TEPES
TURKISH JOURNAL OF
ELECTRICAL POWER
AND ENERGY SYSTEMS

TEPES Vol 1., Issue. 2, 99-107, 2021 **DOI:** 10.5152/tepes.2021.21043

RESEARCH ARTICLE

# A Comparative Study on the Performances of Power Systems Load Forecasting Algorithms

**Titus Oluwasuji Ajewole[1], Abdulsemiu Alabi Olawuyi[2], Mutiu Kolawole Agboola[3], Opeyemi Onarinde[1]**

[1]Department of Electrical and Electronic Engineering, Osun State University, Osogbo, Nigeria
[2]Department of Electrical and Electronic Engineering, Osun State Polytechnic Iree, Osun State, Nigeria
[3]Department of Electrical and Electronics Engineering Technology, Federal Polytechnic Ede, Osun State, Nigeria

**Cite this article as:** Ajewole Titus O, Olawuyi Abdulsemiu A, Agboola Mutiu K, Onarinde O. A comparative study on the performances of power systems load forecasting algorithms. *Turk J Electr Power Energy Syst.* 1(2), 99-107, 2021.

**ABSTRACT**

In this study, the efficiencies of three different neural network load forecasting algorithms are compared to determine the best performance. The algorithms—Levenberg–Marquardt, gradient descent, and gradient descent with momentum and adaptive learning rate backpropagation are used to train a neural network (NN) model for energy demand prediction on a power system. Prior loads, weather parameters (temperature, relative humidity, and precipitation), and customer population of the supplied region are employed as training inputs. To ascertain the accuracy of the predictions, mean absolute error and mean square error are used as evaluation indices, and the algorithm with the least index values is deployed on a transmission substation. The Levenberg–Marquardt algorithm was found to be the most efficient candidate, and this algorithm is therefore recommended for adequate and proper system management, planning, and expansion, to enhance the efficiency, effectiveness, and accessibility of power supply.

**Index Terms—**Algorithm, comparison, load forecasting, model training, neural network, transmission substation.

## I. INTRODUCTION

The ever-increasing human population and the need for industrialization have led the human race to a dire need for stable and quality electrical energy [1, 2]. Proper planning for adequate electrical energy is therefore an absolute necessity. Load forecasting is an important planning practice in power system industries, as its relevance stems from both the energy perspective and the economic angle [3]. Accurate load forecasting has many benefits, both managerially and economically. In the absence of efficient and effective forecasting of load, wastage is inevitable. Thus, robust forecasting is absolutely essential for the stakeholders in the energy sector [3]. Electrical load forecasting plays a key role for energy providers, economic consortia, and other corporations in the domain of electrical energy [4]. However, for a load forecast to best serve its ultimate purpose, it must be accurate, fast, and robust [5]; and the loss function should be optimally minimized [6].

There has been a lot of attention on load forecast studies using different methods with various time bounds [7]. While some studies have used statistical techniques [8-10], there are others that have used the artificial intelligence (IA) algorithms or machine learning models [11, 12]. One of the machine learning models that has gained a lot of relevance in load forecasting is the neural network (NN), which is a machine learning pattern that mimics the working function of the brain [13]. Machine learning uses data and produces a model to perform a task [14].

Load forecast in a power system is generally classified into short-term load forecast (STLF), medium-term load forecast (MTLF), and long-term load forecast (LTLF) [4]. However, [5] presents a fourth type, with the addition of very-short-term load forecast (VSTLF). The VSTLF has the least time of forecast, as [6] highlights that the period of this forecast is from one minute to one day. Conversely, [4] proposed that the range of VSTLF is from a few minutes to an hour ahead. The time range given by the latter is worth noting because if the time range extends to a day, then it is STLF [15]. The predictions of load for various time horizons are noted for various operations [10]. Very-short-term load forecast is significant because it helps the electric utilities and grid operators in making important decisions on real-time scheduling of electricity generation, real-time operation,

**Corresponding author:** Abdulsemiu Alabi Olawuyi, abdulsemiuolawuyi@gmail.com

**Received:** August 18, 2021
**Accepted:** September 20, 2021
**Published online:** October 15, 2021

demand–response, security assessment, sensitivity analysis, and load frequency control [3]. Furthermore, it is also helpful in real-time control of the electrical power system [4]. While [16] proposes that load prediction from a few hours to a few days is STLF, the authors in [4, 8–10] are more specific that STLF is often between an hour and one week. Short-term load forecast also gives hourly forecast results and is useful in power system decision making in overload condition and in spinning reserve planning [6]. It also plays an important role in grid stability [16], and moreover, [15] add that STLF provides useful notifications for power system administrators to enhance load usage. In the case of MTLF, the range is mostly between 7 days and 12 months [3, 13, 14], and its significance includes providing the power system stakeholders with adequate notification for system expansion, power system equipment requirements, and employment of staff [17]. Any load prediction that is for more than a year is grouped as LTLF [18], which lasts years and even decades, and is useful for future expansion, planning, as well as recruitment of staff [15, 19].

