

Artificial Neural Network and Particle Swarm Optimization for Medium Term Electrical Load Forecasting in a Smart Campus

Isaac A. Samuel^{1*}, Temitope M. Adeyemi-Kayode², Ayobami A. Olajube³,
Shorinwa T. Oluwasijibomi⁴, Adekitan I. Aderibigbe⁵

^{1,2,3,4}*Department of Electrical and Information Engineering, Covenant University, Ota.*

⁵*Department of Electrical Engineering and Information Technology, Technische Universität Ilmenau, Ilmenau, Germany.*

¹ORCID: 0000-0001-8010-4835, ³ORCID: 0000-0001-8704-445X

Abstract

Energy demand has continued to increase rapidly not exempting Covenant University. As the university continues to witness infrastructural expansion and population increase, it has become a necessity for energy consumption to be predicted. Hence, this research work developed a medium-term load forecasting system to solve this problem and ensure an efficient electricity supply from the power system operators of Covenant University. The forecast was carried out on real-time monthly load data collected from the university community power plant between 2015 and 2018, using the Artificial Neural Network (ANN) model. A medium-term load forecast was evaluated based on three different ANN algorithms. The FeedForwardNet, Cascadeforwardnet and Fitnet are tested against three (3) different learning algorithms namely Levenberg Marquardt, Bayesian regularization and BFGS quasi-Newton backpropagation with a particle swarm optimizer. And the network performance was obtained using Normalized Root Mean Square Error (nRMSE %). The result revealed an nRMSE of 0.0634%, a correlation coefficient, r , of 0.9082 and the fastest computation speed of 171.789 seconds. Hence, this study provides a point of reference for other related studies and future energy forecast improvement in the study location.

Keywords: Artificial Neural Network, Particle Swarm Optimization, Load Forecasting, Long Term, Medium Term, Short Term, nRMSE.

I. INTRODUCTION

As part of its functions, the electric utility company must be able to accurately predict the load requirement at every instance of time. Load forecasting is a very significant tool in operation and planning. In literature, Load forecasting can be divided into two broad categories: Based on the duration of the forecast and based on the methodology of evaluating the forecast. With regards to the duration of forecasting, load forecast can be

divided into three (3) broad categories. Short term, Medium-term and Long-term Load forecast. Short term load forecasting considers prediction ranging from a few seconds, minutes to a few hours, Medium-term load forecast deals with forecasting load from weeks to several months while Long term forecast is used for load forecast no less than a year [1][2][3][4].

In terms of application, Short term load forecasts are very important for daily operations of the energy plant, evaluation of net interchange, unit commitment, scheduling and other system security analysis. Medium-term load forecast is used majorly for fuel scheduling, maintenance planning etc. Long term load forecast is used for effective management of grid and expansion planning, future investments and revenue analysis. Based on the methodology, there are two broad approaches-Linear or Non-Linear. Linear models consist of Time series and Dynamic programming methods while non-linear models consist of methods such as Support vector machine, Markov chain, Artificial Neural Networks and Stochastic distribution shown in Figure 1.

In [5] Medium and Long-term Load Forecast Based on Particle Swarm Optimization, Least Square method and Support Vector Regression (PSO-LS-SVR) were discussed. The dataset used here are Annual total consumption of electricity 1990-2009, Electricity consumption, GDP, Population, Average electricity tariff, Average temperature in winter and Average temperature in summer. They presented a model based on the Least Square-Support Vector Regression and PS optimization.

Also in [6], a medium load forecast for Covenant University was conducted using linear, compound-growth and cubic regression methods. The best method from the result was the linear model with a 0.5795 and 41.34 for Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) respectively. Similarly, an ANN base short-term electricity load forecast was performed on the 133/33KV transmission

substation in Nigeria. It proved to be a good method with an R-Value of 0.988 and Mean Square Error (MSE) of 0.27 [7].

According to [8], Grey model [GM(1,1)] alone does not apply to long term load forecasting because of parameter 'a' in the model unless it is modified by including a vector θ which is incorporated into the model; GM(1,1, θ) – the residual error model and solved using the Particle Swarm Optimization (PSO). A detail explanation of GM(1,1, θ) model, its

optimization and solution with PSO are provided in this paper. The authors discovered that GM(1,1, θ) performed better than

GM(1,1). They also concluded that the prediction data is always closer to the real value regardless of whether residual error data average or after-test variance ration C is used. The authors also compared the two model to the textual method. The textual method outperformed the two previous methods.

