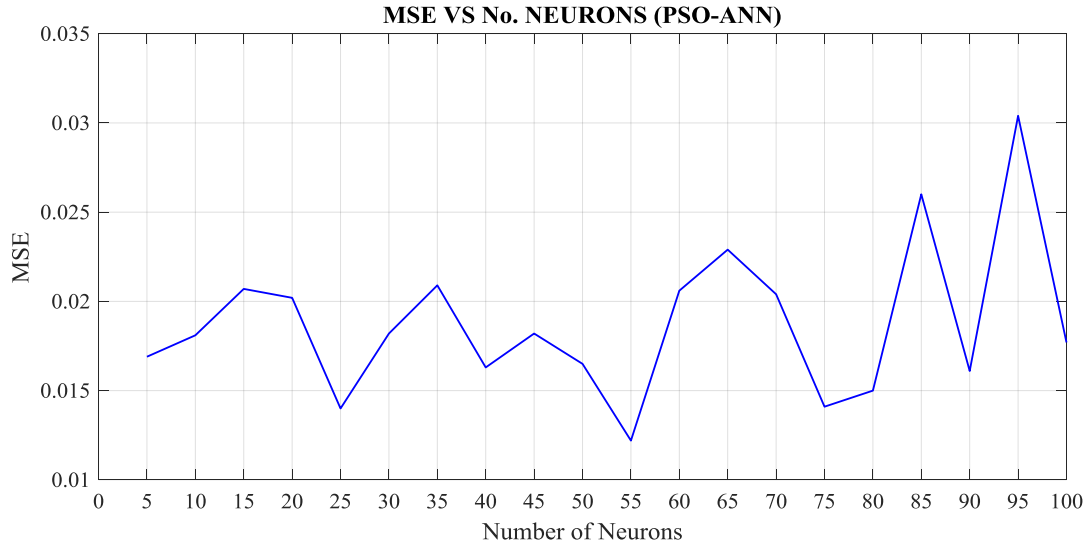


Figure 3-8: PSO-ANN Model Performance Nodes Based on the MSE

Table 3-6: HGA-ANN Model Performance

Day	Network	inputs	hidden neuron	MSE	MAPE	R
Weekday	GA-ANN	24	5	0.025	0.201	0.986
			10	0.017	0.138	0.996
			15	0.011	0.020	0.978
			20	0.024	0.153	0.965
			25	0.017	0.039	0.954
			30	0.038	0.329	0.995
			35	0.044	0.387	0.986
			40	0.048	0.029	0.968
			45	0.033	0.282	0.951
			50	0.052	0.435	0.964
			55	0.019	0.129	0.970
			60	0.031	0.263	0.953
			65	0.042	0.348	0.935
			70	0.032	0.203	0.866
			75	0.029	0.250	0.961
			80	0.019	0.094	0.995
			85	0.015	0.109	0.955
			90	0.025	0.102	0.863
95	0.028	0.239	0.955			
100	0.034	0.022	0.962			

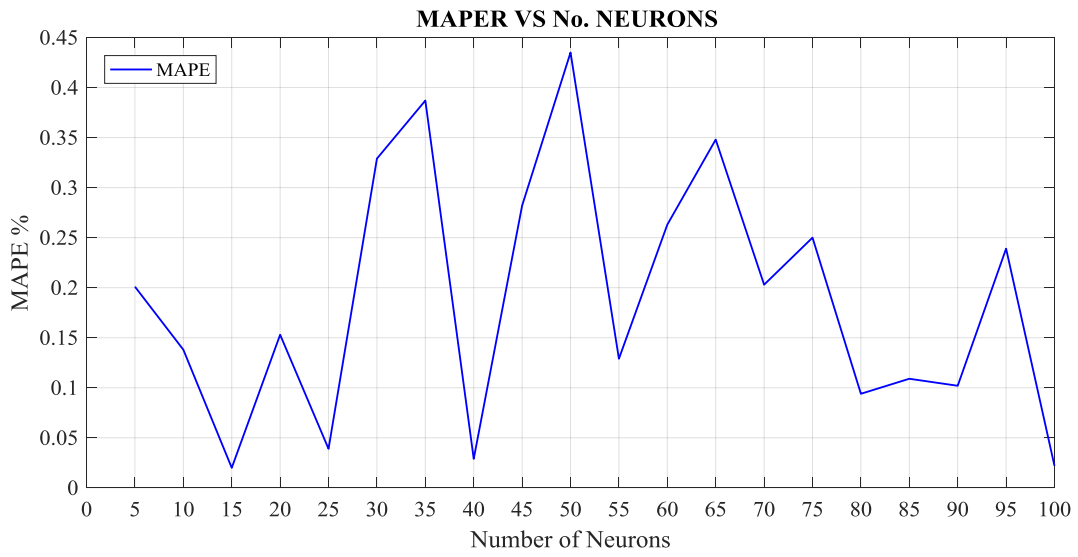


Figure 3-9: HGA-ANN Model Performance Based on the MAPE

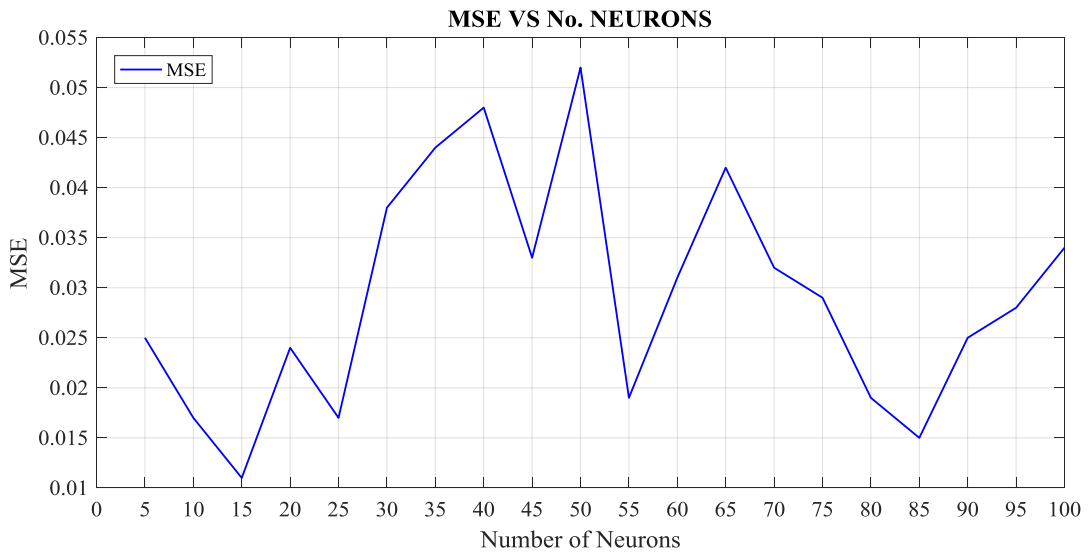


Figure 3-10: GA-ANN Model Performance Based on the MSE

CHAPTER 4 : RESULTS AND DISCUSSION

This chapter presents the result obtained for different 24-hour load forecast using learned ANNs. MAPE and MSE for the forecaster inputs are presented showing significance of the approaches and graph plots for the forecaster inputs were also presented to visualized the relationships between the forecast load and the real load. To simplify the size of the forecaster discussion, the correlation analysis results for chosen 24-hour load forecast days are tabularized and discussed. Load forecast results are presented along with 24-hour load forecast profile plots for selected days.

The Tables from 4-1 to 4-3 show the MAPE and MSE obtained by different approaches for different days respectively. The GAPSO-ANN showed the best performances for working and weekend day. Tables4-4 to 4-5 show the daily load correlations for various approaches which indicated a positive correlation between the actual and forecast loadand Table 4-6 presented one Week Forecast models comparison of the forecasters.

Table 4-1: MAPE% (05/02/2010 to 11/02/2010)

MAPE for Different days				
Day		PSO-ANN[70]	GA-ANN[72]	HGAPSO-ANN
Working Day	Monday	0.230	0.048	0.116
	Tuesday	0.272	0.150	0.125
	Wednesday	0.274	0.134	0.185
	Thursday	0.295	0.134	0.200
	Friday	0.318	0.217	0.253
Weekend	Saturday	0.295	0.196	0.030
	Sunday	0.282	0.069	0.003
Average		0.281	0.135	0.130

Table 4-2: MAPE (25/07/2010 to 31/07/2010)

MAPE for Different days				
Day		PSO-ANN[70]	GA-ANN[72]	HGAPSO-ANN
Working Day	Monday	0.014	0.024	0.139
	Tuesday	0.160	0.164	0.011
	Wednesday	0.183	0.167	0.035
	Thursday	0.123	0.093	0.008
	Friday	0.080	0.058	0.062
Weekend	Saturday	0.019	0.089	0.089
	Sunday	0.040	0.194	0.002
Average		0.088	0.113	0.049

Table 4-3: MAPE% (06/11/2010 to 13/11/2010)

MAPE for Different days				
Day		PSO-ANN[70]	GA-ANN[72]	HGAPSO-ANN
Working Day	Monday	0.118	0.061	0.083
	Tuesday	0.041	0.076	0.030
	Wednesday	0.004	0.122	0.074
	Thursday	0.044	0.053	0.015
	Friday	0.015	0.087	0.048
Weekend	Saturday	0.137	0.101	0.015
	Sunday	0.144	0.069	0.013
Average		0.072	0.081	0.040

Table 4-4: Correlation R (04-10/01/2010) and Correlation R (06-13/11/2010)

Daily Load R 2010 Networks				Daily Load R 2010 Network			
Day	PSO-ANN	GA-ANN	HGAPSO-ANN	Day	PSO-ANN	GA-ANN	HGAPSO-ANN
Working Day	0.992	0.964	0.975	Working Day	0.963	0.957	0.961
	0.995	0.974	0.983		0.969	0.957	0.964
	0.994	0.993	0.994		0.982	0.977	0.982
	0.997	0.995	0.997		0.964	0.952	0.959
	0.982	0.985	0.987		0.979	0.972	0.977
Weekend	0.949	0.962	0.964	Weekend	0.968	0.991	0.990
	0.990	0.993	0.993		0.964	0.989	0.987

Table 4-5: Correlation R (06-13/11/2010) and Correlation R (25-31/07/2010)

Daily Load R (06-13/11/2010)Network				Daily load correlation (25-31/7)network			
Day	PSO-ANN	GA-ANN	HGAPSO-ANN	Day	PSO-ANN	GA-ANN	HGAPSO-ANN
Working Day	0.982	0.988	0.991	Working Day	0.988	0.993	0.995
	0.991	0.993	0.993		0.969	0.972	0.969
	0.979	0.985	0.988		0.977	0.978	0.976
	0.990	0.992	0.992		0.987	0.989	0.988
	0.989	0.991	0.992		0.995	0.998	0.995
Weekend	0.969	0.979	0.983	Weekend	0.962	0.969	0.968
	0.988	0.993	0.995		0.929	0.950	0.946

Table 4-6: One Week Forecast models comparison

Week Forecast			
NETWORK	MAPE	APE	R
PSO-ANN	0.002	0.276	0.992
GA-ANN	0.010	1.691	0.990
HGAPSO-ANN	0.000	0.009	0.987

4.1 Load Forecast Results

The 24-hour-ahead load forecast results for selected days from a week are tabularized in Tables 4-7 to 4-11. The MAPE results in Tables 4-7 to 4-11 for forecasters (PSO-ANN, GA-ANN, and HGAPSO-ANN) have an approximate average range of 1.99% to 3.04 %, 2.23% to 2.64%, and 1.47% to 1.98% respectively. The results obtained are compared to MAPE results found in the STLF [70][71][72][73].

