Transactions on Machine Intelligence 3(4) (2020) 226-239

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Energy Demand Prediction in Smart Grids Using Neural Networks Based on Optimization

A. Mohammadi-Pour[1,*], M. Setayesh-Nazar[2]

[1] Power Group, School of Electrical Engineering, Shahid Beheshti University, Tehran, Iran
[2] Associate Professor, School of Electrical Engineering, Shahid Beheshti University, Tehran, Iran

ARTICLE INFO

Article History:
Received 14 June 2020
Received in revised form 23 October 2020
Accepted 20 December 2020
Available online 21 December 2020

Keywords:
Load Forecasting, Neural Network, Energy Consumption, Demand-Side Management, Optimization Techniques, Genetic Algorithm, Smart Grid

ABSTRACT

Demand prediction plays a crucial role in the real-time operation of electrical systems, particularly for monitoring, planning, and optimizing the operation of electrical devices. Accurate demand prediction ensures effective coordination between consumers and power companies, which is essential for efficient power grid management. This paper presents a novel approach for energy demand prediction using a neural network combined with an optimization-based method. Initially, a conventional neural network is employed to predict the required energy demand based on historical data. However, to improve prediction accuracy, a genetic algorithm (GA) is introduced to adjust the neural network's weights automatically. This optimization method fine-tunes the network, enabling it to achieve better performance in predicting short-term energy demand. The integration of the genetic algorithm helps in overcoming the limitations of traditional training methods, such as slow convergence or local minima. Experimental results, based on real-time data randomly selected from various sources, demonstrate that the genetic algorithm-based neural network outperforms conventional approaches in terms of prediction accuracy and computational efficiency. The proposed method is validated through extensive testing, showing its potential for accurate short-term load forecasting in dynamic and complex energy systems. This research highlights the effectiveness of optimization algorithms in enhancing the predictive power of neural networks for energy demand forecasting.

1. INTRODUCTION

Among the factors creating a smart grid are demand-side management, home area networks, advanced metering, and planning and forecasting [1,2]. Currently, demand-side management is an essential component of the electric energy network, identifying electricity needs, controlling usage, and consumer activities to keep issues related to electric energy (blackouts, brownouts, and power outages) under control. Specifically, load forecasting plays a crucial role in improving the efficiency of the smart grid at the distribution end, particularly for power plants [3]. Additionally, if consumers install renewable energy sources at home, it is suggested they sell the excess energy to

* Corresponding Author: a.mohammadipour@mail.sbu.ac.ir
Power Group, School of Electrical Engineering, Shahid Beheshti University, Tehran, Iran

http://dx.doi.org/10.47176/TMI.2020.226

the grid [4]. Population growth increases energy demand and electrical equipment costs. Proper energy utilization, achievable through various mechanisms such as monitoring, control, and forecasting or weather prediction, is essential to prevent costly equipment failures [5]. Energy use and demand prediction are fundamental and complex tasks. Proper coordination between consumers and power companies is needed for monitoring, planning, and operating electrical devices. An optimization-based neural network approach for energy demand prediction is proposed. The conventional neural network approach is used to find the required energy demand prediction on the consumer side. The genetic algorithm-based neural network approach is executed where the neural network weights are automatically adjusted.

Energy transmission from the power plant to the distribution network or end consumer is highly costly. Before planning to create transmission network infrastructure, demand response information is an additional advantage for calculating the capacity of the electric network. This capacity factor is entirely based on the loads' and future generations' needs. Demand prediction and response also play a crucial role in the power transmission network and help shape transmission development planning [6]. Energy use or load forecasting can generally be classified into (a) very short-term forecasting, (b) short-term forecasting, (c) medium-term forecasting, and (d) long-term forecasting [7]. Very short-term forecasting can be used to determine the energy needed for a specific day or hour or even for a few minutes, helping minimize costs in power plants [8]. Short-term forecasting is used to predict the required energy in terms of days or weeks. Medium-term forecasting can be used for weeks to months. Long-term forecasting infers energy needs for several years, aiding in understanding overall consumer behavior in the power grid. Additionally, understanding consumer behavior helps create future power grid infrastructure and power plant generation [9].

Generally, artificial intelligence and statistical techniques can solve real-world problems [10]. Artificial intelligence uses a combination of fuzzy logic, neural networks, genetic algorithms, support vector machines, and machine learning techniques [11]. The genetic algorithm is an evolutionary algorithm [12].

A neural network is a graph-based structure with processing elements called nodes connected with varying weights and adjustable using a learning method to obtain the desired output. The main idea behind the neural network is to mimic the human brain. It acquires knowledge through learning, and the knowledge obtained is stored in the connection strengths (weights). Through appropriate training, testing, and validation, the artificial neural network nearly produces the correct solution to formulated problems. Currently, artificial neural networks are widely used for energy demand prediction, energy market price forecasting [13], and renewable energy sources prediction [14].

The literature on energy consumption prediction issues based on neural network methods is reviewed. In some cases, neural networks alone cannot perform well, necessitating improving the training results using optimization algorithms like the genetic algorithm. The integrated model of the neural network with an optimization algorithm provides better results than the conventional neural network. The proposed method optimizes the weights during the neural network training session to achieve faster convergence rates with high accuracy. The motivation for this research is to implement an optimization-based neural network for demand-side management in the smart grid. The main innovations of this paper are as follows:

- A new adaptive intelligent optimization-based neural network approach for energy consumption prediction.