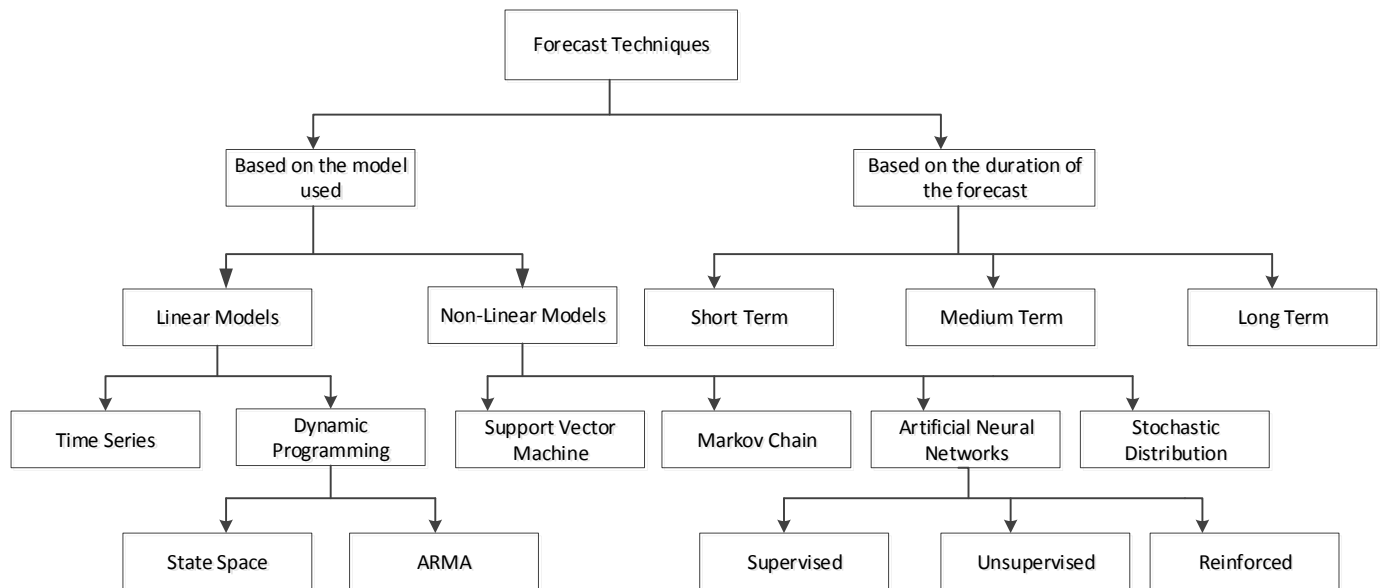


Fig. 1. Classification of Load Forecasting Techniques

In [9], the paper discussed the groups of methods used in long term load forecasting. The major methods are static and dynamic state estimation techniques. It presents a new way of estimating parameters for a long-term load forecast using Particle Swarm Optimization (PSO). The mathematical formulation here simulates how electric load demand varies with other variables such as time, meteorological conditions etc. An appropriate model is selected for the forecast. The authors reported that many of the models used at that time revolved around multiple regression, statistical methods and general exponential smoothing. They used models developed in reference [10] and [11]. The authors compared the developed PSO models to both linear and quadratic equations. The PSO parameters were detailed in the paper. The results from comparing the actual load forecast with the Linear and Quadratic PSO methods as well as the Least Estimation Square method were presented. From the results, the quadratic equation optimized with Particle Swarm (PS) performed best. For the PSO parameters Population =10, Stop Criterion = 1000 Iterations, Velocity (min) = 0; Velocity (max) = 2, Acceleration

constants $C_1 = 3$ and $C_2 = 3$; Inertia weights $W_{max} = 0.9$ and $W_{min} = 0$.

[12] worked on parameter optimization of nonlinear grey Bernoulli model using particle swarm optimization. There is a modification of the Grey model called the Nonlinear grey Bernoulli model (NGBM). This is simply the infusion of the Grey model with the Bernoulli differential equation. Unlike what was stated in, the authors here stated that the Grey model is suitable for Long term Load forecasting. In this work, the parameters power index 'r' and production coefficient 'p' present in the NGBM model was optimized using the Particle Swarm Optimization. The paper contains a thorough breakdown of the NGBM model as well as the infusion of the PSO. The result of this framework reveals that the NGBM model with PSO infusion has the least MAPE values when compared with other models. The authors also used the model to predict 3 future years.

II. MATERIALS AND METHOD

II. I Artificial Neural Network (ANN)

The ANN is analogous to the biological neurons. The perfect example of the biological neuron is the human brain [13]. It is the most complex and powerful structure involving neurons known today. ANNs are composed of highly interconnected elements called neurons. The neurons are connected by links denoted by weights which transfer signals from one neuron to another. ANN is defined as an array of parallel combinations of simple processing units which are highly capable of modifying their parameters via a learning process in response to their environment to capture information [14][15].

ANN has an advantage of not being explicitly programmed but only being trained hence, allowing it to capture and recognize variations between variables which it was not trained for [16]. It is also highly resistant to noise.

However, depending on the complexity of the network model it might need higher computational capacity for proper training and also the training time of the network increases with its size [17]. Figure 2 shows a sample of ANN architecture.