The higher error values highlighted with a yellow colour and the minimum error value highlighter with a green colour represent hours where the actual load profile experienced a planned or unexpected outages or other abrupt load change of the system.

Table 4-7: 24-Hourly forecast results date 19/07/2010

Hour	PSO-ANN[70]			GA-ANN[72]		GAPSO-ANN	
	Actual Load (kW)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)
1	3209	3296	2.739	3142	2.067	3237	0.895
2	2941	2820	4.103	2989	1.637	2843	3.308
3	2785	2754	1.113	2926	5.055	2781	0.143
4	2680	2721	1.524	2792	4.171	2748	2.536
5	2662	2716	2.032	2786	4.685	2743	3.047
6	2792	2756	1.256	2928	4.897	2783	0.291
7	2957	2829	4.359	2997	1.324	2851	3.594
8	3295	3174	3.652	3207	2.677	3362	2.036
9	3612	3763	4.184	3516	2.656	3571	1.140
10	3922	3965	1.075	3917	0.141	3870	1.338
11	4063	4200	3.371	4113	1.231	3968	2.344
12	4120	4292	4.175	4192	1.745	4048	1.741
13	4563	4863	6.575	4722	3.495	4616	1.167
14	4976	5135	3.205	5019	0.865	4969	0.136
15	5321	5237	1.585	5148	3.262	5136	3.487
16	5557	5272	5.131	5498	1.055	5503	0.968
17	5698	5485	3.744	5319	6.649	5631	1.177
18	5701	5585	2.027	5420	4.928	5532	2.966
19	5489	5264	4.112	5186	5.528	5387	1.871
20	5215	5214	0.023	5116	1.888	5094	2.309
21	4999	5145	2.914	5230	4.624	4983	0.313
22	4720	4892	3.655	4858	2.920	4773	1.118
23	4224	4352	3.023	4332	2.563	4193	0.723
24	3662	3538	3.382	3575	2.370	3530	3.615
AVERAGE(MAPE)			3.040		3.618		1.875

Table 4-8:24-Hourly forecast results date 20/07/2010

Hour	PSO_ANN[70]			GA_ANN[72]		GA_PSO_ANN	
	Actual Load (kW)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)
1	3237	3233	0.097	3163	2.286	3118	3.673
2	2955	3009	1.843	2995	1.383	2850	3.540
3	2756	2883	4.615	2916	5.798	2771	0.553
4	2644	2823	6.794	2781	5.217	2638	0.207
5	2612	2708	3.657	2763	5.762	2660	1.820
6	2715	2860	5.365	2803	3.240	2758	1.612
7	2915	2982	2.293	2977	2.134	2832	2.854
8	3239	3236	0.105	3164	2.321	3119	3.706
9	3606	3702	2.654	3509	2.693	3564	1.173
10	3954	3972	0.462	3961	0.183	3814	3.530
11	4285	4354	1.606	4411	2.938	4276	0.208
12	4529	4589	1.312	4689	3.532	4579	1.092
13	4733	4756	0.477	4868	2.844	4784	1.086
14	4762	4777	0.317	4889	2.673	4810	1.011
15	4831	4876	0.928	4936	2.179	4867	0.741
16	4933	4892	0.830	4996	1.290	4941	0.166
17	5097	4984	2.208	5074	0.455	5039	1.138
18	5149	5010	2.700	5094	1.085	5065	1.647
19	4947	4901	0.943	5004	1.141	4950	0.058
20	4737	4658	1.655	4871	2.824	4788	1.080
21	4548	4605	1.258	4708	3.520	4600	1.142
22	4407	4475	1.554	4558	3.445	4434	0.627
23	3934	3999	1.644	3933	0.029	3886	1.221
24	3493	3579	2.451	3386	3.068	3442	1.483
AVERAGE(MAPE)			1.990		2.585		1.474

Table 4-9: 24-Hourly forecast results date 22/07/2010

Hour	PSO-ANN[70]			GA-ANN[72]		GAPSO-ANN	
	Actual Load (kW)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)
1	3308	3208	3.035	3217	2.750	3272	1.077
2	3005	2942	2.104	3020	0.486	2874	4.354
3	2777	2880	3.722	2923	5.260	2778	0.048
4	2641	2758	4.419	2881	9.071	2737	3.644
5	2603	2703	3.838	2670	2.587	2727	4.783
6	2675	2563	4.187	2790	4.327	2747	2.691
7	2853	2897	1.534	2951	3.439	2806	1.649
8	3226	3150	2.356	3155	2.201	3110	3.594
9	3524	3429	2.691	3419	2.985	3574	1.421
10	3851	3867	0.423	3819	0.839	3772	2.056
11	4165	4333	4.019	4253	2.121	4112	1.284
12	4404	4525	2.747	4555	3.428	4431	0.601
13	4610	4775	3.571	4766	3.375	4666	1.204
14	4816	4962	3.028	4926	2.280	4855	0.794
15	4962	5000	0.774	5012	1.001	4960	0.039
16	5086	5024	1.228	5069	0.332	5033	1.042
17	5158	5034	2.413	5097	1.187	5069	1.728
18	5137	5031	2.069	5089	0.936	5059	1.525
19	4968	5002	0.682	5015	0.941	4964	0.082
20	4686	4913	4.842	4831	3.102	4741	1.186
21	4508	4611	2.274	4668	3.546	4555	1.039
22	4368	4490	2.789	4513	3.312	4385	0.383
23	3850	3865	0.388	3817	0.851	3671	4.665
24	3397	3286	3.272	3294	3.044	3249	4.355
AVERAGE(MAPE)			2.600		2.642		1.885

Table 4-10: 24-Hourly forecast results date 28/07/2010

Hour	PSO-ANN[70]			GA-ANN[72]		GAPSO-ANN	
	Actual Load (kW)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)
1	2767	2748	0.691	2920	5.517	2775	0.289
2	2531	2685	6.092	2553	0.842	2711	7.095
3	2364	2458	3.992	2419	2.322	2380	0.694
4	2281	2388	4.673	2405	5.409	2268	0.578
5	2267	2246	0.908	2303	1.567	2366	4.380
6	2368	2359	0.395	2419	2.168	2281	3.684
7	2574	2694	4.698	2603	1.141	2621	1.825
8	2908	2804	3.580	2974	2.263	2829	2.736
9	3250	3132	3.612	3172	2.392	3127	3.772
10	3612	3563	1.353	3516	2.656	3571	1.139
11	3934	3984	1.280	3933	0.021	3786	3.755
12	4170	4170	0.017	4260	2.170	4119	1.226
13	4406	4505	2.255	4557	3.437	4433	0.616
14	4724	4895	3.635	4861	2.906	4776	1.117
15	4949	5124	3.517	5005	1.122	4952	0.045
16	5142	5194	1.018	5091	0.985	5061	1.562
17	5226	5216	0.180	5120	2.024	5099	2.423
18	5232	5218	0.274	5122	2.107	5102	2.494
19	5085	5176	1.790	5069	0.325	5032	1.038
20	4875	4988	2.308	4963	1.810	4900	0.511
21	4713	4887	3.710	4852	2.965	4766	1.140
22	4502	4463	0.884	4662	3.542	4548	1.015
23	4145	4232	2.103	4226	1.973	4084	1.469
24	3578	3514	1.778	3477	2.804	3532	1.266
AVERAGE(MAPE)			2.281		2.270		1.911

Table 4-11: 24-Hour forecast hourly results date 30/07/2010

Hour	PSO-ANN[70]			GA-ANN[72]		GAPSO-ANN	
	Actual Load (kW)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)	Forecast Load (kW)	APE (%)
1	2874	2789	2.955	2960	2.988	2815	2.066
2	2635	2709	2.806	2709	2.808	2736	3.823
3	2457	2500	1.765	2496	1.619	2496	1.603
4	2316	2352	1.550	2410	4.071	2373	2.456
5	2322	2353	1.307	2311	0.473	2374	2.217
6	2390	2362	1.200	2423	1.380	2384	0.254
7	2567	2693	4.901	2661	3.664	2619	2.016
8	2734	2737	0.116	2809	2.736	2764	1.113
9	2971	2836	4.569	3003	1.067	2858	3.828
10	3214	3101	3.512	3146	2.111	3201	0.398
11	3447	3341	3.073	3341	3.089	3496	1.421
12	3645	3512	3.642	3555	2.474	3509	3.721
13	3808	3772	0.955	3761	1.243	3714	2.468
14	3953	4016	1.595	4059	2.685	3813	3.558
15	4040	4162	3.027	4081	1.019	3935	2.589
16	4093	4249	3.801	4155	1.501	4020	1.784
17	4102	4263	3.945	4167	1.600	4023	1.916
18	4104	4267	3.963	4170	1.608	4026	1.903
19	3970	4045	1.901	3883	2.178	3936	0.835
20	3728	3641	2.316	3657	1.899	3611	3.135
21	3598	3443	4.323	3500	2.731	3655	1.573
22	3460	3457	0.075	3353	3.075	3409	1.473
23	3123	3151	0.899	3086	1.177	3105	0.577
24	2819	2867	1.675	2938	4.215	2793	0.928
AVERAGE (MAPE)			2.495		2.225		1.986

Figures 4-1 through 4-13 illustrate the forecasted and actual load shapes for the 24-hour period. These plots also show the absolute percent error (APE) profile over the 24-hour period.

Tables 4-7 through 4-11 lists the hourly actual load, predicted load, and associated with each absolute percent errors (APE) forecasted load profile, in Figures 4-1 through 4-013 show the load get peak between 05:00-07:00 hours and 15:00- 17:00 hours, when the load advanced toward the peak from minimum load levels, and leftward the peak to minimum load levels.

The higher error values on 19 July 2010 occurred during peak load level and on 28 July 2010 occurred during minimum load level. This demonstrates the chaotic load profiles experienced during the weekdays. The weekdays load profiles are not as smooth as the profiles during the weekend load profiles, so the artificial neural network ANN has a drawback in predicting the changes in the load profile.