- Using an evolutionary genetic algorithm to automatically adjust the artificial neural network weights to achieve faster convergence with higher efficiency.

The rest of the paper is organized as follows: Section 2 discusses the modeling and characteristics of conventional artificial neural networks and optimization-based approaches for energy prediction. Section 3 examines various energy demand prediction scenarios and their results using optimization-based neural network methods. The conclusion and future directions of the proposed work are presented in Section 4.

2. MODELING

In the modeling section, the methods of artificial neural networks and genetic algorithm-based artificial neural networks are examined, and the governing equations are analyzed. In the second part, the proposed hybrid method is presented.

2.1. Neural Network

Artificial neural networks are a type of information processing that, inspired by human brain neural networks, analyze, process, and learn methods, patterns, and information. A neural network results from learning through changes in weights and directions. The principle of learning in neural networks is through repetition. That is, the algorithm is injected with a set of data multiple times, and it can recognize differences in the training data by adjusting weights and directions.

Backpropagation, an algorithm for supervised learning of neural networks, calculates the gradient of the error function concerning the weights of the neural network for a given network and error function. The classic backpropagation algorithm is designed for regression problems with sigmoid activation units, though backpropagation can also be used for classification problems and networks with non-sigmoid activation functions. The sigmoid function's mathematical properties, when combined with an appropriate output activation function, make the algorithm easy to understand. Neural networks are designed and exist in layers, with input, output, and hidden layers. Recorded data values form the input layer, which is the first layer, while the output layer, the last layer, contains output nodes for each class with its specific values. The hidden layer lies between the input and output layers. In an artificial neural network, there may be single or many hidden layers.

Once it is determined how much error the algorithm has based on the weights and directions, the second stage in an iteration begins. In this stage, weights and directions can be updated. That is, weights and directions are modified to produce a result closer to the actual output with less error in the next iteration. This process repeats until the network output for all training data approaches the nearest actual value (i.e., the value provided by the training data).

Thus, the difference between the neural network output and the expected values is the error, which must be minimized. To minimize this error, the genetic algorithm optimization is used. This algorithm will be explained in the following.

Figure 1 shows the artificial neural network. This figure includes input blocks, the artificial neural network, and the target or output block.

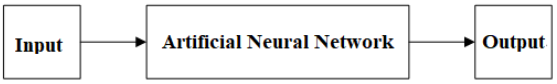


Fig.1. Block Model of the Neural Network

Figure 2 shows the input block with the number of inputs x1 to xn, the artificial neural network block with hidden layers, transfer functions, and activation functions. A sigmoid function is widely used to convert the activation level of a neuron, which is the weighted sum of inputs, to an output signal.

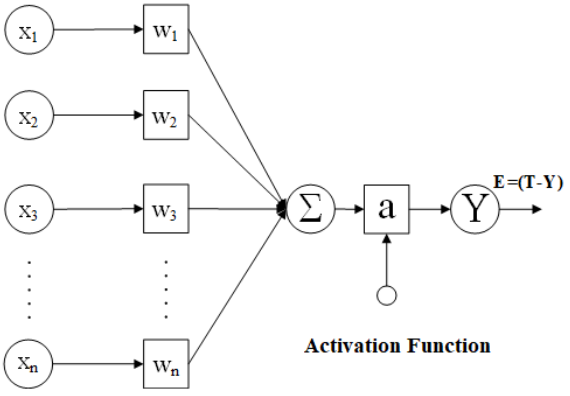


Fig.2. Computational Model of the Neural Network

Generally, sigmoid functions are often used to introduce nonlinearity into the artificial neural network model. A neural network element computes a linear combination of its input values and applies a sigmoid function to obtain the result. The popularity of the sigmoid function in artificial neural networks stems from its ability to satisfy the derivative and itself properties. Additionally, it is easy to compute and implement compared to other functions. Hence, examining errors during the neural network training process is easily done. The output represents the actual target result. The general neural network model can be represented by the following equation:

$$Y=x_i \times w_i + g \tag{1}$$

where y denotes the output, xi represents the set of input values, wi represents the weights, and g is the activation function. For example, the training process of an artificial neural network starts with the input neurons and their corresponding weights. The system evaluates the amount of error or difference from the actual and desired output, then continues to adjust weights and refine the algorithm until the inputs produce the desired values, achieving higher efficiency and accuracy, and this process is repeated.

The main challenge in artificial neural networks is selecting datasets for training, validation, and testing. Some networks never learn because the input data do not contain information to derive the desired output. If sufficient data are not available for complete learning, the networks do not converge. Ideally, there should be enough data to hold some as a validation set. In the past, only a few data points, such as production, load patterns, and pricing models, were considered for training artificial neural networks. Most existing studies only consider a set of data for prediction or prediction-related issues, and in some studies, only random data are used.