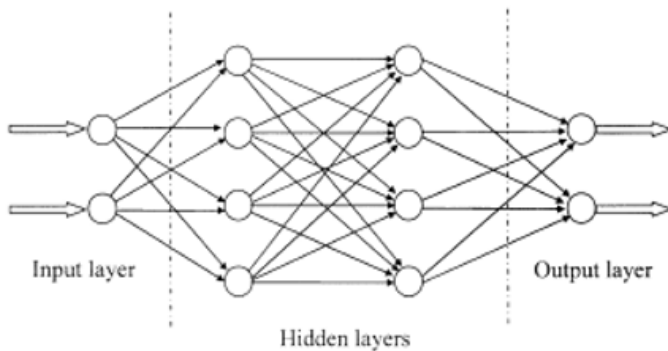


Fig. 2 ANN Architecture

II.II Particle Swarm Optimization

This technique is modelled after the behaviour of fishes and birds in a swarm. The movements are structured such that the fishes and birds do not intercept with others and move towards the destination as well as the centre of the group. The idea of PSO was formulated in the 1990s. The individuals, called 'agents' are characterized by a two-dimensional space based in its position (x and y) and its velocity V_x and V_y . Each agent tries to optimize its movement towards the destination in doing so; it tracks (a) the best value of the objective function which has been achieved so far called 'pbest', (b) the best value of the objective function which the other agents have achieved so far called 'gbest'. The position of the agents is modified by noting its current position, velocity and the distances between the current position with 'pbest' and 'gbest'.

The new position of an agent i , V_i^{k+1} in iteration $k + 1$ (S_i^{k+1}) can be determined from its current (iteration k) position (S_i^k); knowing its velocity at iteration $k + 1$ (V_i^{k+1}).

$$V_i^{k+1} = \omega V_i^k + C_1 rand_1 (pbest_i - S_i^k) + C_2 rand_2 (gbest_i - S_i^k) \quad (1)$$

Where:

ω = weighting factor

C_1 and C_2 = weighting coefficient

$rand_1$ and $rand_2$ = two random numbers between 0 and 1

The first term ωV_i^k , results in agent movement in the same direction as before as a result of exploring new search space. ω is called the diversification coefficient. Usually, it is defined as:

$$\omega = \bar{\omega} - \left(\frac{\bar{\omega} - \underline{\omega}}{iter} \right) iter \quad (2)$$

$\bar{\omega}$ and $\underline{\omega}$ are typically selected to be 0.9 and 0.4 respectively.

The second and third term; $C_1 rand_1 (pbest_i - S_i^k)$ and $C_2 rand_2 (gbest_i - S_i^k)$ result in the so-called intensification C_1 and C_2 may be typically selected as 2.0.

This classical PSO algorithm has been used in many literatures. The PSO parameters used in this study is provided in Table 1.

$$V_{im}^{(k+1)} = \omega V_{im}^{(k)} + C_1 r_1^{(k)} (P_{ipm}^{(k)} - P_{im}^{(k)}) + C_2 r_2^{(k)} (P_{gm}^{(k)} - P_{im}^{(k)}) \quad (3)$$

$$P_{im}^{(k+1)} = P_{im}^{(k)} + V_{im}^{(k+1)} \quad (4)$$

Where:

$V_i = V_{i1}, V_{i2}, \dots, V_{im}, \dots, V_{im}$; velocity of the particles

$P_i = P_{i1}, P_{i2}, \dots, P_{im}, \dots, P_{im}$; position of the particles

P_{ipm} = individual optimum value, best historical position of the current particle

C_1 and C_2 = Learning factors

ω = inertia weights

$r_1^{(k)}$ and $r_2^{(k)}$ = random numbers between 0 and 1

Table 1: PSO parameters utilized in this research

PSO parameters	Values
Population	10
Stop Criterion (Iterations)	1000
Velocity	$V_{max} = 2.0, V_{min} = 0$
Acceleration Constants	$C_1 = 1.5, C_2 = 2.5$
Inertia weights	$W_{max} = 0.9, W_{min} = 0.4$

2. Input the value of the current particle into the NN function and get the output value
3. Calculate the fitness of each particle
4. Update the individual optimal value and group optimal value
5. Determine whether the group optimal value meets the maximum number of iterations. If this condition is reached, it brings an end to the algorithms. If not, continue to step 7.
6. Confirm that the algorithm does not fall into premature convergence. If this happens, skip to step 7
7. Update the values of the velocity and position to generate the next generation particles, repeat step 3-6.

II.II.I Flowchart of the PSO-NN

The Steps taking to evaluate the PSO and ANN algorithm as shown in Figure 3 includes:

1. Initialize PSO-NN

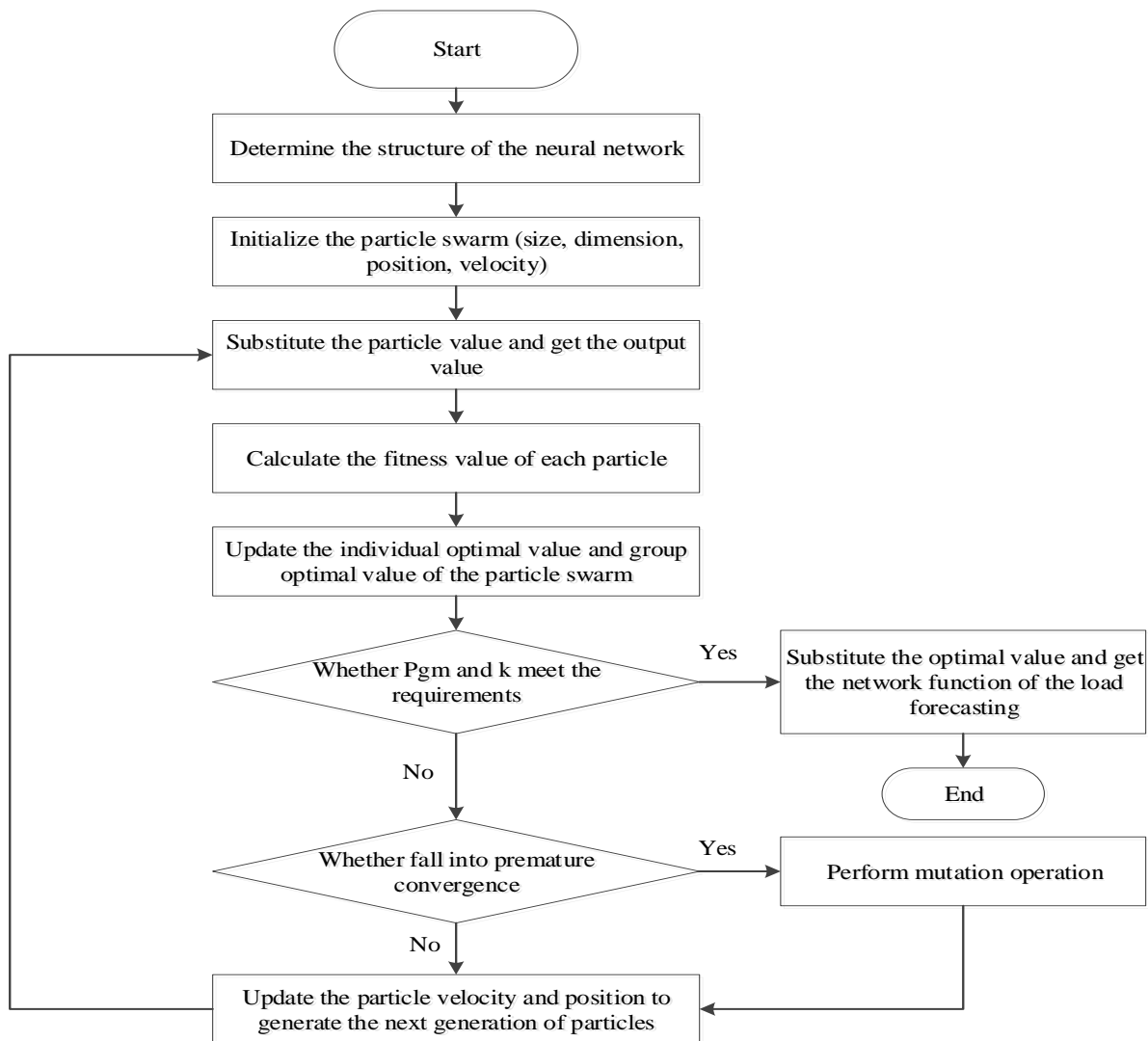


Fig. 3. PSO-ANN flowchart

II.II.II Data Description

The historical data used in this study was obtained from the substation datacenter at Covenant University. The data retrieved includes monthly load demand and temperature of the university community. The data values collected spanned from 2015 to 2018. For better utilization of the data, it is of necessity that it is preprocessed and analyzed for proper understanding.

The load demand dataset contained the temperature of the surrounding environment, as this has been found to have effects on load demand. In 2016, *Hong T* discussed the effect of weather particularly temperature on load forecasting. It was stated that a drop of 1°C in temperature during winter would lead to an additional demand of 1.8 GW in France. Figure 4 -7 provides the line plot for load demanded from 2015 to 2018.