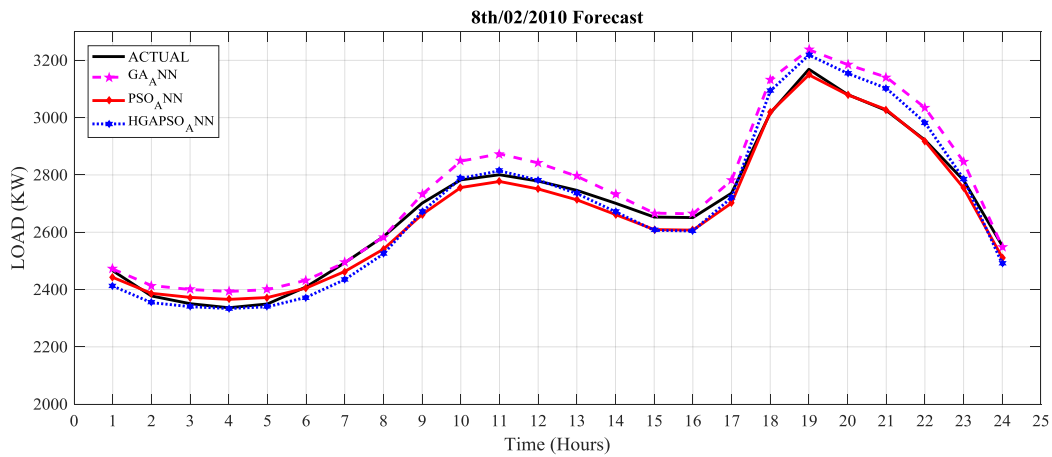


Figure 4-1: 24-Actual and forecasted load profile for 8th February 2010

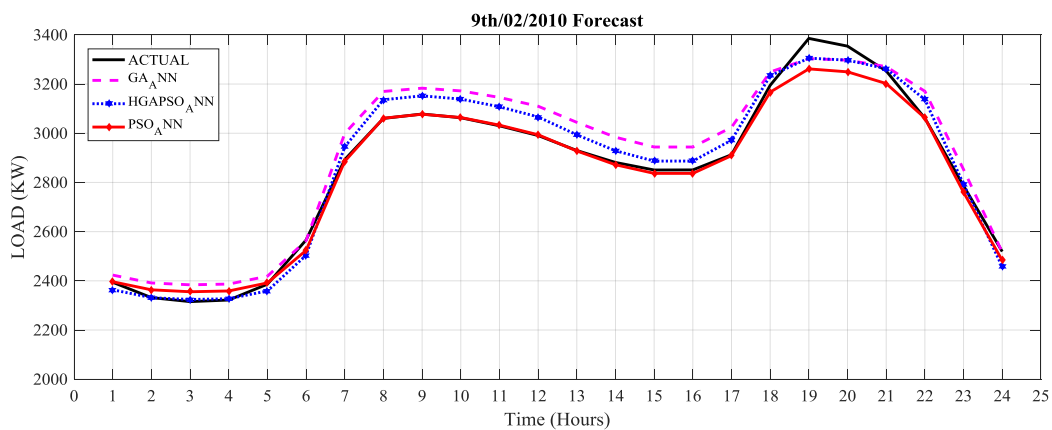


Figure 4-2: 24-Actual and forecasted load profile for 9th February 2010

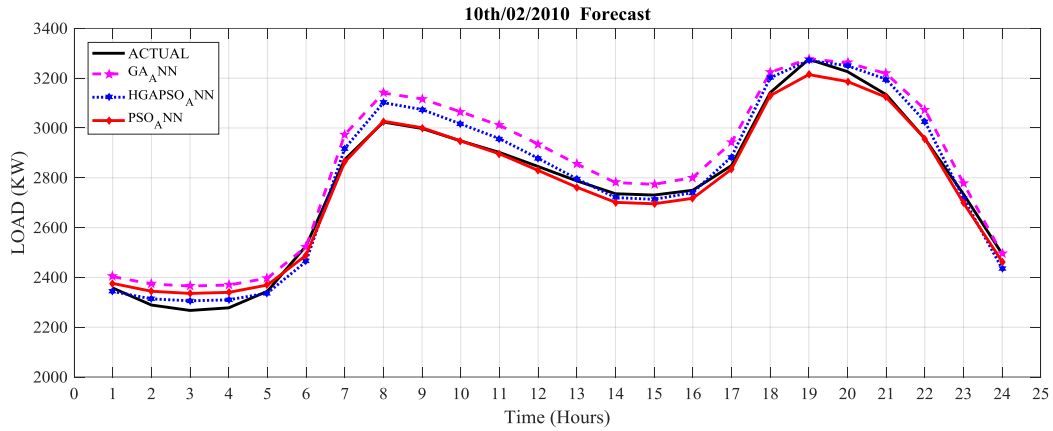


Figure 4-3: 24-Actual and forecasted load profile for 10th February 2010

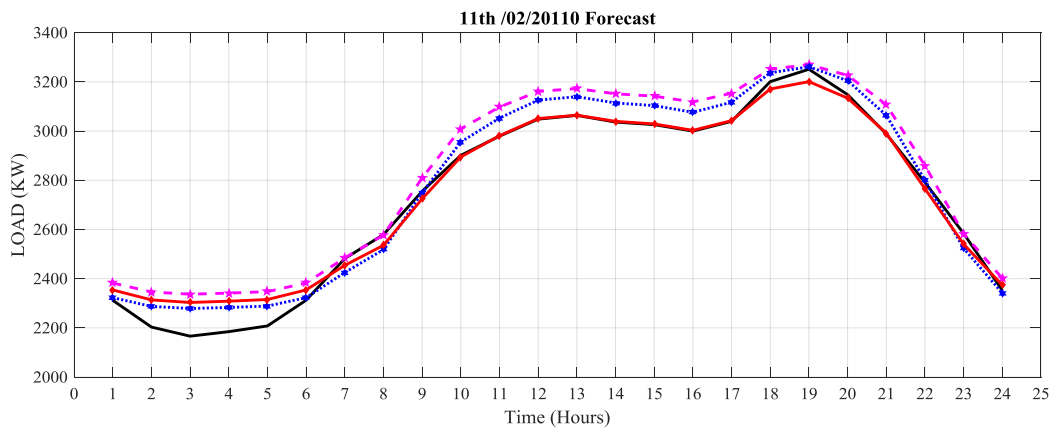


Figure 4-4: 24-Actual and forecasted load profile for 11th February 2010

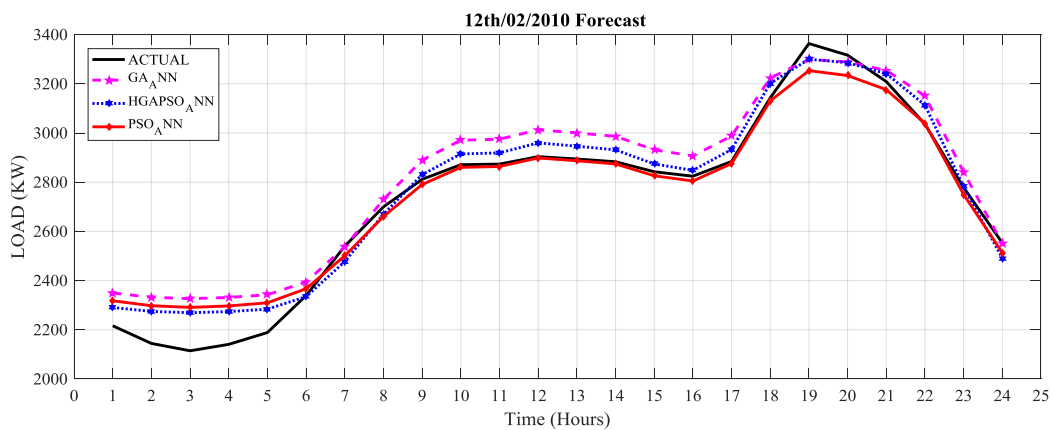


Figure 4-5: 24-Actual and forecasted load profile for 12th February 2010

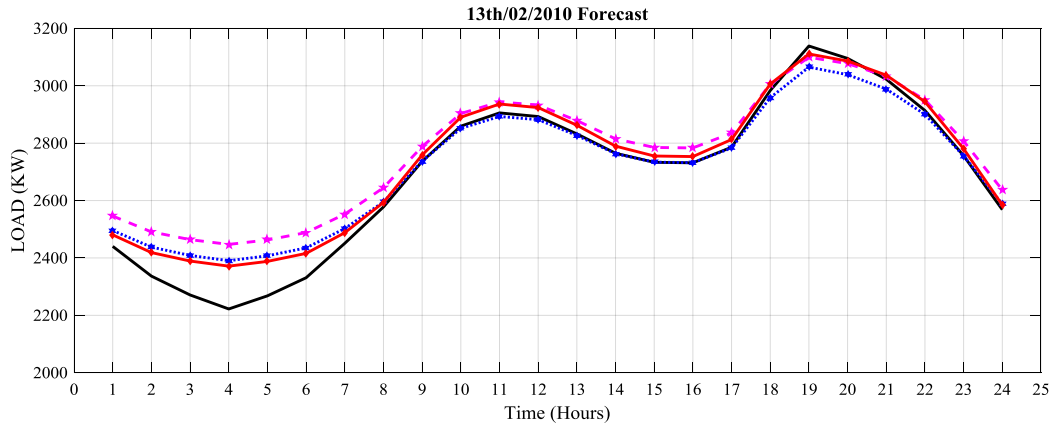


Figure 4-6: 24-Actual and forecasted load profile for 13th February 2010

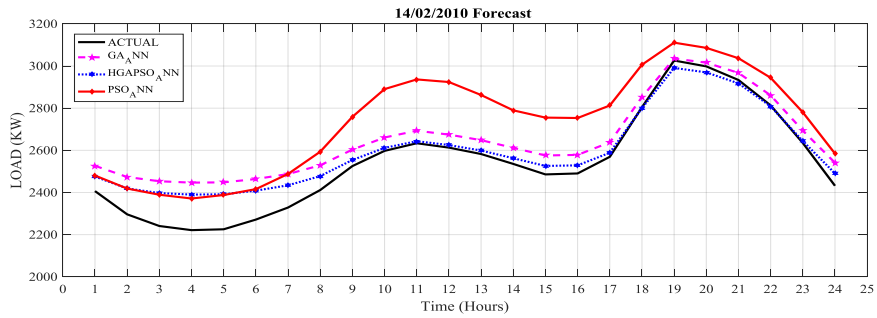


Figure 4-7: 24-Actual and forecasted load profile for 14th February 2010

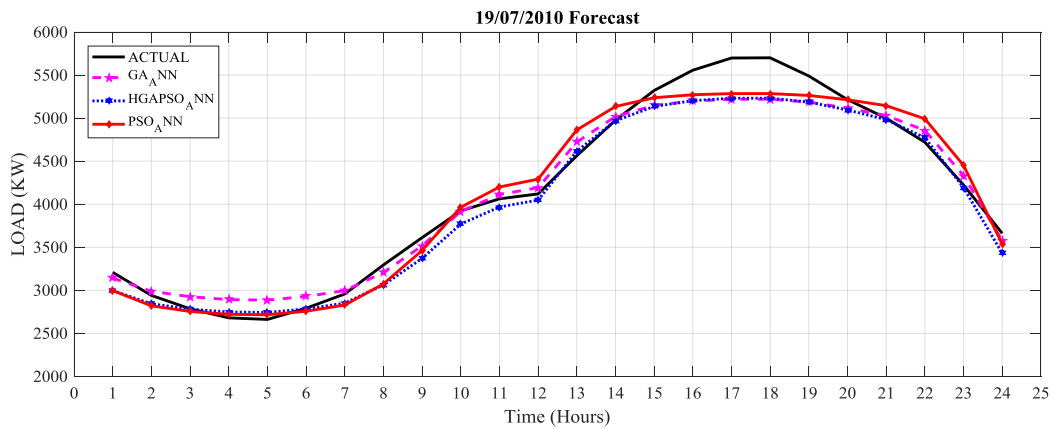


Figure 4-8: 24-Actual and forecasted load profile for 19th July 2010

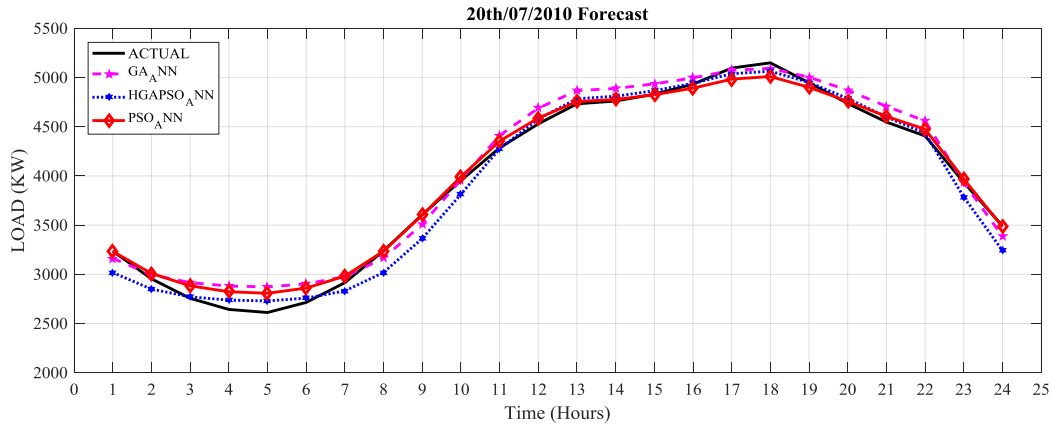


Figure 4-9: 24-Actual and forecasted load profile for 20th July 2010

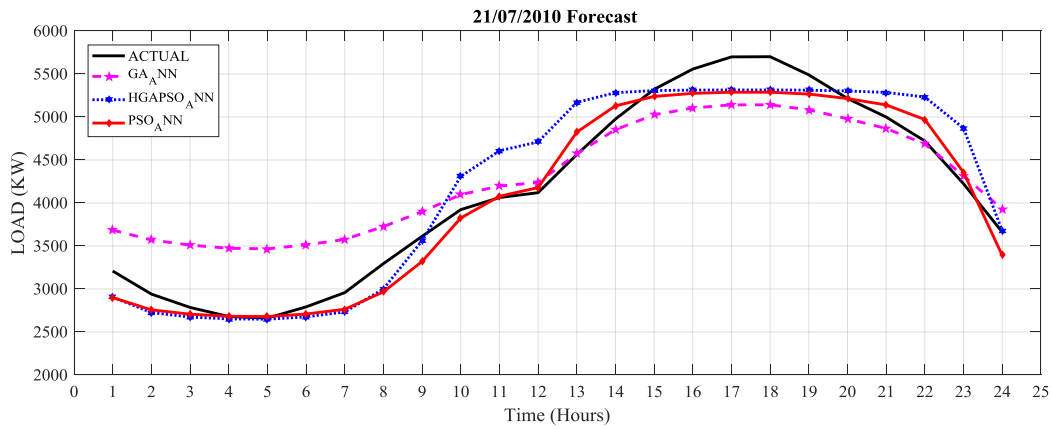


Figure 4-10: 24-Actual and forecasted load profile for 21th July 2010

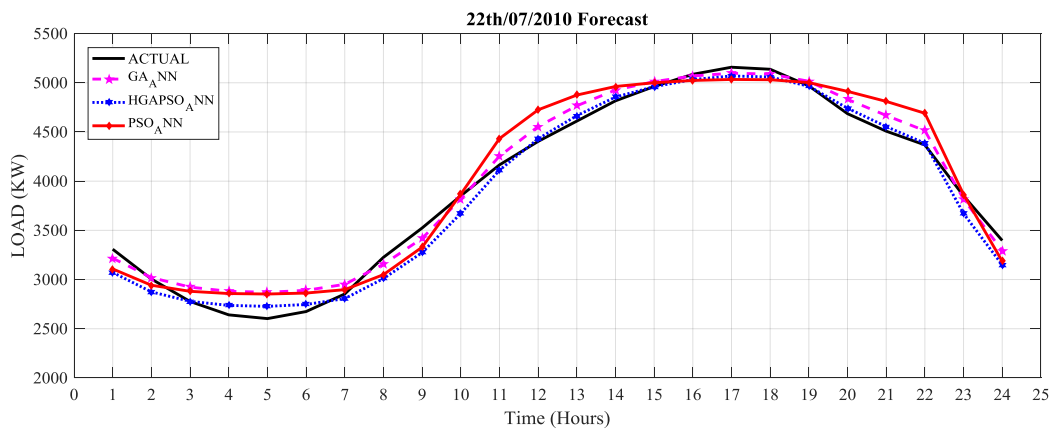


Figure 4-11: 24-Actual and forecasted load profile for 22th July 2010

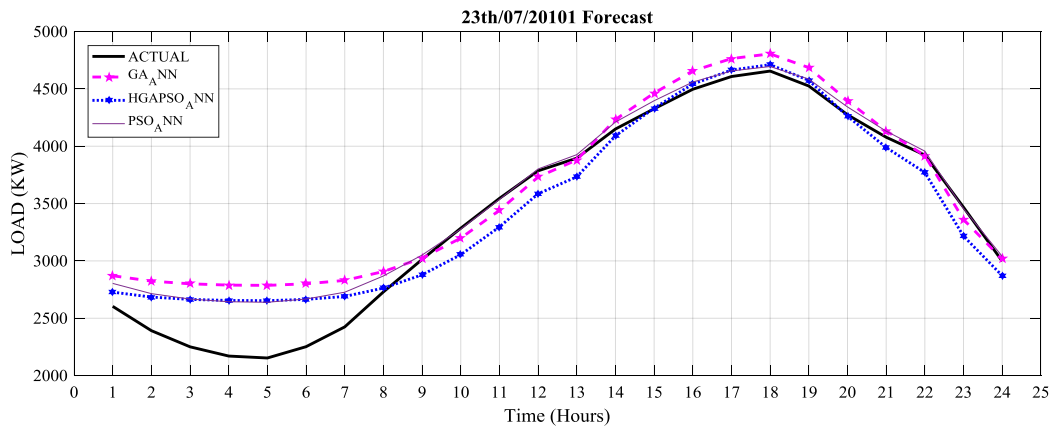


Figure 4-12: 24-Actual and forecasted load profile for 23th July 2010

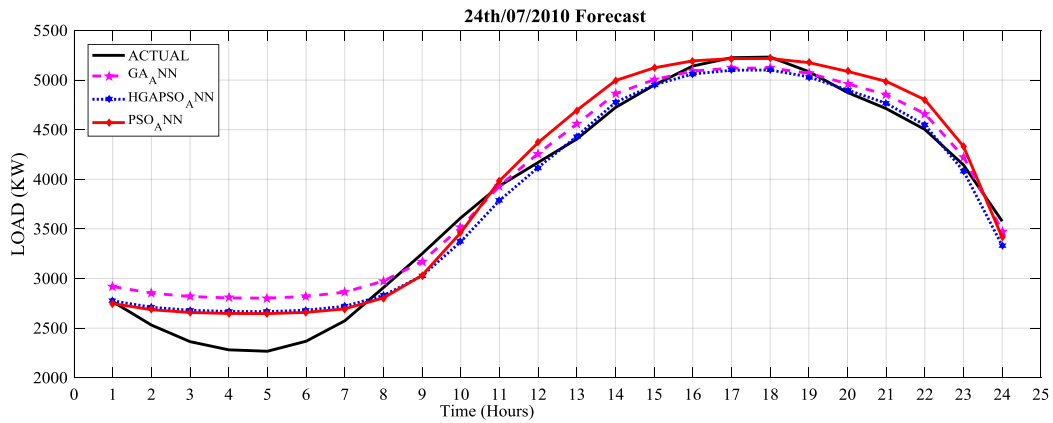


Figure 4-13: 24-Actual and forecasted load profile for 24th July 2010