The neural network, which includes the input layer, hidden layers, and output layer, is described. If X is the input to the input layer, then the neural network can be represented as:

$$X = x_1, x_2, x_3, \ldots, x_4 \tag{2}$$

Given input values from x_n ، … ، x_2،x_1 , a neural network consists of various layers, each with its own specific weights for training. These can be represented as:

$$W = W(1), W(2), W(3), \ldots, W(n) \tag{3}$$

Where W denotes the weight matrix for the neural network, and W(1), W(2), ..., W(n) represent the weight vectors for n layers. If there are n hidden layers, the bias B can be expressed as:

$$B = B(1), B(2), B(3), \ldots, B(n) \tag{4}$$

Weights and biases are initially generated randomly and later adjusted during the neural network training to achieve the desired output. The net input (Z) for (n) layers in the neural network is computed as follows:

$$Z = Z(1), Z(2), Z(3), \ldots, Z(n) \tag{5}$$

Where Z(2), Z(3), …,Z(n) represent the net input values from layer 1 to n. The product of input values and weight values in the input layer is shown as $\sum((X) * W))$, where W1 denotes the weight value for the corresponding layer, and the bias values for the input layer are B(1). The net input Z for the first hidden layers is then calculated as:

$$Z(1) = B(1) + \sum((X) \times W(1)) \tag{6}$$

If A is the activation function for the individual layers in the neural network, it is represented as follows:

$$A = A(1), A(2), A(3), \ldots, A(n) \tag{7}$$

Where A(1), A(2), …,A(n) indicate the activation function in each hidden layer from 1 to n. The activation function A is applied to the net input for all hidden layers of the neural network involved during the training session:

$$A(1) = \frac{1}{1+e^{-Z(1)}} \tag{8}$$

Here, due to computational efficiency and quick implementation, the applied activation function is the sigmoid function. To evaluate the net input in the output layer, the following relationship is considered:

$$Q = B(n) + \sum(A \times W(n)) \tag{9}$$

Where Q is the net input of layer n. The n-th net input layer n is calculated as the sum of the bias values and the sum of the product of the activation function and weight values. This equation is used to find the net input of the final output layer. Now, the output layer is considered, and the activation function f(Q) is applied to the net input Q in the output layer to obtain the actual learning output Y:

$$Y = f(Q) \tag{10}$$

Where Y represents the actual output. The obtained output value is compared with the existing target value to determine the error:

$$E = T - Y \tag{11}$$

Where E is the error value, T is the target value, and Y is the obtained output. The difference between the output and the target is the error value. The objective function min(E) is used in the proposed method to minimize the error value and is also used to reduce the error during the training phase of the neural network. The main goal of the objective function is to minimize the error value for better convergence.

$$E(1) = (T - Y)f'(Q) \tag{12}$$

Where f'(Q) is the derivative of the cost function and is used to find the accuracy of the error criterion. This equation is used to calculate the error E = T - Y or the difference in output. This expression evaluates how quickly values change when weights and biases are automatically adjusted. If the error is greater than the expected value, then the error must be minimized by adjusting the weights. For instance, if the error value is greater than the tolerance error level, an optimization algorithm is executed to obtain a new set of weights. During the neural network training process, each iteration error value is checked. If the error is minimal, the training process may stop; otherwise, weights must be readjusted in the next iteration until the error is minimized. To find the weight change in the output layer, the equation can be presented as follows:

$$W_c(1) = \alpha \times E(1) \times A \tag{13}$$

Where α is 0.3 for the training process. Changes in bias values during neural network training can be expressed as follows:

$$B_c(1) = \alpha \times E(1) \tag{14}$$

Where $B_c(1)$ represents the bias changes in the n-th layer. Bias changes, if any, during neural network training improve the result. Now, the error value for hidden layers in the neural network is computed using the following equation:

$$Z'(n) = \sum(E(1) \times W(n)) \tag{15}$$

Where Z' represents the reverse input net, and Z'(n) is used to find the net input (E × W) in the reverse phase. The error value in the hidden layer is equal to:

$$E(2) = Z(n)f'(Z(n)) \tag{16}$$

Where $f'(Z)$ is the derivative of the cost function with respect to the weight in the reverse phase. The weight change in hidden layers during recursive updating or reverse phase can be calculated as follows:

$$W_c(2) = \alpha \times E(2) \times Z(n) \tag{17}$$

Similarly, changes in bias values in hidden layers during forward or reverse updating can be expressed as follows:

$$B_c(2) = \alpha \times E(2) \tag{18}$$

In the output layer, after the optimization technique, the new weights are updated as follows:

$$W_n^{'}(1) = W(2) \times W_c(1) \tag{19}$$

Where $W'_n(1)$ represents the new weight value for the output layer. The new biases after the optimization technique involved in the output layer are updated as follows:

$$B_n^{'}(1) = B(2) \times B_c(1) \tag{20}$$

Where B' represents the new bias values for the output layer in the reverse direction. In the hidden layer, weights and biases are updated using these equations respectively:

$$W_n(2) = W(1) \times W_c(2) \tag{21}$$

$$B_n(2) = B(1) \times B_c(2) \tag{22}$$

The results obtained from the two above steps are fed back from the last hidden layer to the input layer to update the new weights and bias values in each layer in the reverse direction.