Considering its numerous aspects of importance in power systems, load forecasting needs to be efficient and effective. The various techniques used in forecasting power system loads are grouped into three, namely, the statistical or classical or parametric method, the machine learning or non-parametric method, and the hybrid method [1]. Because electrical loads are affected by several factors like class of consumers, variation in the calendar, holidays, the time of day, economic activities, random activities like sports and festivals, meteorological parameters, and so on, load-prediction techniques need to be compared for optimal choice. Among the meteorological factors, temperature is the most important and most common input [16, 17]. In an evaluation of the statistical methods as presented in [8], three analytical techniques are employed to address the MTLF problem, with mean absolute percentage error and root mean square error used as evaluation metrics. A comparison of the three techniques shows that the technique of linear regression performs better than both compound growth and quadratic regression techniques. The NN is employed by [1] to predict a power system, with mean square error (MSE) and mean absolute error (MAE) used as evaluation indices in the work which compares the backpropagation neural network (BPNN) and the radial basis function neural network (RBFNN). The BPNN has a better model, with ten hidden neurons, while the RBFNN has better architecture, with 15 neurons. In the work, the Levenberg–Marquardt (LM) algorithm performed better than the GD with momentum and adaptive learning rate backpropagation (GD+) algorithm. Meanwhile, the shortcoming in the work relates to the large values of the performance metrics. In Ref [19], NN performed better than the support vector machine, k-nearest neighbors, generalized regression neural network, and the Gaussian process regression and recurrent neural network; with the least value of 1.5 during the validation process, while the other machine learning methods had values greater than 1.5.

In this present study, three different algorithms are compared, as they are employed to train the artificial neural network (ANN) and to ascertain the one that performs optimally. To ascertain the accuracy of the predictions, MAE and MSE are used as evaluation indices. The optimal algorithm is consequently used for electrical load prediction in a transmission substation and then recommended for adequate and proper system management, planning, and expansion, to enhance the efficiency, effectiveness, and accessibility of power supply. The rest of this paper is structured as follows: while Section II presents the methodology of the study, the results obtained and the analyses of same are contained in Section III, and Section IV concludes the study.

## II. MATERIALS AND METHODS

Performances of Levenberg–Marquardt (LM), GD, and gradient descent with momentum and adaptive learning rate backpropagation (GD+) are compared to ascertain the optimal algorithm, as the three are used in the training of ANN. The best performing one is thereafter deployed for load prediction on a transmission substation. The Osogbo Substation in Southwest Nigeria is strategically located very close to the National Control Centre; therefore, the Transmission Company of Nigeria uses the substation for grid stability. Electrical load data were obtained from the Regional Control Centre, while information on weather parameters was obtained from the National Aeronautics and Space Administration (NASA), and the population data were obtained online. As shown in Fig. 1, feed-forward backpropagation is employed in modeling the NN, with six inputs—temperature, relative humidity, precipitation, population, actual load of year 2011 and actual load of year 2012—feeding the model. While Table I shows the values of the input parameters, Fig. 2 shows that there are 15 neurons in the hidden layer of the model, with hyperbolic tangent as the activation function.

The target of the model is the actual load for year 2013, which is the model's output. The MSE and MAE are used to evaluate the network, and are described as [1]:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=0}^{N}\left(y - y\right)^2 \tag{1}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=0}^{N}\left|y - y\right| \tag{2}$$

For design and training of the NN, the perceptron and the algorithms are described. Shown in Fig. 3 is the block diagram of the perceptron, while Fig. 4 depicts a single-layer NN.

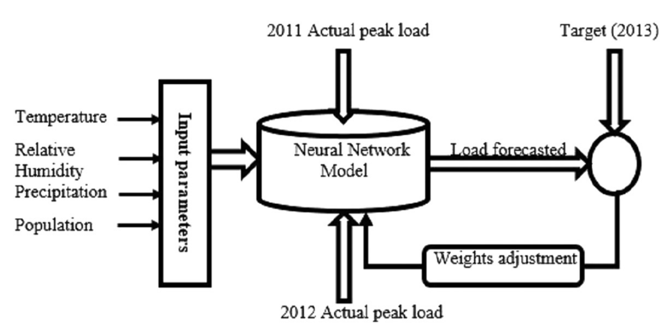


**Fig. 1.** ANN block diagram representation.

**TABLE I**
INPUT PARAMETERS OF THE NEURAL NETWORK

| | | | Input Parameters of the Neural Network | | | |
|---|---|---|---|---|---|---|
| **Months** | ***T*** **(°C)** | **Relative Humidity** | **Precipitation (mm)** | **Population** | **2011 Peak Load (MW)** | **2012 Peak Load (MW)** |
| January | 24.18 | 70.90 | 31.98 | 614917 | 66.1 | 65.2 |
| February | 25.73 | 77.81 | 47.31 | 615834 | 63.7 | 66.2 |
| March | 26.14 | 84.09 | 60.72 | 616750 | 66.8 | 68.9 |
| April | 25.69 | 84.15 | 138.97 | 617667 | 69.4 | 68.9 |
| May | 25.02 | 87.72 | 209.95 | 618583 | 70.9 | 72.1 |
| June | 24.35 | 88.58 | 204.55 | 619500 | 60.4 | 77.5 |
| July | 23.60 | 89.26 | 214.75 | 620416 | 62.1 | 77.5 |
| August | 23.39 | 88.86 | 182.19 | 621333 | 65.1 | 79.6 |
| September | 24.14 | 89.25 | 370.49 | 622249 | 60.7 | 67.9 |
| October | 24.81 | 88.39 | 203.07 | 623166 | 56.4 | 68.7 |
| November | 24.96 | 83.43 | 72.98 | 624083 | 68.9 | 79.0 |
| December | 24.45 | 70.68 | 28.27 | 625000 | 66.1 | 78.4 |