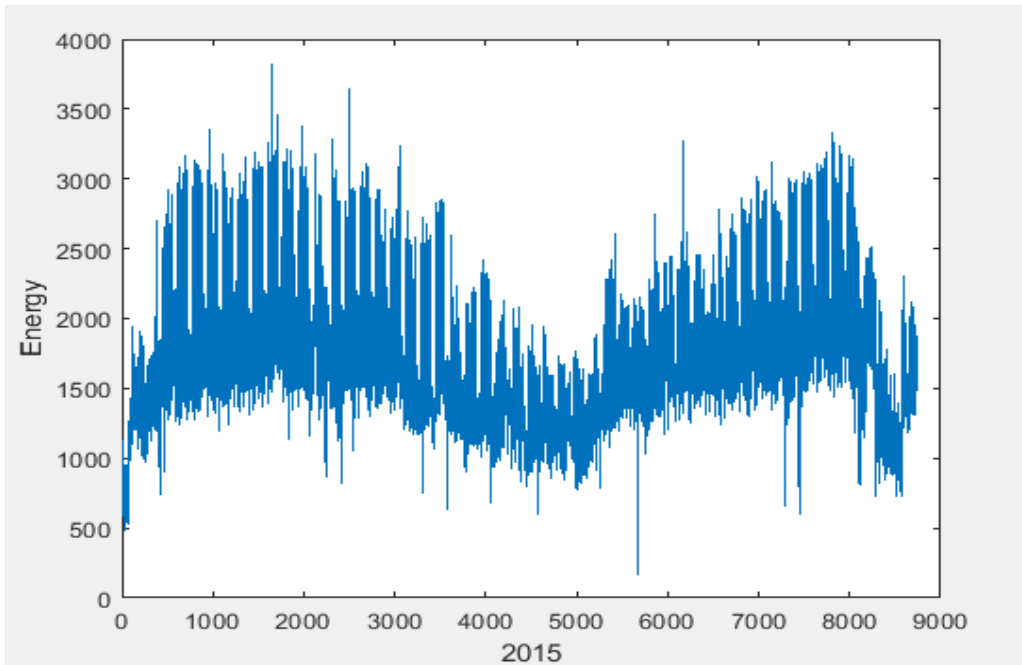


Fig. 4: Line Plot of Load demand for 2015

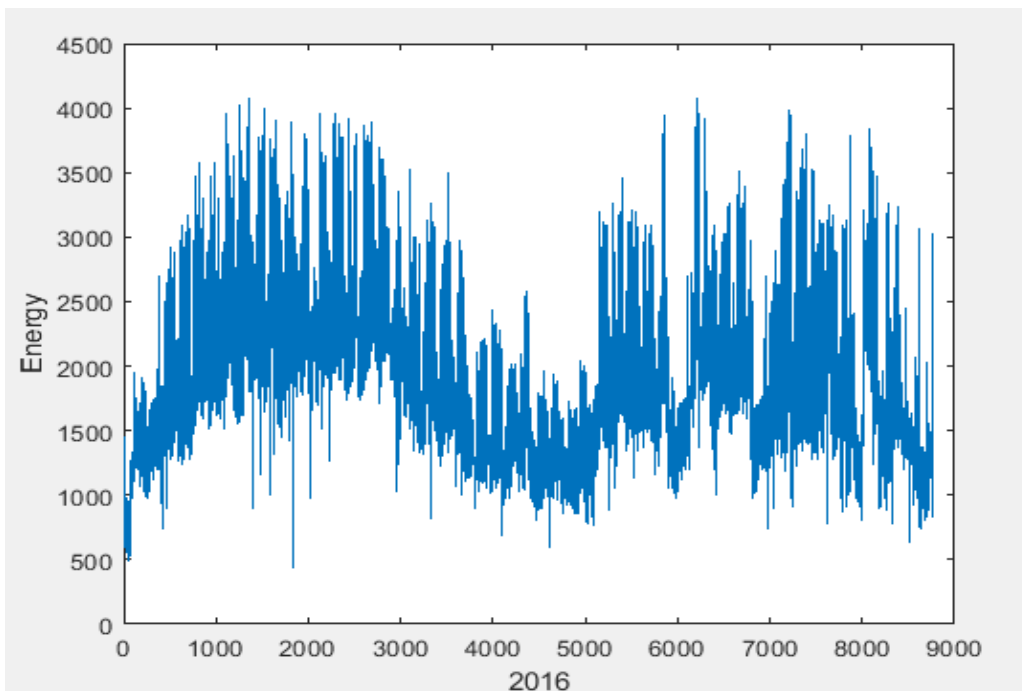


Fig. 5. Line Plot for Load demand in 2016

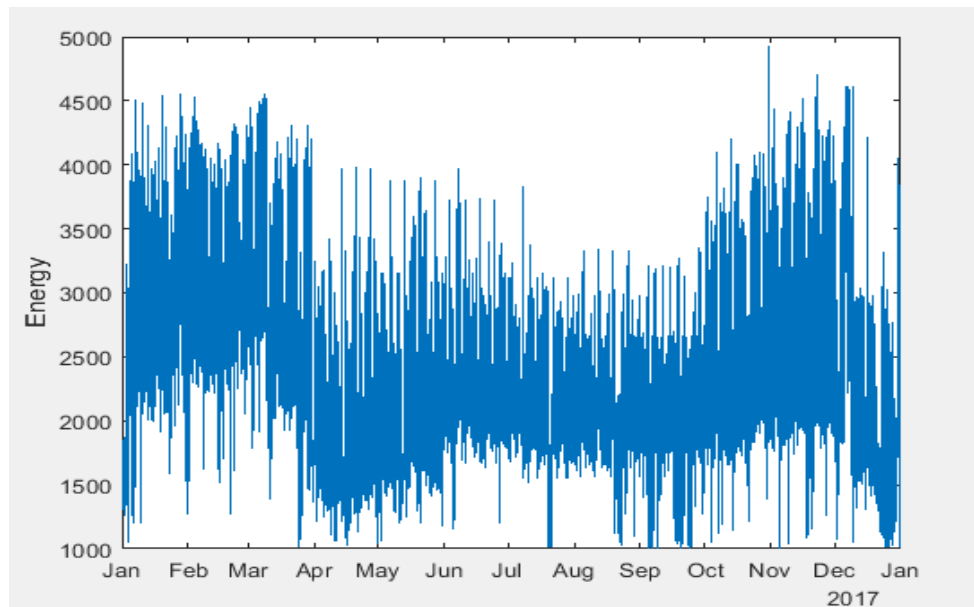


Fig.6. Line Plot for Load Demand in 2017

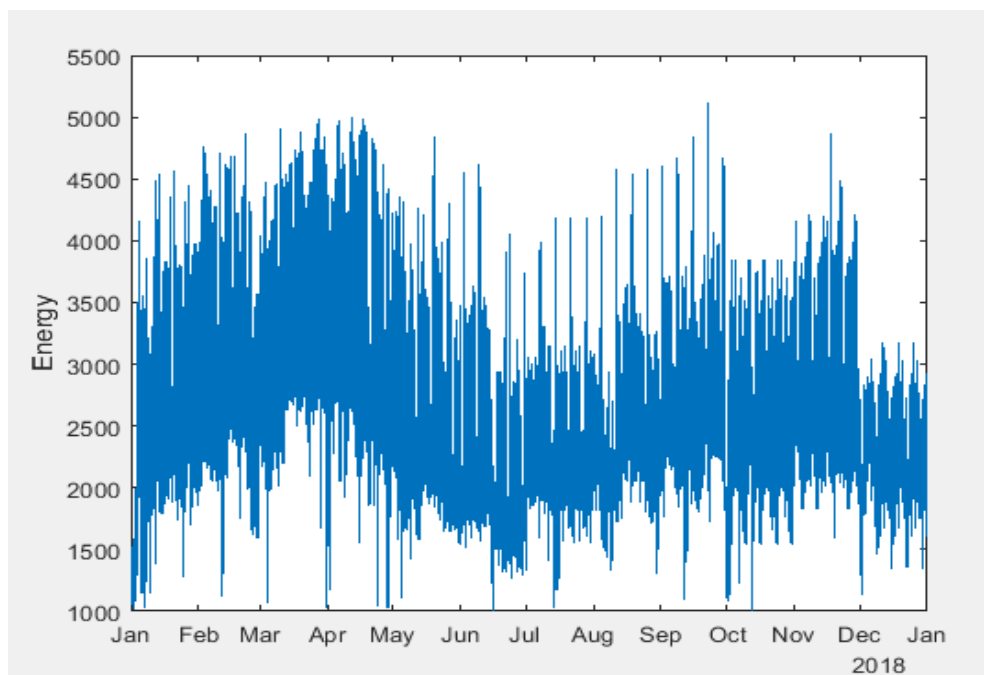


Fig.7. Line Plot for Load Demand for 2018

II.II.III Practical application

The ANN model used for the load prediction was designed with MATLAB software. There are four basic stages in load forecasting: the model design, model training, model validation and forecasting the trained model. The system is modelled to have several input neurons, a couple of hidden layers and an output. Thereafter the model is then trained to perform the required task. The dataset is divided into about 70% for training and the rest for testing and validation. In this training stage, the ideal bias and weight are gotten. Using the validation dataset,

the trained model is then validated and the accuracy of the forecast is determined. After the ANN model has been trained and validated it is then used to make forecasts as regards the input and output comparison.

A medium-term load forecast was evaluated based on three different Artificial neural network algorithms; FeedForwardNet, Cascadeforwardnet and Fitnet are tested against three (3) different learning algorithms namely Levenberg Marquardt, Bayesian regularization and BFGS quasi-Newton backpropagation with a particle swarm

optimizer. The input layer has three neurons representing the three different inputs. The inputs used are Year, Month of the year and External temperature (°C). The months of the year were assigned certain numerical values from 1 to 12. The output consists of maximum energy demanded for each month.