2.2. Genetic Algorithm-Based Neural Network

Similar to the human brain, neurons in an artificial neural network are trained with known historical data to learn the real-time environment by comparing their classifications. To prevent incorrect learning, the error from previous iterations is fed back to the network, and weights are optimized. Figure 3 generally shows the weight optimization technique with main components. Finally, the optimization block performs the weight optimization of the network.

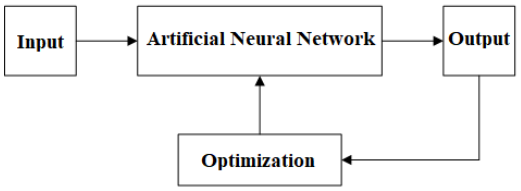


Fig.3. Block Diagram of Genetic Algorithm-Based Neural Network Optimization

Another important task in a neural network is selecting the appropriate network model for the given application. Neither a suitable mechanism nor an appropriate method is available for this. To date, researchers have developed and tested various network approaches to identify the suitable network model for required applications. There is a need to find a better optimal approach to solve the optimization problem with high accuracy. All weights, along with hidden connections, need to be computed to achieve the desired output. Therefore, there is a need to select or develop a better neural network algorithm to obtain a general optimal solution. Figure 4 presents a detailed model of how weight computations and optimizations occur in a neural network.

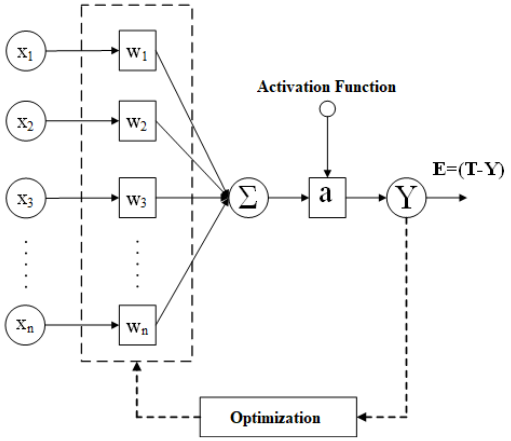


Fig.4. Computational Model of Neural Network Optimization

After determining the network structure for a specific application, the network can be trained. There are two main methods for training the network: supervised and unsupervised training. If the network is provided with input and output, in a way that it can process them and compare the resulting value with the desired output, it is known as supervised learning. Errors are then displayed through the system, which can be used to adjust weights in the hidden layers, and the process is repeated to achieve accurate weights. If the network is provided with inputs but not with desired outputs, it is adaptive or unsupervised learning. The backpropagation algorithm is the most popular neural network algorithm for prediction problems. The neural network is trained by taking individual rows of data and initial weights through an optimization technique. During the learning process, errors may occur with the respective inputs. Major advantages of neural networks include their high flexibility with time series data and their ability to classify and predict patterns they were not trained on.

In the genetic algorithm, initially, the input layer of the neural network reads input values X. Weights are randomly generated using the genetic algorithm, considered as the initial population. The net input in the hidden layer is computed using an appropriate activation function. Initialization and application of the activation function in each layer are performed to obtain the output. At the end of each iteration, the neural network calculates the error by comparing the target T with the obtained output Y. The fitness function evaluates the objective function and identifies the model fitness values. Genetic operations such as selection, crossover, and mutation are performed to generate a new population. Weight values are updated with the new population set, and the process is repeated until convergence. In each iteration, the error value is checked. If the error value is minimized, the neural network supported by the genetic algorithm has obtained the required results, and the process stops; otherwise, the process repeats until the error value is minimized.

The genetic algorithm is an evolutionary algorithm that may be called a search method for linear and nonlinear problems. The genetic algorithm mimics the process of natural selection. The best solution will survive, and the rest of the solutions are eliminated from the selection list. This starts with the initial population (weights). Based on the validation of the weights from the obtained population, the most suitable among the initial population is selected. For subsequent steps and to eliminate minimum values, 40% is retained [15]. The most suitable weights are crossed with each other to create new generations. Then there is an opportunity for mutation, and finally, the obtained population is reached. Therefore, during this process, the genetic algorithm trains the neuron and updates the weights in each iteration. The flowchart for the genetic algorithm-based neural network approach is shown in Figure 5.

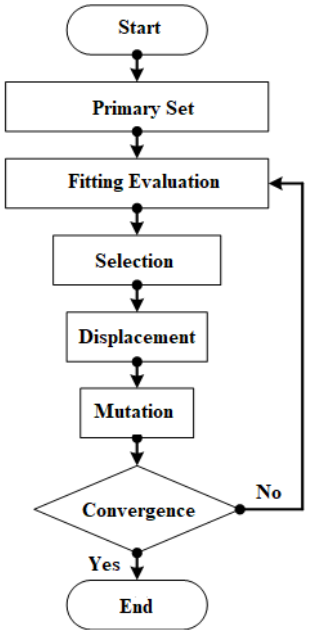


Fig.5. Flowchart for Genetic

3. RESULTS

The dataset available for training, testing, and validation is divided into three sets. The collected dataset is sufficient for performing neural network predictions with appropriate scheduling types, such as short-term and medium-term forecasts. Predictions are made one day in advance with a resolution of 15 minutes, which is more suitable for satisfying the neural network model. The inputs, hidden layer neurons, and target variables are determined before starting the experiment. The input variables are day, month, and year. The hidden layer is determined and used to improve the accuracy and precision of network training. The electrical energy consumption variable is involved as a target data point during the experiment.