Table 4-12 and Figure 4-14 show the comparison of mean average percentage errors of respective days for different methods. This comparison is for checking which method gives best results, the GAPSO based ANN model results has lowest minimum MAPE as compared to the other methods.

Table 4-12: Comparison of MAPE of different methods

DATE	PSO	GA[72]	GAPSO
	MAPE (%)	MAPE (%)	MAPE (%)
19/07/2010	3.04	3.618	1.875
20/07/2010	1.99	2.585	1.474
22/07/2010	2.60	2.642	1.884
28/07/2010	2.495	2.225	1.911
30/07/2010	2.495	2.225	1.986
Average	2.524	2.659	1.826

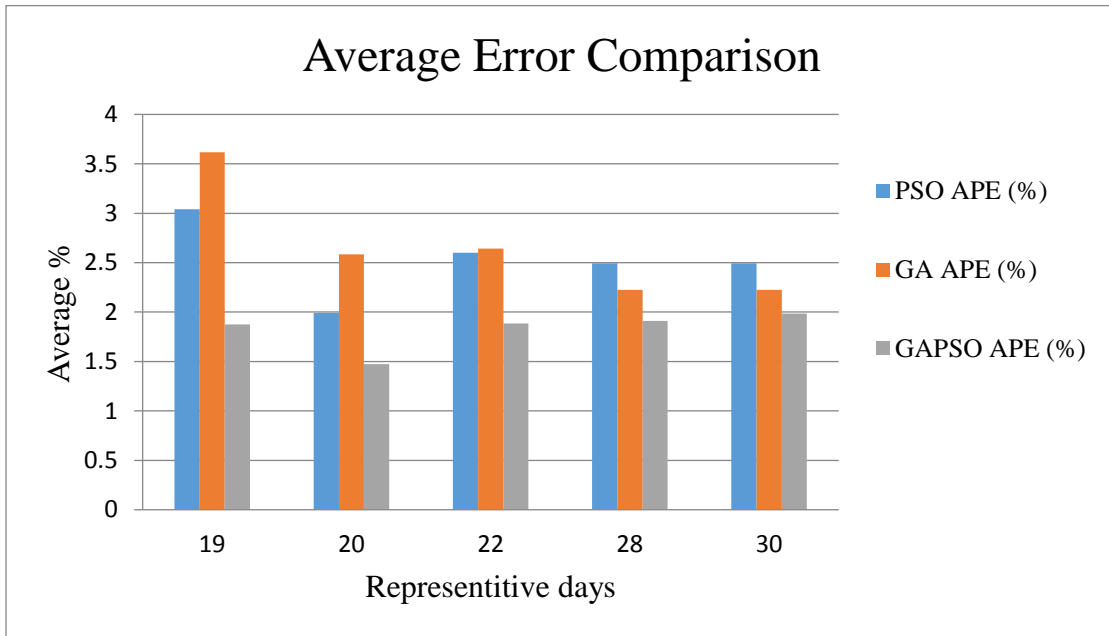


Figure 4-14: Comparison of average error of different methods

CHAPTER 5 : CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The planning, control, and operation of electric power utilities involve various elements which determine the degree and the performance of power systems, load prediction is of one these elements. It's most essential in electric utilities as it allows power system operators to control and plan for their power system operations. The necessity of the prediction is that the forecaster should have an accurate error precision as possible; this will ensure that a utility is able to minimize its generation costs by providing the operators with necessary information for making decisions regarding scheduling equipment for maintenance as well as energy purchasing. Therefore, a large load forecast error may affect the economic viability of the Power Company as well as reliability of power systems.