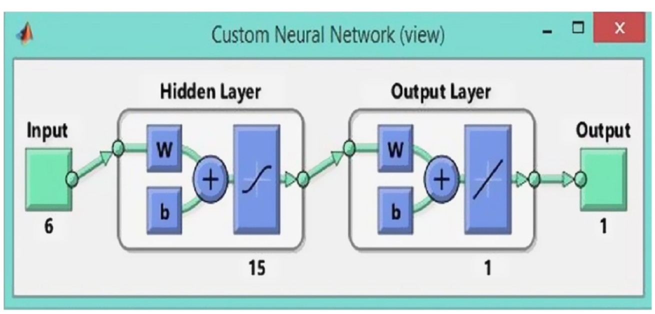


**Fig. 2.** Neural network architecture.

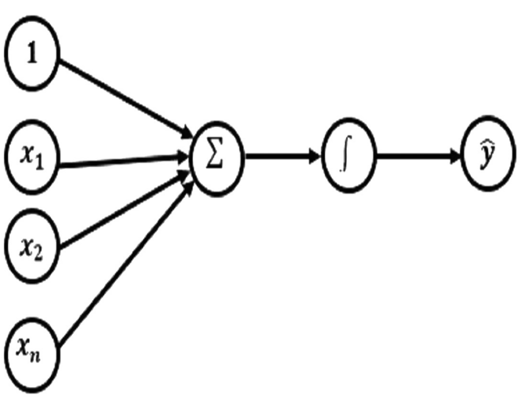


**Fig. 3.** A perceptron.

For the perceptron:

$$y = g\left(w_0 + X^T W\right) \tag{3}$$

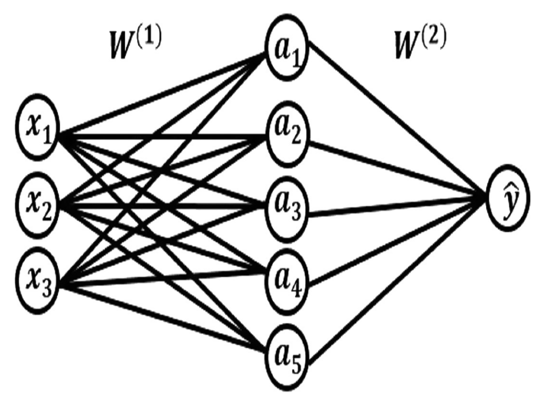


**Fig. 4.** Single-layer neural network.

where $y$ is the output, $g$ is the activation function, $w_0$ is the bias, $X$ is the inputs matrix, and $W$ is the network weights [6].

In the NN, the weights that separate the inputs and the hidden stratum are $W^{(1)}$, while those weights that separate the hidden stratum and the final stratum are $W^{(2)}$. As given in [6], the hidden layer is described as:

$$a_i = w_{0,i}^{(1)} + \sum_{j=1}^{m} x_j w_{j,i}^{(1)} \tag{4}$$

Thus, the hidden layer output will be $g(a_i)$ which corresponds to the inputs that feed the output layer:

$$y = g\left(w_{0,i}^{(2)} + \sum_{j=1}^{d1} g\left(a_j\right) w_{j,i}^{(2)}\right) \tag{5}$$

Therefore,

$$y = g\left(w_{0,1}^{(2)} + g(a_1)w_{1,1}^{(2)} + g(a_2)w_{2,1}^{(2)} + g(a_3)w_{3,1}^{(2)} + g(a_4)w_{4,1}^{(2)} + g(a_5)w_{5,1}^{(2)}\right) \tag{6}$$

The algorithms of this study are LM, GD, and GD+. While LM is a modification of Newton's method [20], GD is a classical algorithm for weight updates in the NN [21], and extension to GD produces the GD+ [22].

### A. Levenberg–Marquardt

Being a modification of Newton's method, LM is represented using Newton's equation [23]:

$$x_{k+1} = x_k - H[x_k]^{-1} g_k \tag{7}$$

where H is the Hessian matrix, $x_k$ the current value of $x$, $g$ is the gradient, and $x_{k+1}$ is the updated value of $x$. The Hessian matrix may not be positive definite. Hence, the LM modification addresses this shortcoming by adding $\mu_k I$ to the Hessian matrix. I is an identity matrix and $\mu_k \geq 0$. Thus,

$$x_{k+1} = x_k - \left(H\left(x_k\right) + \mu_k I\right)^{-1} g_k \tag{8}$$

And by introducing a step size, $\alpha_k$ (8) becomes,

$$x_{k+1} = x_k - \alpha_k \left(H\left(x_k\right) + \mu_k I\right)^{-1} g_k \tag{9}$$