The output layer has only one neuron. Monthly Data from 2015 to 2017 was used to train to network while monthly data from 2018 was used to validate the network. A summary of the ANN parameter set used in this study is provided in Table 2.

Table 2: ANN parameter settings

ANN Layer	No of Neurons	ANN Architecture Settings
Input Layer	3	Year, Month of the year and External temperature (°C).
Output Layer	1	Maximum energy demanded for each month

III. RESULTS AND DISCUSSION

As stated earlier, the results of three different ANN training algorithms were compared against three different learning algorithms to compare the difference in the correlation coefficient (r) as well as the normalized root mean square error (nRMSE %). Equation 5 is the mathematical representation of the normalized root mean square error used for data validation. The nRMSE is based on the maximum hourly power output, $P_{m,h}$

$$nRMSE\% = \sqrt{\frac{\sum_{i=1}^N |P_{m,h} - P_{p,h}|^2}{N \max(P_{m,h})}} \quad (5)$$

Where:

$P_{m,h}$ = average actual power in the hour

$P_{p,h}$ = the prediction provided by one of the forecasting methods

N = Number of hours considered in the evaluation period

In Table 3, the result of the PSO-NN network based on three neural network algorithm FeedForwardNet (FFN), Cascadeforwardnet (CFN), Fitnet (FN) against the Levenberg Marquardt training algorithm. The result revealed that the FeedForwardNet and the Fitnet had similar r and nRMSE values of 0.90822 and 0.0634 respectively. However, the FeedForwardNet had a faster computational speed of 338.42 seconds compared to Fitnet's computational speed of 352.884 seconds. Table 4 provides the comparison of actual and predicted energy demand for 2018 using the three neural network algorithms.