In this research, the main objective was to develop a hybrid forecaster of Genetic Algorithm with Particle Swarm Optimization for learned Artificial Neural Network. The models HGAPSO, PSO and GA-ANN were tested using data obtained from Juba Power Station (JPP) and JIAWS and final results were obtained. The HGAPSO, PSO and GA-ANN results were compared to determine the best model and their performances were assessed using the mean absolute percentage error (MAPE). The resulting mean absolute percentage error (MAPE) of HGAPSO found to be range from 1.47% to 1.98%. Therefore, a hybridized HGAPSO algorithm with ANN improves forecast accuracy.

A hybrid PSO-ANN and GA-ANN resulting mean absolute percentage error (MAPE) was found to be range from 1.9% to 3.40%, and 2.23% to 3.65% respectively. Therefore, by introducing hybridization concept, the minimum forecast error results can be obtained.

The observation from this work is that a hybrid of GAPSO-ANN has better forecasting results, by comparing results of these three modeling functions. Thus, concluded that the hybrid HGAPSO-ANN has lower mean absolute performance error (MAPE) in all selected days, and it was established that the lowest mean absolute percentage error (MAPE), is about 1.47% and 1.98%. Therefore, a hybridized HGAPSO algorithm with ANN improves forecast accuracy.

5.2 Beneficiaries of this work

The beneficiaries of this work both directly and indirectly, are Power systems operators, distribution and retail energy providers in price settings. The direct beneficiaries from this work include;

- This research work will help Electric Power Generation Company in reducing both generations cost and spinning reserve in their networks. This reduction in generation cost will enable them to enhance the transmission cost, distribution cost and hence improvement in their profits.
- The research work will also ensure that the Power System Operators will carry on their daily activity effectively, such as avoiding overloading and reduce occurrences of equipment failures, schedule spinning reserve allocation properly.
- Other beneficiaries of this research work are the end users who will have enough supply demand for their uses.

5.2 Recommendations

The research recommendations in this study have been based on the limitations of the study and they are:

1. There is a need to investigate the application of the proposed model for Mid-Term and Long-Term Load forecasting
2. There is a need to incorporate weather uncertainty factors such Dewpoint, Wind Speed/Wind Direction, Dry and wet bulb, and Sky Cover to proposed model
3. The study should be carry on application of HGAPSO for different classes of customer such as commercial, residential, and industrials load forecast.
4. More study should carry on how to determining the hidden neuron of ANN,

REFERENCES

- [1] D. . K. and I. . Nagrath, *Modern Power System Analysis*, Third Edit. New Delhi: Tata McGraw Hill Education Private Limited, 2003.
- [2] Hossein Seifi and Mohammed Sadegh Sepasian, *Electric Power System Planning*, S. Springer Heidelberg Dordrecht London New Yerk, 2011.
- [3] H. L. Willis, *Spatial Electric Load Forecasting*, Second Edi. New York.: Marcel Dekker, Inc., 2002.
- [4] F. Javed, N. Arshad, F. Wallin, I. Vassileva, and E. Dahlquist, “Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting,” *Appl. Energy*, vol. 96, pp. 150–160, 2012.
- [5] E. a Feinberg and D. Genethliou, “Load Forecasting,” *Appl. Math. Restructured Electr. Power Syst.*, pp. 269–285, 2005.
- [6] F. J. Nogales, J. Contreras, A. J. Conejo, and R. Espínola, “Forecasting next-day electricity prices by time series models,” *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 342–348, 2002.
- [7] F. Elkarmi and N. Abu Shikhah, *Power System Planning Technologies and Applications*. 2012.
- [8] E. Almeshaie and H. Soltan, “A methodology for Electric Power Load Forecasting,” *Alexandria Eng. J.*, vol. 50, no. 2, pp. 137–144, 2011.
- [9] P. S. R. Murty, *Power System Analysis*. Giriraj Lane, Sultan Bazar, BS Publications, 2007.
- [10] M. C. W Charytoniuk, “Very Short term load forecasting using Artificial Neural Networks”, IEEE Transactions on Power Systems,” *IEEE Trans. Power Syst.*, vol. Vol.15, no. No. 1, p. PP .263 – 268.
- [11] J. Automation, S. L. Forecasting, U. Time, and A. Moving, “2 . Time Series and Data Mining 3 . Forecasting Methods,” vol. 3, pp. 122–132, 2014.
- [12] H. M. Al-Hamadi and S. a. Soliman, “Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model,” *Electr. Power Syst. Res.*, vol. 68, no. 1, pp. 47–59, 2004.

- [13] M. Ghofrani, M. Ghayekhloo, A. Arabali, and A. Ghayekhloo, "A hybrid short-term load forecasting with a new input selection framework," *Energy*, vol. 81, pp. 777–786, 2015.
- [14] A. Jain and B. Satish, "Integrated Approach for Short Term Load Forecasting using SVM and ANN," no. March, pp. 1–6, 2009.
- [15] A. K. Topalli and I. Erkmen, "A hybrid learning for neural networks applied to short term load forecasting," *Neurocomputing*, vol. 51, pp. 495–500, 2003.
- [16] S. A. H.M. Al-Hamadi, Soliman, "Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model," *Electr. Power Syst. Res.*, vol. 68, 2004.
- [17] J. Nowicka-zagrajek and R. Weron, "Modeling electricity loads in California : ARMA models with hyperbolic noise," *Signal Procesign*, vol. 82, pp. 1903–1915, 2002.
- [18] H. Nie, G. Liu, X. Liu, and Y. Wang, "Energy Procedia Hybrid of ARIMA and SVMs for Short-Term Load Forecasting," *Int. Conf. Futur. Energy, Environ. Mater.*, vol. 16, no. 2011, pp. 1455–1460, 2012.
- [19] M. De Felice, A. Alessandri, and P. M. Ruti, "Electricity demand forecasting over Italy : Potential benefits using numerical weather prediction models," *Electr. Power Syst. Res.*, vol. 104, pp. 71–79, 2013.
- [20] D. Kriesel, *A Brief Introduction to Neural Networks*. 2005.
- [21] A. Krenker, J. Bešter, and A. Kos, "Introduction to artificial neural networks.," *Eur. J. Gastroenterol. Hepatol.*, vol. 19, no. 12, pp. 1046–1054, 2011.
- [22] D. Kriesel, *Neural Networks*. 2013.
- [23] K. Gurney, *An introduction to neural networks*, vol. 14, no. 1. 1997.
- [24] A. Abraham, *Neural Networks*. 2005.
- [25] A. P. Alves and L. S. Moulin, "Confidence Intervals for Neural Network Based Short-Term Load Forecasting," *IEE Trans. Power Syst.*, vol. 15, no. 4, pp. 1191–1196, 2000.
- [26] S. Quaiyum, Y. I. Khan, S. Rahman, and P. Barman, "Artificial Neural Network based Short Term Load Forecasting of Power System," *Int. J. Comput. Appl.*, vol. 30, no. 4, pp. 1–7, 2011.

- [27] I. K. Ibraheem, D. Ph, and M. O. Ali, “Short Term Electric Load Forecasting based on Artificial Neural Networks for Weekends of Baghdad Power Grid,” *Int. J. Comput. Appl.*, vol. 89, no. 3, pp. 30–37, 2014.
- [28] S. Abdel-Hady, *Modern Optimisation Techniques With Applications In Electrical Power Systems*. 2010.
- [29] I. Drezga, “Input Variable Selection For Ann-Based Short-Term Load Forecasting,” *IEE Trans. Power Syst.*, vol. 13, no. 4, pp. 1238–1244, 1998.
- [30] L. Li, Song Wang, Peng Goel, “Short-term load forecasting by wavelet transform and evolutionary extreme learning machine,” *Electr. Power Syst. Res.*, vol. 122, pp. 96–103, 2015.
- [31] S. a. Soliman, *Fuzzy Regression Systems and Fuzzy Linear Models*. 2010.
- [32] I. O. Harrison, “Short Term Electric Load Forecasting of 132 / 33KV Maiduguri Transmission Substation using Adaptive Neuro-Fuzzy Inference System (ANFIS),” *Int. J. Comput. Appl.*, vol. 107, no. 11, pp. 23–29, 2014.
- [33] J. López, M. Valero, S. Senabre, C. Aparicio and A. Gabaldon, “Application of SOM neural networks to short-term load forecasting: The Spanish electricity market case study,” *Electr. Power Syst. Res.*, vol. 91, pp. 18–27, 2012.
- [34] B. Wang, N. Tai, H. Zhai, J. Ye, J. Zhu, and L. Qi, “A new ARMAX model based on evolutionary algorithm and particle swarm optimization for short-term load forecasting,” *Electr. Power Syst. Res.*, vol. 78, pp. 1679–1685, 2008.
- [35] C. C. Pandian, S Chenthur Duraiswamy, K Asir and N. Kanagaraj, “Fuzzy approach for short term load forecasting,” *Electr. Power Syst. Res.*, vol. 76, pp. 541–548, 2006.
- [36] a. S. Khwaja, M. Naeem, a. Anpalagan, a. Venetsanopoulos, and B. Venkatesh, “Improved short-term load forecasting using bagged neural networks,” *Electr. Power Syst. Res.*, vol. 125, pp. 109–115, 2015.
- [37] A. Abdoos, M. Hemmati, and A. A. Abdoos, “Knowledge-Based Systems Short term load forecasting using a hybrid intelligent method,” *Knowledge-Based Syst.*, no. December 2014.
- [38] S. Chenthur Pandian, K. Duraiswamy, C. C. A. Rajan, and N. Kanagaraj, “Fuzzy approach for short term load forecasting,” *Electr. Power Syst. Res.*, vol. 76, no. 6–7, pp. 541–548, 2006.