Furthermore, when $\mu_k \to 0$, the LM modification tends to behave like the pure Newton's method. Also, when $\mu_k \to \infty$, the algorithm attains a pure GD with a small learning rate. The LM algorithm is, on the other hand, obtained from the Gaussian method [20] in (10):

$$x_{k+1} = x_k - \left(J^T J\right)^{-1} J^T e \tag{10}$$

The Jacobian matrix is denoted by $J$ and $e$ stands for network errors. Therefore,

$$x_{k+1} = x_k - \left(J^T J + \mu_k I\right)^{-1} J^T e \tag{11}$$

### B. Gradient Descent

For the GD algorithm, the loss function is minimized by calculating the slope, which is used in updating the weights, and is mathematically modeled as [24]:

$$x_{k+1} = x_k - \alpha_k g_k \qquad (12)$$

From (12), $\alpha_k$ is the learning rate, and in the NN, the weights are updated to optimize the errors; $x_k$ denotes the previous weights, while $x_{k+1}$ denotes the updated weights; and $g_k$ is the derivative of the loss function with respect to the weights. During training, the LM algorithm moves from being close to GD to being close to Newton's method. This shows that the LM algorithm is the hybridization of GD and Newton's method

### C. Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation

Produced by extension to the GD, the GD+ algorithm ensures elimination of the possibility of being trapped in the local minimum during the training process, by adding a momentum constant to the GD algorithm as [22].

$$x_{k+1} = x_k - \alpha_k V_t \qquad (12)$$

Where,

$$V_t = \beta V_{t-1} + \left(1-\beta\right) g_k \tag{13}$$

Where, $\beta$ is momentum constant, taking values $0 < \beta < 1$. When $\beta = 0$, (13) becomes $V_t = g_k$. Therefore, when the momentum constant is zero, GD is obtained. The default value of $\beta$ is 0.9 [25].

## III. RESULTS AND DISCUSSION

### A. Correlation Analyses of the Inputs Variables

Fig. 5–8 represent the correlation plots of the input variables in the NN with respect to electrical load, in order to verify the effects of the inputs on the load. The temperature has a positive correlation of 0.3606 as shown in Fig. 8, which implies that an increase in temperature will lead to an increase in electrical load in the supplied region. Moreover, relative humidity, precipitation, and population have corelation coefficients of –0.3458, –0.4394 and –0.2533, respectively. They all have negative correlation with respect to the load. However, the correlation of population shows that an increase