Table 3: Comparative Result for PSO-NN network based on the Levenberg Marquardt Training algorithm

Neural Network Algorithm	Time (Secs)	R	nRMSE %
FeedForwardNet (FFN)	338.42	0.90822	0.0634
Cascadeforwardnet (CFN)	401.658	0.74851	0.5039
Fitnet (FN)	352.884	0.90822	0.0634

Table 4: Comparison of Actual and Predicted energy demand based on the Levenberg Marquardt Training algorithm

Month	Actual (MW)	FFN (MW)	CFN (MW)	FN(MW)
1	4567	3639.5	4990.1	3639.5
2	4863	4962.0	6113.9	4962.0
3	4982	3771.2	5275.7	3771.2
4	5004	4116.6	5642.8	4116.6
5	4848	4327.4	5985.6	4327.4
6	4624	4455.1	6245.3	4455.1
7	4181	4220.4	6473.5	4220.4
8	4578	4196.4	6581.5	4196.4
9	5109	4249.0	6435.3	4249.0
10	3846.2	5296.1	6952.8	5296.1
11	4872	5454.7	6912.7	5454.7
12	3175.2	5181.5	6440.7	5181.5

In Table 5, the result of the PSO-NN network based on three neural network algorithm FeedForwardNet (FFN), Cascadeforwardnet (CFN), Fitnet (FN) against the Bayesian regularization training algorithm. The result revealed that the FeedForwardNet and the Fitnet had similar r and nRMSE values of 0.90822 and 0.0634 respectively. However, the Fitnet had a faster computational speed of 342.891 seconds compared

to FeedForwardNet's computational speed of 373.810 seconds but CascadeForwardNet had the fastest computational speed of 242.57 seconds but had a poor r value and nRMSE of 0.74851 and 0.5039 respectively. Table 6 provides the comparison of actual and predicted energy demand for 2018 using the three neural network algorithms.

Table 5: Comparative Result for PSO-NN network based on Bayesian regularization training algorithm Learning Algorithm = 'trainbr'

Neural Network Algorithm	Time (Secs)	R	nRMSE %
FeedForwardNet	373.810	0.90822	0.0634
cascadeforwardnet	242.57	0.74851	0.5039
Fitnet	342.891	0.90822	0.0634

Table 6: Comparison of Actual and Predicted energy demand based on the Bayesian regularization training algorithm

Month	Actual (MW)	FFN (MW)	CFN (MW)	FN(MW)
1	4567	3639.5	4990.1	3639.5
2	4863	4962.0	6113.9	4962.0
3	4982	3771.2	5275.7	3771.2
4	5004	4116.6	5642.8	4116.6
5	4848	4327.4	5985.6	4327.4
6	4624	4455.1	6245.3	4455.1
7	4181	4220.4	6473.5	4220.4
8	4578	4196.4	6581.5	4196.4
9	5109	4249.0	6435.3	4249.0
10	3846.2	5296.1	6952.8	5296.1
11	4872	5454.7	6912.7	5454.7
12	3175.2	5181.5	6440.7	5181.5

In Table 7, the result of the PSO-NN network based on three neural network algorithm FeedForwardNet (FFN), Cascadeforwardnet (CFN), Fitnet (FN) against the BFGS quasi-Newton backpropagation training algorithm. The result revealed that the FeedForwardNet and the Fitnet had similar r and nRMSE values of 0.90822 and 0.0634 respectively.

However, the FeedForwardNet had a faster computational speed of 171.789 seconds compared to Fitnet's computational speed of 222.543 seconds. Table 8 provides the comparison of actual and predicted energy demand for 2018 using the three neural network algorithms.

Table 7: Comparative Result for PSO-NN network based on BFGS quasi-Newton backpropagation training algorithm Learning Algorithm = 'trainbfg'

Neural Network Algorithm	Time (Secs)	R	nRMSE %
FeedForwardNet	171.789	0.90822	0.0634
cascadeforwardnet	330.365	0.74851	0.5039
fitnet	222.543	0.90822	0.0634

Table 8: Comparison of Actual and Predicted energy demand based on the BFGS quasi-Newton backpropagation training algorithm Learning Algorithm = ‘trainbfg’

Month	Actual (MW)	FFN (MW)	CFN (MW)	FN(MW)
1	4567	3639.5	4990.1	3639.5
2	4863	4962.0	6113.9	4962.0
3	4982	3771.2	5275.7	3771.2
4	5004	4116.6	5642.8	4116.6
5	4848	4327.4	5985.6	4327.4
6	4624	4455.1	6245.3	4455.1
7	4181	4220.4	6473.5	4220.4
8	4578	4196.4	6581.5	4196.4
9	5109	4249.0	6435.3	4249.0
10	3846.2	5296.1	6952.8	5296.1
11	4872	5454.7	6912.7	5454.7
12	3175.2	5181.5	6440.7	5181.5

IV. CONCLUSION

A medium-term load forecast was evaluated based on three different Artificial neural network algorithms; FeedForwardNet, Cascadeforwardnet and Fitnet are tested against three (3) different learning algorithms namely Levenberg Marquardt, Bayesian regularization and BFGS quasi-Newton backpropagation with a particle swarm optimizer.