The results are obtained using the MATLAB simulation environment, which is very suitable for time-series-based prediction problems. To achieve better performance for energy forecasting, a neural network optimization-based approach is implemented. The results obtained are classified as follows: (a) energy consumption pattern for a day and (b) short-term load prediction for the consumer.

3.1. Energy Consumption in Summer

Figure 6 shows the energy consumption pattern for the summer season on an hourly basis on a summer day. Figure 6 clearly shows that energy consumption is high during the day due to the extensive use of cooling appliances (e.g., air conditioning). On summer holidays, the energy consumption pattern is much lower compared to summer working days. Additionally, the power generated from renewable energy sources (especially solar) is high in this season.

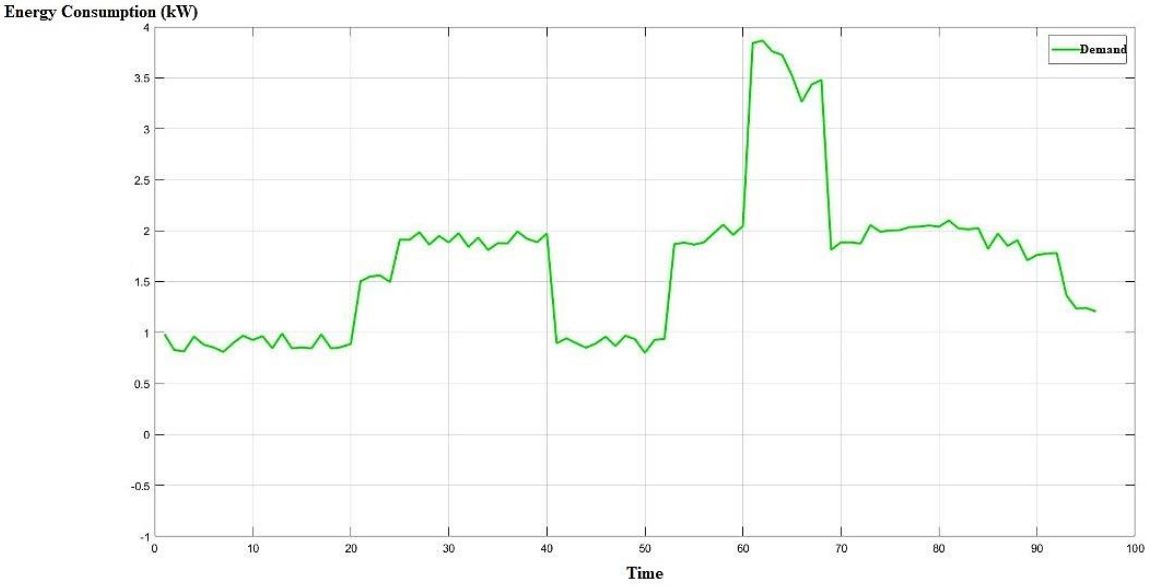


Fig.6. Energy consumption chart for a summer day

3.2. Energy Consumption in Winter

Figure 7 shows a winter day on an hourly basis. Figure 7 indicates that electricity consumption in winter is lower compared to summer. It is also stated that the power generated by renewable energy sources is lower than that of a summer day, due to the average day being either sunny or cloudy.