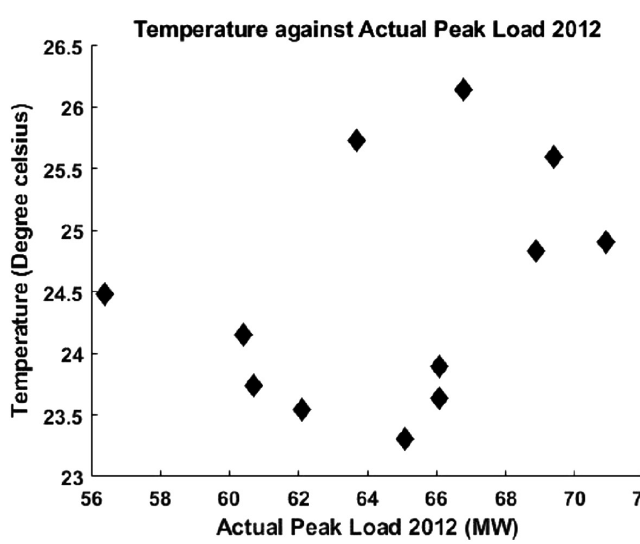


**Fig. 5.** Scatter plot of temperature (OC) against actual peak load (MW) of 2012.

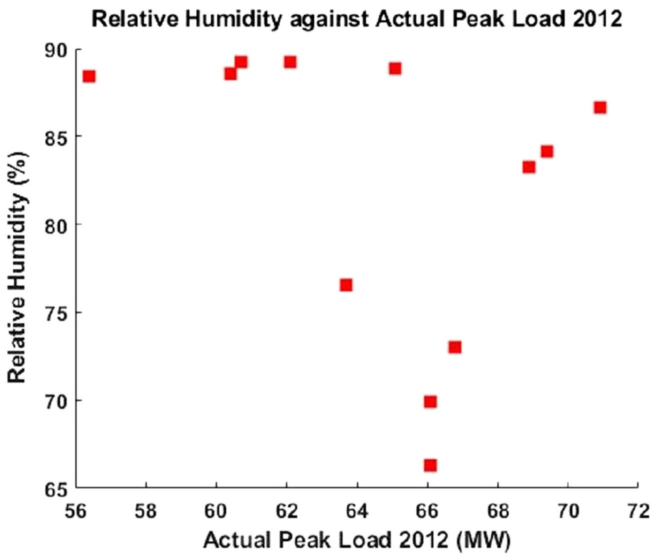


**Fig. 6.** Scatter plot of relative humidity (%) against actual peak load (MW) of 2012.

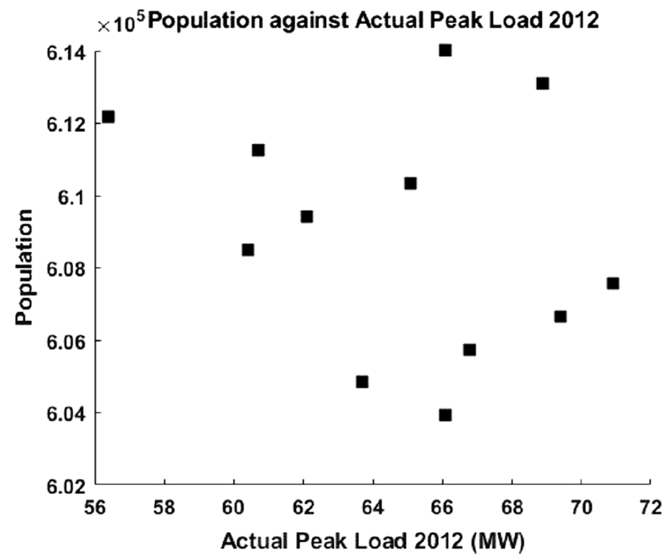


**Fig. 8.** Scatter plot of population against actual peak load (MW) of 2012.

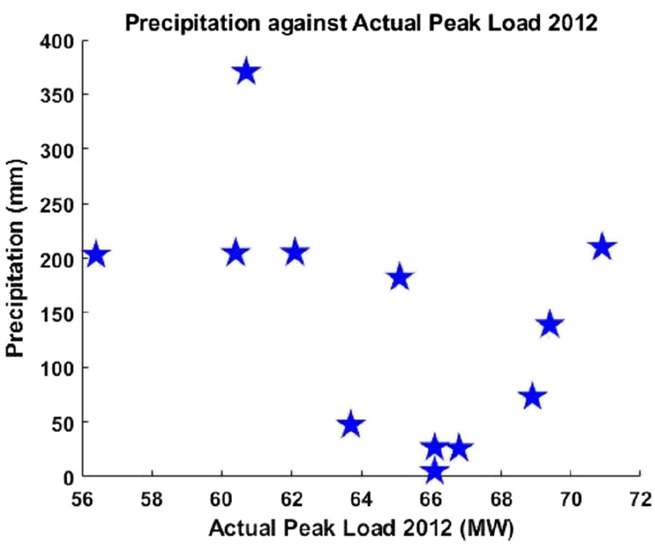


**Fig. 7.** Scatter plot of precipitation (mm) against actual peak load (MW) of 2012.

in population does not translate to increase in electrical load in the region under study. This problem could be mitigated by using renewable energy [26]. The stakeholders ought to look at this aspect critically to enhance the development needed in Osogbo, because the availability and accesibililty of elecrtrical load are synonymous with development.

**B. Regression Analyses**

The datasets used for the simulation were divided into 70% for training and 15% each for validation and testing of the NN model. The regression plots for the three algorithms have been presented in Fig. 9–11. Each of the plots has the output against the target. The closer the target to the output, the better the regression plots. Likewise, the more the regression value is to 1, the better. The output value represents the equation of a straight line. The coefficient of the target is the gradient and the constant value is the intercept on output axis. Also, the more the slope is to unity and the intercept to zero, the better the regression plot. Each of the algorithms has four different plots; the training, the validation, the test, and the all plots. The plots of the LM algorithm are shown in Fig. 9. The algorithm was well trained and so has regression value of 1, while the GD and GD+ algorithms have values of 0.9968 and 0.98741 respectively. All the three algorithms performed well during validation and testing, as each has a regression value of unity. However, the all plots give the overall best performing algorithm. The LM, GD, and GD+ algorithms have values of 0.96799, 0.83317 and 0.93658 respectively. These results mean that the LM algorithm has the best performance during the training, because its value of 0.96799 is the closest to 1.

**C. Performance Metrics of the Algorithms During Training**

The best performing algorithm was also validated using the evaluation metrics. The MAE and MSE functions in the MATLAB Neural Network toolbox were used to evaluate the performance of the three algorithms during the training process. The MAE and MSE of the algorithms are shown in Fig. 12. The LM algorithm has the least values of MAE and MSE, 0.602 and 2.0768 respectively, while GD has the highest values, 1.4559 and 9.9834 respectively, and GD+ performed better than GD because of the momentum it adds and because its learning function could adapt better. The work of [1] also proved that LM is better than GD+. Theoretically, both GD and GD+ are first-order algorithms while LM is a second-order algorithm [23],