- [39] M. M. Othman and I. Musirin, "Expert Systems with Applications A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 541–548, 2012.
- [40] A. Badri, Z. Ameli, and A. M. Birjandi, "Application of Artificial Neural Networks and Fuzzy logic Methods for Short Term Load Forecasting," *Energy Procedia*, vol. 14, no. 2011, p. 1, 2012.
- [41] S. S. Pappas *et al.*, "Electricity demand load forecasting of the Hellenic power system using an ARMA model," *Electr. Power Syst. Res.*, vol. 80, no. 3, pp. 256–264, 2010.
- [42] M. Buhari and S. S. Adamu, "Short-Term Load Forecasting Using Artificial Neural Network," *Proceeding Int. MultiConference Eng. Comput. Sci.*, vol. I, 2012.
- [43] G. Liao and T. Tsao, "Application of fuzzy neural networks and artificial intelligence for load forecasting," *Electr. Power Syst. Res.*, vol. 70, pp. 237–244, 2004.
- [44] K. Sastry and D. Goldberg, *Genetic Algorithms*. 1975.
- [45] M. G. R. Cheng, *Genetic Algorithms & Engineering Optimization*, Hamid R. P. JOHN WILEY & SOBS, INC, 2000.
- [46] Q. Zhu and A. T. Azar, *Complex System Modelling and Control Through Intelligent Soft Computations*, vol. 319. 2015.
- [47] R. Lowen and A. Verschoren, *Foundations of generic optimization Volume 2*, R. Laubenb. Springer, 2008.
- [48] M. Melanie, *An Introduction to Genetic Algorithms*. Massachusetts Institute of Technology, 1999.
- [49] Z. Honghui and L. Yongqiang, "Application of an Adaptive Network-Based Fuzzy Inference System Using Genetic Algorithm for Short Term Load Forecasting," *2012 Int. Conf. Comput. Sci. Electron. Eng.*, pp. 314–317, 2012.
- [50] B. Islam, Z. Baharudin, Q. Raza, and P. Nallagownden, "Hybrid and Integrated Intelligent System for Load Demand Prediction," in *B. Islam, Z. Baharudin, Q. Raza et al.*, 2013, no. June, pp. 178–183.
- [51] F. Yu and X. Xu, "A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network," *Appl. Energy*, vol. 134, pp. 102–113, 2014.

- [52] P. Li, Y. Li, Q. Xiong, Y. Chai, and Y. Zhang, “Electrical Power and Energy Systems Application of a hybrid quantized Elman neural network in short-term load forecasting,” *Int. J. Electr. Power Energy Syst.*, vol. 55, pp. 749–759, 2014.
- [53] L. Yongchun, “Application of Elman neural network in short-term load forecasting,” in *Proceedings - International Conference on Artificial Intelligence and Computational Intelligence, AICI 2010*, 2010, vol. 2, pp. 141–144.
- [54] S. Yu, K. Wang, and Y. Wei, “A hybrid self-adaptive Particle Swarm Optimization – Genetic Algorithm – Radial Basis Function model for annual electricity demand prediction,” *Energy Convers. Manag.*, vol. 91, pp. 176–185, 2015.
- [55] A. Jain, M. B. Jain, and E. Srinivas, “A Novel Hybrid Method for Short Term Load Forecasting using Fuzzy Logic and Particle Swarm Optimization,” in *International Conference on Power System Technology*, 2010, pp. 1–7.
- [56] P. Duan, K. Xie, T. Guo, and X. Huang, “Short-Term Load Forecasting for Electric Power Systems Using the PSO-SVR and FCM Clustering Techniques,” *Energies*, vol. 4, pp. 173–184, 2011.
- [57] S. Manickam, G. U. Sivam, and T. Yuvapriya, “Particle Swarm Optimization For A Metamaterial Spiral Resonator,” *Int. J. Recent Adv. EGINEERING Technol.*, vol. 1, no. 2, pp. 2347–2812, 2013.
- [58] Z. A. Bashir, “Applying Wavelets to Short-Term Load Forecasting Using PSO-Based Neural Networks,” *IEE Trans. Power Syst.*, vol. 24, no. 1, pp. 20–27, 2009.
- [59] H. M. I. Pousinho, V. M. F. Mendes, and J. P. S. Catalão, “A hybrid PSO – ANFIS approach for short-term wind power prediction in Portugal,” *Energy Convers. Manag.*, vol. 52, no. 1, pp. 397–402, 2011.
- [60] D. K. Chaturvedi, A. P. Sinha, and O. P. Malik, “Electrical Power and Energy Systems Short term load forecast using fuzzy logic and wavelet transform integrated generalized neural network,” *Int. J. Electr. POWER ENERGY Syst.*, vol. 67, pp. 230–237, 2015.
- [61] N. I. E. Shan-kun, W. Yu-jia, X. Shanli, and C. Ke, “A Hybrid of Particle Swarm Optimization And Genetic Algorithm for Training Back-Propagation Neural Network,” *Int. J. Res. Eng. Sci.*, vol. 4, no. 5, pp. 48–58, 2016.
- [62] G. Feng and L. Li, “Application of Genetic Algorithm and Neural Network in Construction Cost Estimate,” in *Proceedings of the 2012 2nd International Conference on Computer and Information Application (ICCIA 2012)*, 2012, no. Iccia, pp. 1036–1039.

- [63] K. A. Tolosa and M. Daño-Luna, “Scoping study on the readiness of the PNP for ISO, focus on the PRO ARMM,” vol. 34, no. 2, pp. 997–1006, 2015.
- [64] Pencheva T, Atanassov K, and Shannon A, “Modelling of a Roulette Wheel Selection Operator in Genetic Algorithms Using Generalized Nets,” *Bioautomation*, vol. 13, no. 4, pp. 257–264, 2009.
- [65] U. Bodenhofer, *Genetic Algorithms: Theory and Applications*, Third Edit. Fuzzy Logic Laboratorium-Hagenberg, 2004.
- [66] N. R. Siriwardene and B. J. C. Perera, “Selection of genetic algorithm operators for urban drainage model parameter optimisation,” vol. 44, pp. 415–429, 2006.
- [67] W. J. DeCoursey, *Statistics and Probability for Engineering Applications*. Elsevier Science (USA), 2003.
- [68] Peter X.-K. Song, *Springer Series in Statistics*, 2nd ed. Springer Science+Business Media, LLC, 2007.
- [69] S. Mishra and S. K. Patra, “Short term load forecasting using neural network trained with genetic algorithm & particle swarm optimization,” *Proc. - 1st Int. Conf. Emerg. Trends Eng. Technol. ICETET 2008*, pp. 606–611, 2008.
- [70] Y. Shangdong, “A New ANN Optimized By Improved PSO Algorithm Combined With Chaos And Its Application In Short-term Load Forecasting,” 2006, pp. 945–948.
- [71] N. K. Singh, A. K. Singh, and M. Tripathy, “Short Term Load Forecasting using Genetically Optimized Radial Basis Function Neural Network,” in *IEEE Power Engineering Internationalconference*, 2014, no. October, pp. 1–5.
- [72] P. N. S. H. I. Ying-ling, “Research on Short-Term Load Forecasting Based on Adaptive Hybrid Genetic Optimization BP Neural Network Algorithm,” no. 70572090, pp. 1563–1568, 2008.
- [73] X. Sun *et al.*, “An Efficient Approach to Short-Term Load Forecasting at the Distribution Level,” vol. 31, no. 4, pp. 2526–2537, 2016.

APPENDICES

APPENDIX A

PSO CODE

```
clc
tic
close all
clear up
rng default
```

Input Data

Loading Data Form Xlsx Data Sheet

```
Loaddata = 'datafile.xlsx';
Traininputdata = 'Sheet1';
Trdatatarget = 'Sheet2';
day_aheat_Data = 'Sheet3';
input = xlsread(Loaddata,Traininputdata,'A1:Z10000');
target = xlsread(Loaddata,Trdatatarget,'A1:Z10000');
Actual_load = xlsread(Loaddata,day_aheat_Data,'A1:Z10000');
inputs=input';
targets=target';
n_input=length(inputs(:,1));
n_output=length(targets(:,1));
hiddenlayersize=45;
net=feedforwardnet(hiddenlayersize,'trainlm');
net=configure(net,inputs,targets);
kk=(n_input*hiddenlayersize)+(hiddenlayersize+hiddenlayersize)+hiddenlayersize+n_output;
```

PSO Parameter Initialization

```
C1=1.49445;           % Personal Learning Coefficient
C2=2.49445;           % Global Learning Coefficient
MaxVel=0.5;           % Maximum Velocity
MinVel=-0.5;          % Minimum Velocity
npop=40;               % Population Size (Swarm Size)
for q=1:kk
MinX(1,q)=-1;
MaxX(1,q)=1;
end
```

PSO Population Initialization

```
for z=1:npop
for q=1:kk
x(z,q)=MinX(1,q)+rand*(MaxX(1,q)-MinX(1,q));
```

```

end
end

NRun=1;
for Run=1:NRun

```

Problem Definition

```

objfun=@(x) myfunc(x,hiddenlayersize,n_input,n_output,net,inputs,targets); % Cost Function
vel=0.1*x; % velocity
for z=1:npop
    fn(z,1)=objfun(x(z,:));
end
[fminx,index]=min(fn);
pbest=x; % particles best position
gbest=x(index,:); % global best position
It=1; MaxIt=50; Tol=1; % Maximum Number of Iterations
while It<=MaxIt && Tol>1e-14

w_max=0.9;
w_min=0.4;
w=0.1+rand*0.4; % Randon weight
initialization

```

PSO Velocity Update

```

for z=1:npop
for q=1:kk
    vel(z,q)= w*vel(z,q)+c1*rand*(pbest(z,q)-x(z,q))+c2*rand*(gbest(1,q)-x(z,q));
% velocity Limit check
if vel(z,q)>Maxvel
    vel(z,q)=Maxvel;
elseif vel(z,q)<Minvel
    vel(z,q)=Minvel;
    vel;
end
end

```

Particles Position Update

```

for z=1:npop
for q=1:kk
    x(z,q)=x(z,q)+vel(z,q);
end
end

```

Check for any Correction Errors

```

for z=1:npop
for q=1:kk

```



```

if x(z,q)>MaxX(q)
    x(z,q)=MaxX(q);
elseif x(z,q)<MinX(q)
    x(z,q)=MinX(q);
end
end
end
end
end

```

Evaluating the Particles Fitness

```

for z=1:npop
    f(z,1)=objfun(x(z,:));
end
% Update Particles best position and their Fitness Function
for z=1:npop
    if f(z,1)<fn(z,1)
        pbest(z,:)=x(z,:);
        fn(z,1)=f(z,1);
    end
end
[Ofgbest,index]=min(fn);           % Sorting the Best Particles
ffmin(It,Run)=Ofgbest;           % Stored
ffIt(Run)=It;                   % Stored the Iteration
% Update the global swarm and best-fit particle
if Ofgbest<fminX
    gbest=pbest(index,:);
    fminX=Ofgbest;
end
% Calculate the limit
if It>20;
    Tol=abs(ffmin(It-20,Run)-fminX);
end

```

Display Iteration Result

```

if It==1
    disp(sprintf('Iteration      Evaluate      ObjFunc.Va'));
end
disp(sprintf('Iteration %6g Evaluate%6g Best Cost   %8.8f',It,index,fminX));
It=It+1;
perf=perform(net,f,targets);
end

```

Loop End

```

xo=gbest;
Eval=objfun(xo);
Xbest(Run,:)=xo;

```

```

Ybest(Run,1)=objfun(xo);
disp(sprintf('+++++'));
disp(sprintf('Iteration      Evalute      ObjFunc.Va'));
disp(sprintf('Maxrun%4g  Eva1%9f  Best Cost      %12g',Run,Eva1,Ybest(Run,1)))

end
toc