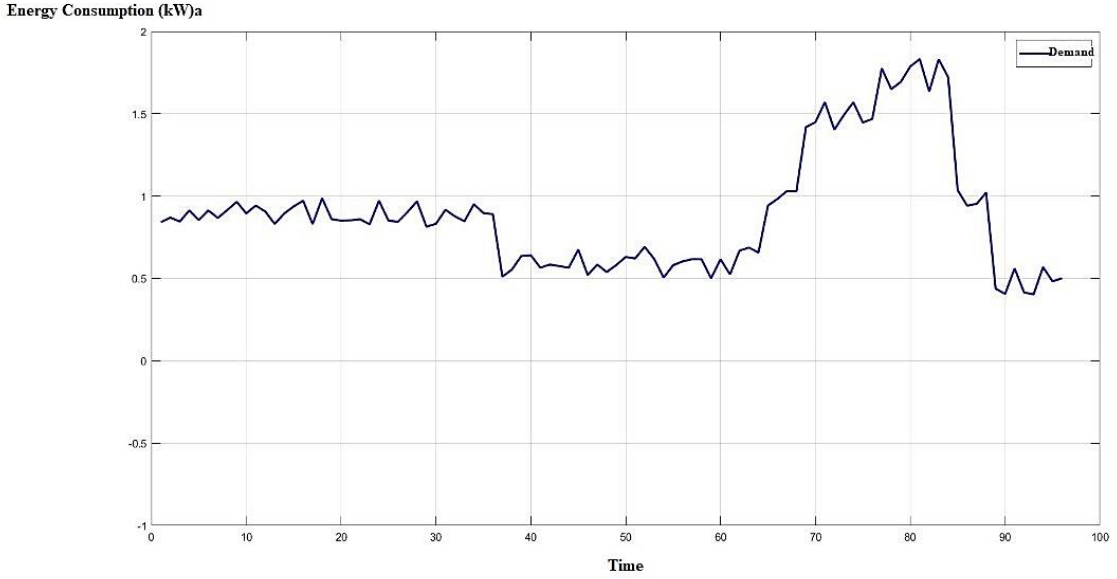
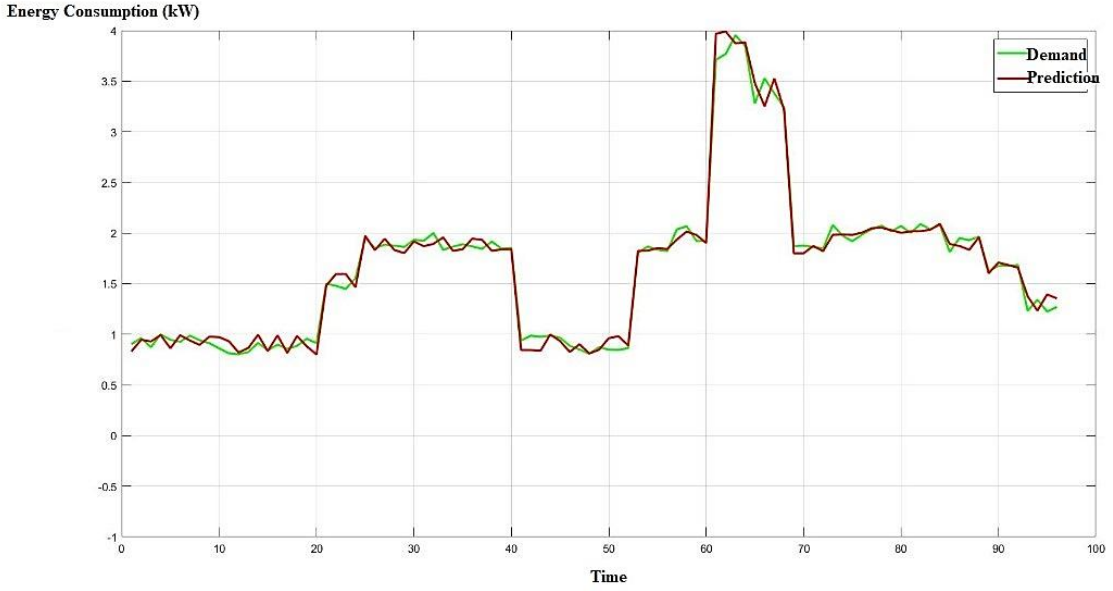


Fig.7. Energy consumption chart for a winter day

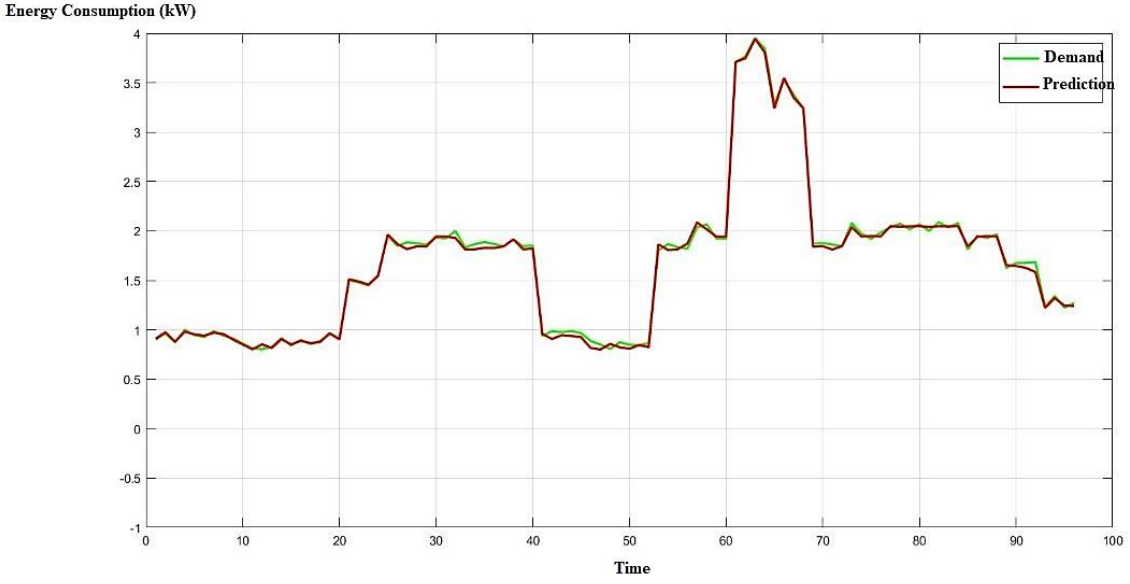
3.3. Energy Demand Forecast for One Day

The average daily energy consumption forecast for one day is shown in Figure 8, which clearly illustrates the consumer's energy consumption pattern behavior over time intervals. A single time interval equals 15 minutes,

meaning input and target data are collected every 15 minutes from the consumer to achieve high effectiveness in neural network training. The results obtained between energy demand prediction and actual energy consumption are shown in Figure 8. Conventional neural network demand prediction is shown in Figure 8 (a), and the integration of a genetic algorithm-based neural network is shown in Figure 8 (b). By analyzing one day's energy consumption prediction, Figure 8 shows that the genetic algorithm-based neural network approach is almost closer to actual consumption compared to the conventional neural network approach. Therefore, the results clearly demonstrate that the genetic algorithm-based neural network produces the highest accuracy for very short-term energy usage and prediction when compared to the conventional neural network algorithm.