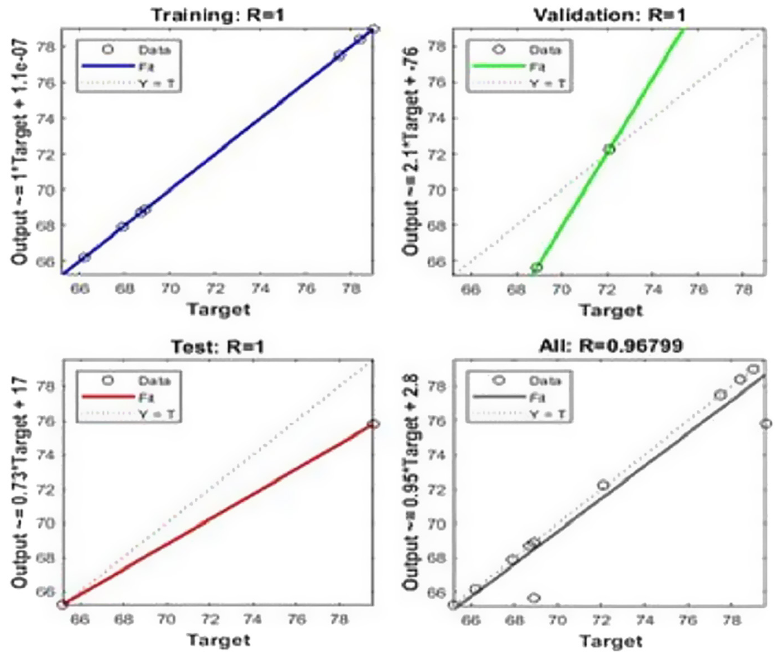


**Fig. 9.** Levenberg–Marquardt algorithm regression plots.

which can solve more complex problems. Consequently, LG was deployed as the forecasting algorithm in this study.

**D. Training and Prediction of the LM Algorithm**

Fig. 13 shows the plots of LM during training. The graph illustrates that the target loads are equal to the output loads, except for the months of March and August. The overall errors are nearly zero. This showcases the good performance of the LM algorithm during training process of the NN model. This model was then used for prediction as presented in Fig. 14, which shows that the forecasted load is closest to the actual load for the months of April, May, and September. The errors are between the range 10 and –10, while the