```

Final Neural Network Model

```

disp('Final nn model is Tr.net')
Tr.net = feedforwardnet(hiddenlayersize,'trainlm');
Tr.net.layers{1}.transferFcn = 'purelin';
Tr.net.layers{2}.transferFcn = 'tansig';
Tr.net=configure(Tr.net,inputs,targets);
[a b]=min(Ybest);
xo=Xbest(b,:);
k=0;
for z=1:hiddenlayersize
for q=1:n_input
k=k+1;
xi(z,q)=xo(k);
end
end
for z=1:hiddenlayersize
k=k+1;
x1(z)=xo(k);
xb1(z,1)=xo(k+hiddenlayersize);
end
for z=1:n_output
k=k+1;
xb2(z,1)=xo(k);
end
Tr.net.iw{1,1}=xi;
Tr.net.lw{2,1}=x1;
Tr.net.b{1,1}=xb1;
Tr.net.b{2,1}=xb2;

```

Calculation of MSE, RMSE, and test_output result

```

Forecast_load = Tr.net(Actual_load');
err=sum((Tr.net(inputs)-targets).^2)/length(Tr.net(inputs));
rmse=sqrt(err);
mape1=sum(abs((inputs-targets)/inputs)/length(inputs))*100;
MSE=sum(((Actual_load)-Forecast_load).^2)/length((Actual_load))
RMSE=sqrt(MSE)
MAPE=sum(abs(((Actual_load')-(Forecast_load))/Actual_load'))/length(Actual_load))*100
APE=abs(((Actual_load')-(Forecast_load))/Actual_load')*100
R=corr2(Forecast_load,Actual_load')

```

Visualization
Results Plot

```
hold all
plot(Actual_load, 'g--') % plot in red, circles connected with lines
plot(Forecast_load, 'b-') % plot in green, circles connected with lines
plotregression(targets,Tr.net(inputs))
legend('Actual', 'Forecast') % legend text
xlabel('Time (Hours)')
ylabel('Actual & Forecast Load (kW)')
day=clock;
disp(datestr(datenum(day(1),day(2),day(3),day(4),day(5),day(6)),0))
```

[Published with MATLAB® R2016a](#)

APPENDIX B

GA CODE

```
clc
tic
close all
clear up
rng default
```

Input data

Loading data from xls data sheet

```
Loaddata = 'datafile.xlsx';
Trdatainput = 'Sheet1';
Trdatatarget = 'Sheet2';
day_aheat_Data = 'Sheet3';
input = xlsread(Loaddata,Trdatainput,'A1:Z10000');
target = xlsread(Loaddata,Trdatatarget,'A1:Z10000');
Actual_load = xlsread(Loaddata,day_aheat_Data,'A1:Z10000');
inputs=input';
targets=target';
n_input=length(inputs(:,1));
n_output=length(targets(:,1));
hiddenlayersize=30;
net=feedforwardnet(hiddenlayersize,'trainlm');
net=configure(net,inputs,targets);
kk=(n_input*hiddenlayersize)+(hiddenlayersize+hiddenlayersize)+hiddenlayersize+n_output;
```

GA Parameter Initialization

```
beta=8; % SELECTION PRESSURE
npop=30; % POPULATION SIZE
mp = 0.01; % MUTATION PERCENTAGE
cp = 0.8; % CROSSOVER PERCENTAGE
nm = round(npop * mp); % NUMBER OF MUTANTS
nc = round(npop * cp); % NUMBER OF OFFSPRINGS
mu=0.1; % MUTATION RATE
Keep = 10; % HOW MANY OF THE BEST INDIVIDUALS TO KEEP FROM GENERATION TO THE NEXT
nVar=10;
It=1;
MaxIt=20;
Npso=28;
for q=1:kk
    MinX(1,q)=-1;
    MaxX(1,q)=1;
end
```

Population Size (Swarm Size)

```
for z=1:npop
for q=1:kk
R0(z,q)=MinX(1,q)+rand*(MaxX(1,q)-MinX(1,q));
end
end
NRun=1;
for Run=1:NRun;

objfun=@(x) myfunc(x,hiddenlayersize,n_input,n_output,net,inputs,targets);
x=R0;
for z=1:npop
f0(z,1)=objfun(x(z,:));
end
% SORT POPULATION
[fmin,inder0]=min(f0);
pbest=R0;
gbest=R0(inder0,:);
BestSol=f0(1);
worstCost=max(f0);
```

Main Loop

Calculate Selection Probabilities

```
Pb=exp(-beta*f0/worstCost);
Pb=Pb/sum(Pb);
inder0=find(Pb==min(Pb));
% BEGIN SELECTION/CROSSOVER LOOP
for t=Keep+1:2:npop
% SELECT TWO PARENTS TO MATE AND CREATE TWO OFFSPRINGS ROULETTE SELECTION
mate=[];
for selParents=1:2
Random_Cost=rand*sum(Pb);
Select_Cost=Pb(1);
Sel_index=1;
while Select_Cost<Random_Cost
Sel_index=Sel_index+1;
if Sel_index>=npop
break;
end
Select_Cost=Select_Cost+Pb(Sel_index);
end
mate=[mate Sel_index];
end
xNew(1,:)=x(mate(1),:);
xNew(2,:)=x(mate(2),:);
% Crossover
if cp> rand
Xover_Pt1 = ceil(rand * npop);
Xover_Pt2 = ceil(rand * npop);
```

```

if Xover_Pt1 > Xover_Pt2
    temp = Xover_Pt2;
    Xover_Pt2 = Xover_Pt1;
    Xover_Pt1 = temp;
end

    offs(t-keep, :) = [xNew(1, 1:Xover_Pt1) xNew(2, Xover_Pt1+1:Xover_Pt2)
xNew(1, Xover_Pt2+1:npop)];
    offs(t-keep+1, :) = [xNew(2, 1:Xover_Pt1) xNew(1, Xover_Pt1+1:Xover_Pt2)
xNew(2, Xover_Pt2+1:npop)];
else
    offs(t-keep, :) = xNew(1,z);
    offs(t-keep+1, :) = xNew(2,z);
end

% UNIFORM CROSSOVER
for z = 1 : npop
if cp > rand
    offs(t-keep, z) = xNew(1, z);
    offs(t-keep+1, z) = xNew(2, z);
else
    offs(t-keep, z) = xNew(2, z);
    offs(t-keep+1, z) = xNew(1, z);
end

end% END SELECTION/CROSSOVER LOOP
% REPLACE THE NON-ELITE POPULATION MEMBERS WITH NEW OFFSPRINGS
for t=keep+1: 2 : npop
    offs(t)= offs(t-keep);
    offs(t+1)= offs(t-keep+1);
end
% MUTATION
for t = keep + 1: npop% Don't allow the elites to be mutated
for p = ceil(rand*npop)
if mp > rand
    x(t) = offs(p);
end
end
end
for t=keep + 1:nc
for q=1:npop
    offs(offs(t)<MinX(q))=MinX(q);
    offs(offs(t)>MaxX(q))=MaxX(q);
end
end
end
    domin=(1-It/MaxIt)*(npop-Npso)+Npso;
    selT=randperm(round(domin));
    s1=selT(1:Npso);
    tol=1; % Maximum Number of Iterations
while It<MaxIt && tol>1e-14

```

Evaluating Fitness

```
for z=1:Npso
    x(z)=offs(s1(z));
    fn(z,1)=objfun(x(z,:));
end
[fmin, index]=min(fn);
pbest=x;% Initial Best
gbest=x(index, :); % Initial gbest
for z= 1:nc
if fn(z,1)<f0(z,1)
    f0(z,1)=fn(z,1);
    pbest(z,:)=x(z,:);
end
end
% UPDATING pbest AND FITNESS
[ofgbest, index]=min(fn);          % Sort out the best particle
gfmin(It,Run)=ofgbest;            % Stored
ffIt(Run)=It;                     % Stored the iteration
% Update the global and best-fit particle
if ofgbest<fmin
    gbest=pbest(index, :);
    fmin=ofgbest;
end
% CALCULATE THE TOLERANCE
if It>20;
    tol=abs(gfmin(It-20,Run)-fmin);
end
```

Display Iterations Result

```
if It==1
    disp(sprintf('Iteration      Evalute      ObjFunc.Va'));
end
disp(sprintf('Iteration %6g  Evalute%6g  Best Cost    %8.8f',It,index,fmin));
It=It+1;
perf = perform(net,fn,targets);
end
```

Algorithm End

```
xo=gbest;
Fval=objfun(xo);
Xbest(Run,:)=xo;
Ybest(Run,1)=objfun(xo);
disp(sprintf('+++++'));
disp(sprintf('Iteration      Evalute      ObjFunc.Va'));
disp(sprintf('Maxrun%4g  Eval%9f  Best Cost    %12g',Run,Fval,Ybest(Run,1)))
```

```
end
toc
```

Final Neural Network Model

```
disp('Final nn model is Tr.net')
Tr.net = feedforwardnet(hiddenlayersize,'trainlm');
Tr.net.layers{2}.transferFcn = 'tansig';
Tr.net=configure(Tr.net,inputs,targets);
[a b]=min(Ybest);
xo=Xbest(b,:);
t=0;
for z=1:hiddenlayersize
for q=1:n_input
    t=t+1;
    xi(z,q)=xo(t);
end
end
for z=1:hiddenlayersize
    t=t+1;
    x1(z)=xo(t);
    xb1(z,1)=xo(t+hiddenlayersize);
end
for z=1:n_output
    t=t+1;
    xb2(z,1)=xo(t);
end
Tr.net.iw{1,1}=xi;
Tr.net.lw{2,1}=x1;
Tr.net.b{1,1}=xb1;
Tr.net.b{2,1}=xb2;
```

Calculation of MSE, RMSE and Test-Output

```
Forecast_load = Tr.net(Actual_load');
err=sum((Tr.net(inputs)-targets).^2)/length(Tr.net(inputs));
rmse=sqrt(err);
mape1=sum(abs((inputs-targets)/inputs)/length(inputs))*100;
mape1=sum(abs((inputs-targets)/inputs)/length(inputs))*100;
MSE=sum(((Actual_load)-Forecast_load).^2)/length((Actual_load))
RMSE=sqrt(MSE)
MAPE=sum(abs(((Actual_load)-(Forecast_load))/Actual_load))/length(Actual_load))*100
APE=abs(((Actual_load)-(Forecast_load))/Actual_load)*100
R=corr2(Forecast_load,Actual_load');
performance1 = perform(Tr.net,targets,inputs);
performance2 = perform(Tr.net,Actual_load,Forecast_load);
```


Visualization
Results Plot

```
figure
hold all
plot(Actual_load, 'g--') % plot in red, circles connected with lines
plot(Forecast_load, 'b-') % plot in green, circles connected with lines
plotregression(targets,Tr.net(inputs))
legend('Actual', 'Forecast') % legend text
xlabel('Time (Hours)')
ylabel('Actual & Forecast Load (kW)')
day=clock;
disp(datestr(datenum(day(1),day(2),day(3),day(4),day(5),day(6)),0))
```

[Published with MATLAB® R2016a](#)