(a) Conventional method

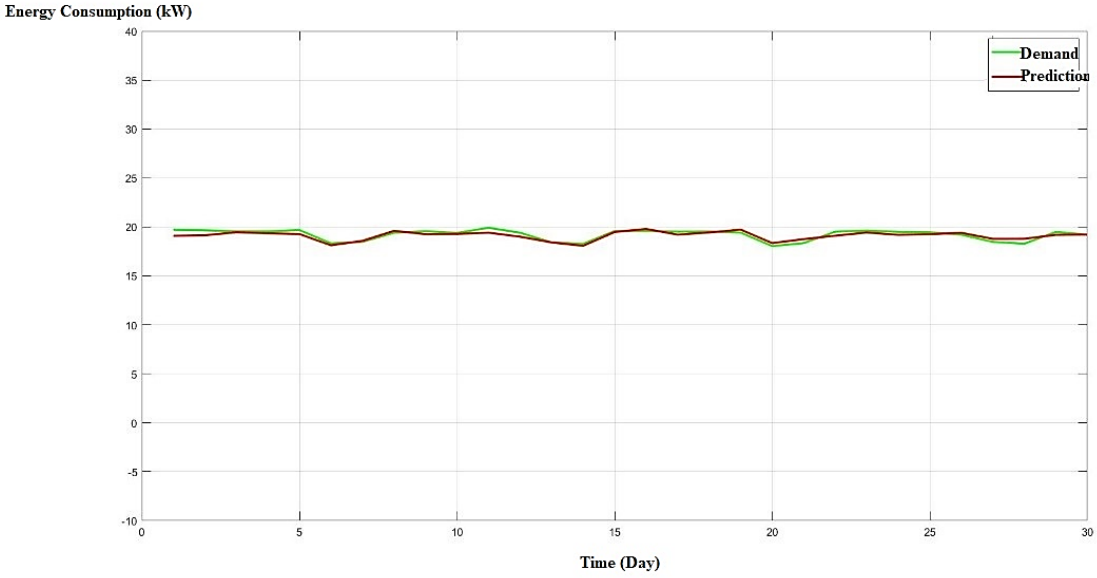


(b) Genetic Algorithm

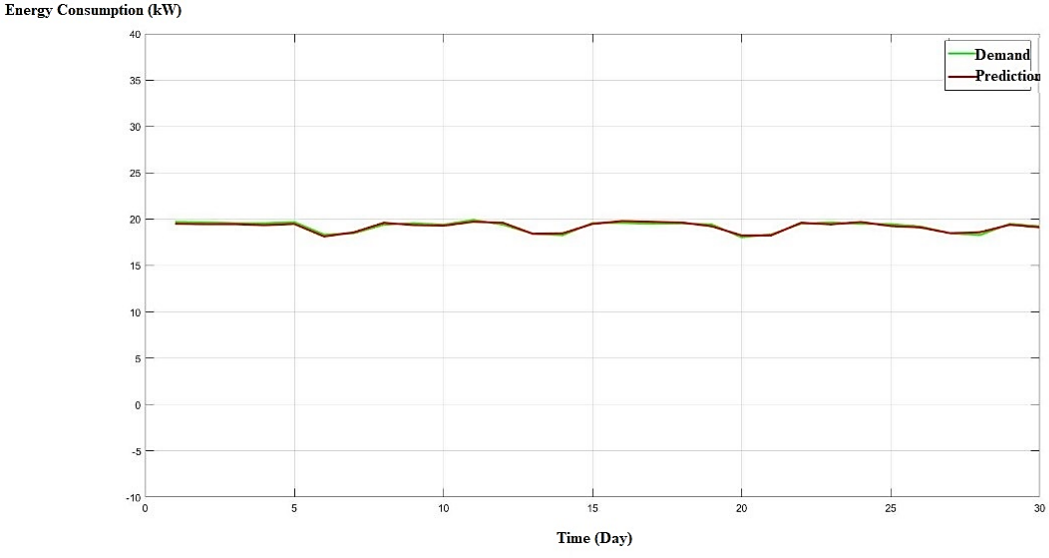
Fig.8. Energy demand and prediction for one day

3.4. Energy Demand Forecast for a Month

Figure 9 shows the energy consumption forecast and usage in kilowatts for one month. The conventional neural network and the genetic algorithm-based neural network for energy prediction are shown in Figures 9 (a) and (b), respectively. The figures illustrate that the genetic algorithm-based neural network produces the minimum error compared to the conventional neural network prediction. It is also shown that the genetic algorithm-based neural network achieves high accuracy in short-term energy usage and prediction.