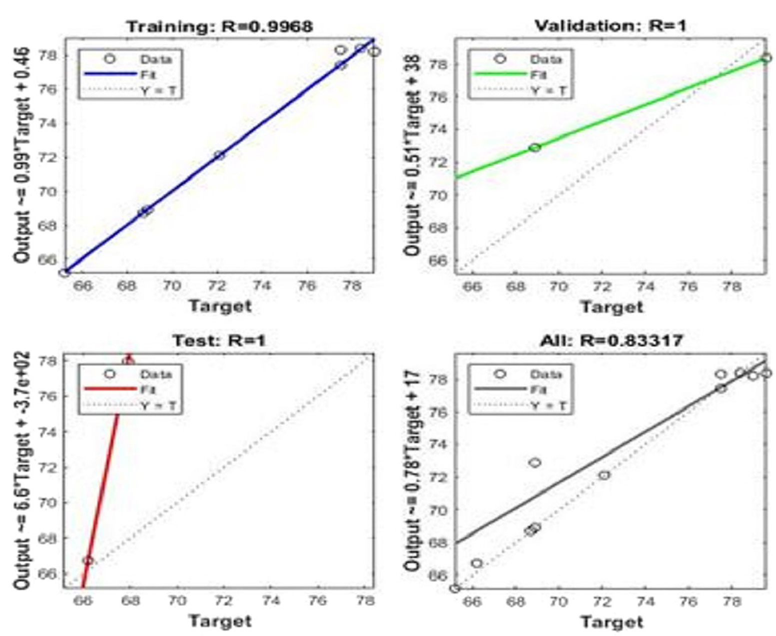


**Fig. 10.** Gradient descent algorithm regression plots.

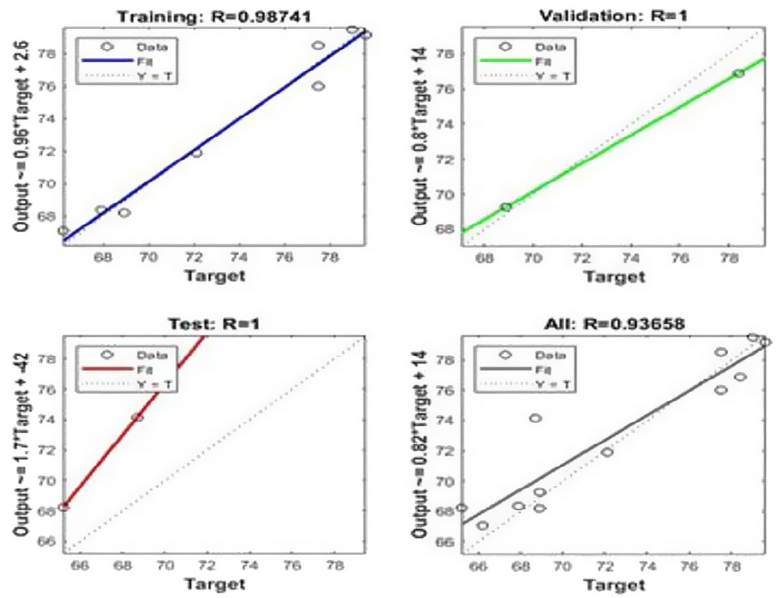


**Fig. 11.** Gradient descent algorithm with momentum and adaptive learning rate regression plots.

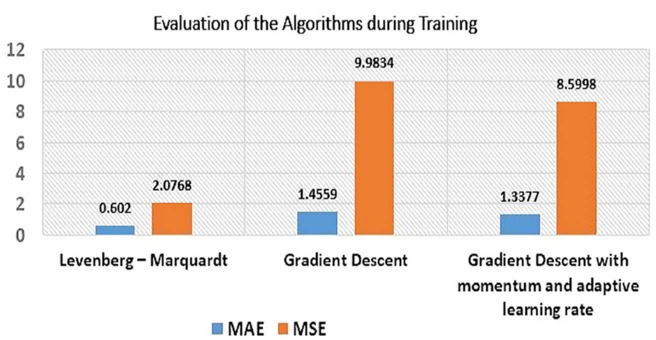


**Fig. 12.** Evaluation of the algorithms during training.

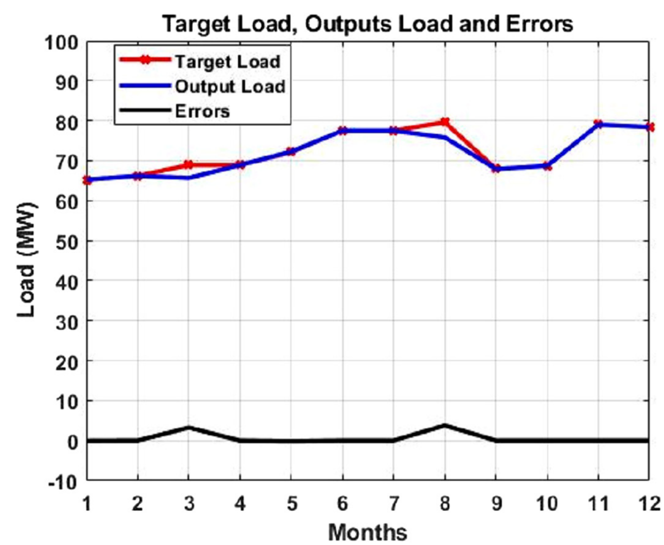


**Fig. 13.** Target load, output load, and errors using the Levenberg–Marquardt algorithm.

average prediction error is –2.05301. The LM algorithm has relatively good performance in this study, as illustrated in Table II which shows MAE to be 6.4675 and MSE 57.9962. The value of MSE is always greater than MAE because MSE penalizes errors more than MAE, as shown in (1) and (2).

## IV. CONCLUSION

Three different NN algorithms have been compared for their electrical load forecasting efficiencies. A NN model was developed for energy demand prediction on power systems, and the Levenberg–Marquardt, gradient descent, and gradient descent with momentum and adaptive learning rate algorithms were used to train the model. The training inputs were prior loads, weather parameters (temperature, relative humidity, and population), and population of the supplied region. From the correlation study of the inputs, it is found that the temperature has a positive correlation of 0.3606, implying that an increase in the temperature will lead to increase in electrical load in the supplied region. In addition, relative humidity, precipitation, and population have a negative correlation of –0.3458, –0.4394, and –0.2533 respectively. The correlation of the population shows that an increase in population does not translate to an increase in electrical load in the region under study. The accuracy of the prediction was appraised using MAE and MSE as evaluation indices; and the algorithm with the least index values was considered the best. Levenberg–Marquardt was found to be the most efficient technique, and was recommended

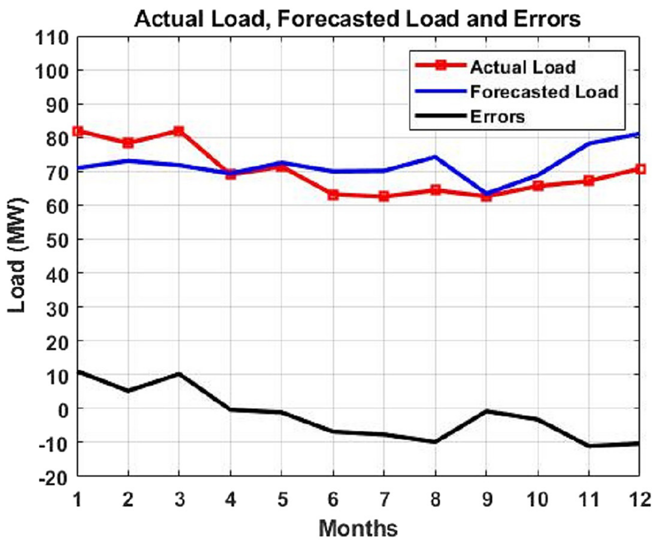


**Fig. 14.** Actual load, forecasted load, and errors with the Levenberg–Marquardt algorithm.

**TABLE II**
PERFORMANCE FUNCTION OF THE LM ALGORITHM DURING PREDICTION

| Indices | LM Algorithm |
|---|---|
| MAE | 6.4675 |
| MSE | 57.9962 |

for adequate and proper system management, planning, and expansion, to enhance the efficiency, effectiveness, and accessibility of power supply.

**Peer-review:** Externally peer-reviewed.

**Acknowledgments:** The authors thank the Regional Control Center, Osogbo, for providing the necessary data on the electrical load utilization of the area from 2011 to 2015. The authors also appreciate the National Aeronautics and Space Administration (NASA), for providing the weather parameters needed for the study.