APPENDIX C

HGAPSO CODE

```
clc
tic
close all
clear up
rng default
```

Input Data

Loading Data for Xlsx Data Sheet

```
Loaddata = 'datafile.xlsx';
Trdatainput = 'Sheet1';
Trdatatarget = 'Sheet2';
day_aheat_Data = 'Sheet3';
input = xlsread(Loaddata,Trdatainput,'A1:Z10000');
target = xlsread(Loaddata,Trdatatarget,'A1:Z10000');
Actual_load = xlsread(Loaddata,day_aheat_Data,'A1:Z10000');
inputs=input';
targets=target';
n_input=length(inputs(:,1));
n_outpu=length(targets(:,1));
hiddenlayersize=45;
net=feedforwardnet(hiddenlayersize,'trainlm');
net=configure(net,inputs,targets);
kk=(n_input*hiddenlayersize)+(hiddenlayersize+hiddenlayersize)+hiddenlayersize+n_outpu;
```

GA and PSO Parameter Initialization

```
beta=8; % Selection Pressure
npop=50; % Population Size
mp = 0.01; % Mutation Percentage
cp = 0.8; % Crossover Percentage
nm = round(npop * mp); % Number of Mutants
nc = round(npop *cp); % Number of Offsprings (also Parnets)
mu=0.1; % Mutation Rate
Keep = 2; % How many of the best individuals to keep from one generation to the
next
nVar=10;
It=1;
MaxIt=50;
Npso=28;
C1=1.49445; % Personal learning coefficient
C2=2.5; % Global learning coefficient
MaxVel=0.5; % Maximum Velocity
MinVel=-0.5; %Minimum Velocity
To1=1; % Maximum Number of Iterations
w_max=0.9;
```

```
w_min=0.4;
for q=1:kk
    MinX(1,q)=-1;
    MaxX(1,q)=1;
end
```

GA Population Initialization

```
for z=1:npop
for q=1:kk
    R0(z,q)=MinX(1,q)+rand*(MaxX(1,q)-MinX(1,q));
end
end
```

Problem Definition

```
NRun=1;
for Run=1:NRun;
    objfun=@(x) myfunc(x,hiddenlayersize,n_input,n_outpu,net,inputs,targets);
    x=R0;
for z=1:npop
    f0(z,1)=objfun(x(z,:));
end
% Sort Population
[fmin0, inder0]=min(f0);
pbest=R0;
gbest=R0(inder0,:);
BestSol=f0(1);
worstCost=max(f0);
```

Main Loop

Calculate Selection Probabilities

```
p = [];
Pb=exp(-beta*f0/worstCost);
Pb=Pb/sum(Pb);
inder0=find(Pb==min(Pb));
% Begin Selection/Crossover Loop

for t=Keep+1:2:npop
% Select Two parents to mate and Create two offspring-roulette wheel selection
    mate=[];
for selParents=1:2
    Random_Cost=rand*sum(Pb);
    Select_Cost=Pb(1);
    Sel_index=1;
while Select_Cost<Random_Cost
        Sel_index=Sel_index+1;
```

```

if Sel_index>=npop
break;
end

    Select_Cost=Select_Cost+Pb(Sel_index);
end

    mate=[mate Sel_index];
end
    xNew(1,:)=x(mate(1),:);
    xNew(2,:)=x(mate(2),:);
% Crossover

if cp> rand

    Xover_Pt1 = ceil(rand * npop);
    Xover_Pt2 = ceil(rand * npop);
if Xover_Pt1 > Xover_Pt2
    temp = Xover_Pt2;
    Xover_Pt2 = Xover_Pt1;
    Xover_Pt1 = temp;
end

    offs(t-Keep, :) = [xNew(1, 1:Xover_Pt1) xNew(2, Xover_Pt1+1:Xover_Pt2)
xNew(1, Xover_Pt2+1:npop)];
    offs(t-Keep+1, :) = [xNew(2, 1:Xover_Pt1) xNew(1, Xover_Pt1+1:Xover_Pt2)
xNew(2, Xover_Pt2+1:npop)];
else
    offs(t-Keep, :) = xNew(1,z);
    offs(t-Keep+1, :) = xNew(2,z);
end

% uniform crossover
for z = 1 : npop
if cp > rand

    offs(t-Keep, z) = xNew(1, z);
    offs(t-Keep+1, z) = xNew(2, z);

else

    offs(t-Keep, z) = xNew(2, z);
    offs(t-Keep+1, z) = xNew(1, z);

end

end% End Crossover Loop
% Replace the non-elite population members with new offsprings
for t=Keep+1: 2 : npop
    offs(t)= offs(t-Keep);
    offs(t+1)= offs(t-Keep+1);
end
% Mutation
for t = Keep + 1: npop% Don't allow the elites to be mutated
for p = ceil(rand*npop)
if mp > rand

    x(t) = offs(p);

end
end
end
for t=Keep + 1:nc

```

```

for q=1:npop
    offs(offs(t)<MinX(q))=MinX(q);
    offs(offs(t)>MaxX(q))=MaxX(q);
end
end
end

domin=(1-It/MaxIt)*(npop-Npso)+Npso;
selT=randperm(round(domin));
sn=selT(1:Npso);

```

Evaluating The Fitness

```

for z=1:Npso
    x(z)=offs(sn(z));
    fn(z,1)=objfun(x(z,:));
end
for z=1:Npso
    if fn(z,1)<f0(z,1)
        f0(z,1)=fn(z,1);
    end
end
[fmin0, index]=min(fn);
pbest=x;
gbest=x(index, :);

```

PSO Population Initialization

```

vel=0.1*x;
w=0.1+rand*0.4;
while It<MaxIt && Tol>1e-14

```

Particles Velocity Update

```

for z=1:Npso
for q=1:kk-1
    vel(z,q)= w*vel(z,q)+C1*rand*(pbest(z,q)-x(z,q))+C2*rand*(gbest(1,q)-x(z,q));
% velocity Limit
if vel(z,q)>Maxvel
    vel(z,q)=Maxvel;
elseif vel(z,q)<Minvel
    vel(z,q)=Minvel;
end
end
end

```

Particles Position Update

```

for z=1:Npso
for q=1:kk-1
    x(z,q)=x(z,q)+vel(z,q);

```

```

end
end
% CHECK FOR ANY CORRECTION ERRORS
for z=1:Npso
for q=1:kk-1
if x(z,:)>MaxX(q)
                x(z,:)=MaxX(q);
elseif x(z,:)<MinX(q)
                x(z,:)=MinX(q);
end
end
end
end
end

```

Evaluation Of The Fitness

```

for z=1:Npso
    fn(z,1)=objfun(x(z,:));
end
for z=1:Npso
if fn(z,1)<f0(z,1)
    f0(z,1)=fn(z,1);
    pbest(z,:)=x(z,:);
end
end
% Updating pbest and Fitness
[Ofgbest, index]=min(fn);           % sort the best particle
ffmin(It,Run)=Ofgbest;             % stored
ffIt(Run)=It;                     % stored the iteration
% Update the global swarm and best-fit particle
if Ofgbest<fmin0
    gbest=pbest(index,:);
    fmin0=Ofgbest;
end

% Calculate the limit
if It>50;
    Tol=abs(ffmin(It-20,Run)-fmin0);
end
end

```

Display Iteration Result

```

if It==1
    disp(sprintf('Iteration      Evalute      ObjFunc.Va'));
end
disp(sprintf('Iteration %6g Evalute%6g Best Cost   %8.8f',It,index,fmin0));
It=It+1;
perf = perform(net,fn,targets);
end

```

Algorithm End

```
xo=gbest;
Fval=objfun(xo);
Xbest(Run,:)=xo;
Ybest(Run,1)=objfun(xo);
disp(sprintf('++++'+));
disp(sprintf('Iteration      Evaluate      ObjFunc.va'));
disp(sprintf('Maxrun%4g Eval%9f Best Cost      %12g',Run,Fval,Ybest(Run,1)))
end
toc
```

Final Neural Network Model

```
disp('Final nn model is Tr.net')
Tr.net = feedforwardnet(hiddenlayersize,'trainlm');
Tr.net.layers{1}.transferFcn = 'purelin';
Tr.net.layers{2}.transferFcn = 'tansig';
Tr.net=configure(Tr.net,inputs,targets);
[a b]=min(Ybest);
xo=Xbest(b,:);
t=0;
for z=1:hiddenlayersize
for q=1:n_input
t=t+1;
xi(z,q)=xo(t);
end
end
for z=1:hiddenlayersize
t=t+1;
x1(z)=xo(t);
xb1(z,1)=xo(t+hiddenlayersize);
end
for z=1:n_outpu
t=t+1;
xb2(z,1)=xo(t);
end
Tr.net.iw{1,1}=xi;
Tr.net.lw{2,1}=x1;
Tr.net.b{1,1}=xb1;
Tr.net.b{2,1}=xb2;
```

Calculation of MSE, RMSE, and Test-Output Result

```
Forecast_load = Tr.net(Actual_load');
err=sum((Tr.net(inputs)-targets).^2)/length(Tr.net(inputs));
rmse=sqrt(err);
mape1=sum(abs((inputs-targets)/inputs)/length(inputs))*100;
mape1=sum(abs((inputs-targets)/inputs)/length(inputs))*100;
MSE=sum(((Actual_load)-Forecast_load).^2)/length((Actual_load))
```

```

RMSE=sqrt(MSE)
MAPE=sum(abs(((Actual_load)-(Forecast_load))/Actual_load))/length(Actual_load)*100
APE=abs(((Actual_load)-(Forecast_load))/Actual_load)*100
R=corr2(Forecast_load,Actual_load)
performance1 = perform(Tr.net,targets,inputs);
performance2 = perform(Tr.net,Actual_load,Forecast_load);

```

Visualization

Results Plot

```

figure
hold all
plot(Actual_load, 'g--') % plot in red, circles connected with lines
plot(Forecast_load, 'b--') % plot in green, circles connected with lines
legend('Actual', 'Forecast') % legend text
xlabel('Time (Hours)')
ylabel('Actual & Forecast Load (kW)')
plotregression(targets,Tr.net(inputs))
day=clock;
disp(datestr(datetime(day(1),day(2),day(3),day(4),day(5),day(6)),0))

```


APPENDIX D

Objective Function

```
function [f] = myfunc(x,hiddenLayerSize,Input_n,Output_n,net,Inputs,Targets);
%Function calculates 24-hour ahead electrical load forecast
% Solve an Input-Output Fitting problem with a Neural Network
% This script assumes these variables are defined:
k=0;
for i=1:hiddenLayerSize
for j=1:Input_n
k=k+1;
xi(i,j)=x(k);
end
end
for i=1:hiddenLayerSize
k=k+1;
xl(i)=x(k);
xb1(i,1)=x(k+hiddenLayerSize);
end
for i=1:Output_n
k=k+1;
xb2(i,1)=x(k);
end
net.iw{1,1}=xi;
net.lw{2,1}=xl;
net.b{1,1}=xb1;
net.b{2,1}=xb2;
% Training function
net.trainFcn = 'trainlm'; % Levenberg-Marquardt
net.layers{1}.transferFcn = 'purelin';
net.layers{2}.transferFcn = 'tansig';
% Calculating the mean squared error
net.performFcn = 'mse'; % Mean absolute error
f=sum((net(Inputs)-Targets).^2)/length(net(Inputs));
```

[*Published with MATLAB® R2016a*](#)