The dataset used in this study (collected at the Energy data centre of Covenant University, Nigeria) contains historical monthly series of highest energy demanded, highest recorded monthly temperature and year of record for 2015 to 2018. The monthly dataset from 2015 to 2017 was used in training the network while data from 2018 was used in validating the network.

Based on the Neural Network Algorithm, the Feedforward network and the Fit net produced similar R and normalized root mean square error values (nRMSE). The values are 0.90822 and 0.0634% respectively. However, based on computational time, the shortest time recorded is 171.789 seconds using the Feedforward net PSO-NN network based on BFGS quasi-Newton backpropagation training algorithm; followed by the 222.543 seconds using the FitNet PSO-NN also based on the BFGS quasi-Newton backpropagation training algorithm. From these results, it can be deduced that FeedForwardNet and Fitnet Neural Network algorithm produce similar results, however, the BFGS quasi-Newton backpropagation produces the results at a faster computational speed.

ACKNOWLEDGEMENT

The authors of this article hold the management of covenant university in high esteem for sponsoring this research.

REFERENCES

- [1] A. Ahmad, N. Javaid, A. Mateen, M. Awais, and Z. A. Khan, "Short-Term load forecasting in smart grids: An intelligent modular approach," *Energies*, vol. 12, no. 1, pp. 1–21, 2019.
- [2] M. Q. Raza, M. Nadarajah, and C. Ekanayake, "On recent advances in PV output power forecast," *Sol. Energy*, vol. 136, no. September 2019, pp. 125–144, 2016.
- [3] M. D., A. A. Ayeni, and M. A., "Short Term Electric Load Forecasting using Neural Network and Genetic Algorithm," *Int. J. Appl. Inf. Syst.*, vol. 10, no. 4, pp. 22–28, 2016.
- [4] A. A. Adewale, A. I. Adekitan, O. J. Idoko, F. A. Agbetuyi, and I. A. Samuel, "Energy audit and optimal power supply for a commercial building in Nigeria," *Cogent Eng.*, vol. 5, no. 1, pp. 1–18, 2018.
- [5] X. Pan, F. Qian, and X. Jin, "Medium and long-term load forecast based on PSO-LS-SVR," *Adv. Mater. Res.*, vol. 614–615, pp. 1962–1965, 2013.
- [6] Isaac A. Samuel, Felly-Njoku Chihurumanya F., Adewale Adeyinka A. and Ayokunle A. Awelewa "Medium-Term Load Forecasting of Covenant University Using the Regression Analysis Methods" *J. of Energy Technol. and Policy*, Vol.4, No.4, pp. 10-17, 2014. .

- [7] I. A. Samuel, S. Ekundayo, A. Awelewa, T. E. Somefun, and A. Adewale, "Artificial neural network based short-term electricity load forecasting: A case study of a 132/33 kv transmission sub-station," *Int. J. Energy Econ. Policy*, vol. 10, no. 2, pp. 200–205, 2020.
- [8] N. Dongxiao, L. Yanchang, and L. Wei, "Research of residual error-Ant Colony Optimization gray model based on Markov in load forecasting," *Proc. - ISECS Int. Colloq. Comput. Commun. Control. Manag. CCCM 2008*, vol. 1, pp. 438–443, 2008.
- [9] M. R. AlRashidi and K. M. El-Naggar, "Heuristic approach for optimal parameter estimation of electric load forecast model," *AIP Conf. Proc.*, vol. 1159, pp. 49–53, 2009.
- [10] K. M. El-Naggar and K. Al-rumaih, "Algorithm, Optimal Filter Estimator and Least Error Squares Technique : Comparative Study," *Int'l J Electrical, Robot. Electron. Commun. Eng.*, vol. 1, no. 6, pp. 941–945, 2007.
- [11] H. K. Temraz, K. M. El-Nagar, and M. M. A. Salama, "Applications of noniterative least absolute value estimation for forecasting annual peak electric power demand," *Can. J. Electr. Comput. Eng.*, vol. 23, no. 4, pp. 145–146, 1998.
- [12] J. Zhou, R. Fang, Y. Li, Y. Zhang, and B. Peng, "Parameter optimization of nonlinear grey Bernoulli model using particle swarm optimization," *Appl. Math. Comput.*, vol. 207, no. 2, pp. 292–299, 2009.
- [13] G. A. Adepoju, S. O. A. Ogunjuyigbe, and K. O. Alawode, "Application of Neural Network to Load Forecasting in Nigerian Electrical Power System," *Pacific J. Sci. Technol.*, vol. 8, no. 1, pp. 68–72, 2007.
- [14] R. Tadeusiewicz, "Neural networks: A comprehensive foundation," *Control Eng. Pract.*, vol. 3, no. 5, pp. 746–747, 1995.
- [15] I. O. Harrison, D. Dan'isa, A. A., and B. Ishaku, "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.
- [16] G. F. Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, vol. 5th. 2005.
- [17] I. A. Samuel, A. Emmanuel, I. A. Odigwe, and F. C. Felly-Njoku, "A Comparative Study of Regression Analysis and Artificial Neural Network Methods for Medium-Term Load Forecasting," *Indian J. Sci. Technol.*, vol. 10, no. 10, pp. 1–7, 2017.