**Conflict of Interest:** The authors have no conflicts of interest to declare.

**Financial Disclosure:** The authors declared that this study has received no financial support.

## REFERENCES

1. E. A. Feilat, D. T. Al-Sha'abi and M. A. Mo-mani, "Long-term load forecasting using neural network approach for Jordan's power system," *Eng. Press*, vol. 1, no. 2, pp. 43–50, 2017.
2. T. O. Ajewole, O. D. Momoh, O. D. Ayedun and M. O. Omoigui, "Optimal component configuration and capacity sizing of a mini integrated power supply system," *Environ. Qual. Manag.*, vol. 28, no. 4, pp. 57–62, 2019. [CrossRef]
3. R. Patel, M. R. Patel and R. V. Patel, "A review: Introduction and understanding of load forecasting," *J. Appl. Sci. Comput.*, vol. 4, pp. 1449–1457, 2019.
4. W. Aribowo and S. Muslim, "Long-term electricity load forecasting based on cascade forward backpropagation neural network," *J. Telecommun. Electron. Comput. Eng.*, vol. 12, no. 2, pp. 1449–1457, 2020.
5. P. Dixit, "Daily peak load forecast using artificial neural network," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, p. 2256, 2019.
6. A. Amini, "6S191_MIT_DeepLearning_L1.pdf - Introduction to Deep Learning Alexander Amini MIT," *6.S191 Introduction to Deep Learning | Course Hero*, 2020. [Online]. Available: https://www.coursehero.com/file/54813875/6S191-MIT-DeepLearning-L1pdf/.
7. S. Singh, S. Hussain and M. A. Bazaz, "Short-term load forecasting using artificial neural network," in Fourth International Conference on Image Information Processing (ICIIP), 2017, pp. 1–5.
8. O. Olabode, O. Amole, T. Ajewole and I. Okakwu, "Medium-term load forecasting in a Nigerian electricity distribution region using regression analysis techniques," in International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), 2020, pp. 1–5.
9. H. Matsila and P. Bokoro, "Load forecasting using statistical time series model in a medium voltage distribution network," in IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, 2018, pp. 4974–4979.
10. A. Y. Saber and A. R. Alam, "Short-term load forecasting using multiple linear regression for big data," in IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–6.
11. A. Baliyan, K. Gaurav and S. K. Mishra, "A review of short-term load forecasting using artificial neural network models," *Procedia Comput. Sci.*, vol. 48, pp. 121–125, 2015. [CrossRef]
12. S. Khatoon and A. K. Singh, "Effects of various factors on electric load forecasting: An overview," in 6th IEEE Power India International Conference (PIICON), 2014, pp. 1–5.
13. Z. Zhang, "A gentle introduction to artificial neural networks," *Ann. Transl. Med.*, vol. 4, no. 19, p. 370, 2016. [CrossRef]
14. S. H. Han, K. W. Kim, S. Kim and Y. C. Youn, "Artificial neural network: Understanding the basic concepts without mathematics," *Dement. Neurocogn. Disord.*, vol. 17, no. 3, pp. 83–89, 2018. [CrossRef]
15. T. Anwar, B. Sharma, K. Chakraborty and H. Sirohia, "Introduction to load forecasting," *Int. J. Pure Appl. Math.*, vol. 119, pp. 1527–1538, 2018.
16. G. M. U. Din and A. K. Marnerides, "Short-term power load forecasting using deep neural networks," in International Conference on Computing, Networking and Communications (ICNC), 2017, pp. 594–598.
17. S. M. Elgarhy, M. M. Othman, A. Taha and H. M. Hasanien, "Short-term load forecasting using ANN technique," in Nineteenth International Middle East Power Systems Conference (MEPCON) [Internet]. Cairo: IEEE Publications, 2017, pp. 1385–1394. Available: http://ieeexplore.ieee.org/document/8301364/.
18. N. Ammar, M. Sulaiman and A. F. M. Nor, "Long-term load forecasting of power systems using artificial neural network and ANFIS," *ARPN J. Eng. Appl. Sci.*, vol. 13, no. 3, pp. 828–834, 2018.
19. H. Sangrody, N. Zhou, S. Tutun, B. Khorramdel, M. Motalleb and M. Sarailoo, "Long-term forecasting using machine learning methods," in IEEE Power and Energy Conference at Illinois (PECI), 2018, pp. 1–5.
20. M. Buhari and S. S. Adamu, "Short-term load forecasting using artificial neural network," in Proceedings of the International Multi-Conference of Engineers and Computer Scientists, Hongkong, vol. 1, pp. 1–71, March 14–16, 2012.
21. X.-S. Yang, "Optimization techniques and applications with examples," *Optim. Tech. Appl. Examples*, 2018.

22. S. H. Haji and A. M. Abdulazeez, “Comparison of optimization techniques based on gradient descent algorithm: A review,” *PalArchs J. Archaeol. Egypt Egyptology*, vol. 18, no. 4, pp. 2715–2743, 2021.
23. H. Yu and B. M. Wilamowski, “Levenberg–Marquardt training,” in *Intelligent systems*. Boca Raton: CRC Press, 2018, pp. 12–11.
24. P. Netrapalli, “Stochastic gradient descent and its variants in machine learning,” *J. Indian Inst. Sci.*, vol. 99, no. 2, pp. 201–213, 2019. [CrossRef]
25. MathWorks. *MATLAB (R2018a)*. USA: MathWorks, 2018.
26. M. O. Lawal, T. O. Ajewole and O. S. Babalola, “Fortification of the 330-KV Nigerian network using renewable energy resources,” *Niger. J. Eng.*, vol. 26, no. 1, pp. 33–40, 2019.