(a) Conventional method



(b) Genetic Algorithm

Fig.9. Energy demand and prediction for one month

3.5. Comparison of Weight Optimization Algorithms

Based on the above analysis for the considered case, it is observed that the model fitness values using the genetic algorithm-based neural network for short-term energy prediction are much lower than those of the conventional neural network. Thus, it can be concluded that the proposed genetic algorithm-based neural network provides better

convergence results for short-term energy prediction. Hence, the proposed genetic algorithm-based neural network algorithm with the network weight update strategy performs well, resulting in better energy prediction outcomes.

The mean squared error between actual output and desired output is computed to understand performance behavior. Figure 10 represents the mean squared error. The main objective of the genetic algorithm-based neural network is to reduce the mean squared error values. This serves as an appropriate stopping criterion for halting the optimization method.

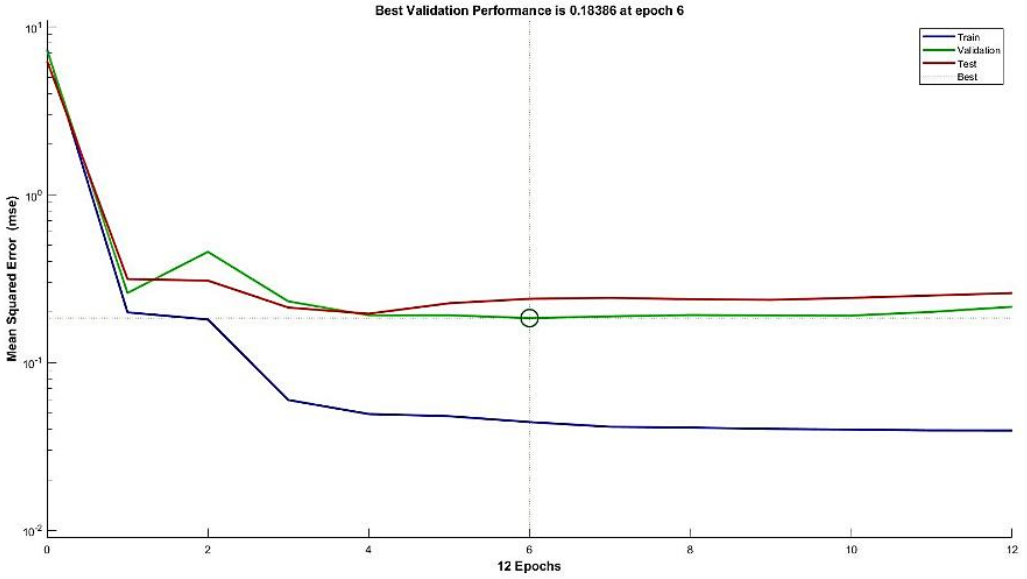


Fig.10. Mean squared error based on iterations

4. CONCLUSION

In this paper, an optimization-based neural network approach is employed for predicting consumer energy consumption, providing a beneficial analysis of energy demand and supply for consumers and service providers. On the consumer side, the neural network-based forecasting method is used to understand consumer behavior patterns under different weather conditions over a year, helping to maintain a minimum cost strategy in their energy bills. The service provider plans and constructs the power plant based on the predicted future energy demand. Moreover, the proposed method shows that the genetic algorithm-based neural network is suitable for short-term energy prediction. Accurate electricity demand prediction is crucial to prevent costly failures in household appliances and power grid infrastructure. The results indicate that the proposed system will be beneficial for energy management, demand-supply strategy, and financial support for consumers through efficient energy production. Therefore, the proposed model is effective in creating future power grid infrastructure with high adaptability for maximum utilization.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2020). Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 11106-11115. http://doi.org/10.1609/aaai.v35i12.17325

[2] Song, C., Lin, Y., Guo, S., & Wan, H. (2020). Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 34(01), 914-921. http://doi.org/10.1609/aaai.v34i01.5438

[3] Gudi, N., Wang, L., & Devabhaktuni, V. (2012). A demand side management based simulation platform incorporating heuristic optimization for management of household appliances. International Journal of Electrical Power & Energy Systems, 43(1), 185-193. https://doi.org/10.1016/j.ijepes.2012.05.023

[4] Lee, J., & Park, G. L. (2015). Dual battery management for renewable energy integration in EV charging stations. Neurocomputing, 148, 181-186. https://doi.org/10.1016/j.neucom.2012.10.045

[5] Yuce, B., Rezgui, Y., & Mourshed, M. (2016). ANN-GA smart appliance scheduling for optimised energy management in the domestic sector. Energy and Buildings, 111, 311-325. https://doi.org/10.1016/j.enbuild.2015.11.017

[6] Baringo, L., & Conejo, A. J. (2012). Transmission and wind power investment. IEEE Transactions on Power Systems, 27(2), 885-893. https://doi.org/10.1109/TPWRS.2011.2170441

[7] Ardakani, F. J., & Ardehali, M. M. (2014). Long-term electrical energy consumption forecasting for developing and developed economies based on different optimized models and historical data types. Energy, 65, 452-461. https://doi.org/10.1016/j.energy.2013.12.031

[8] Gottwalt, S., Ketter, W., Block, C., Collins, J., & Weinhardt, C. (2011). Demand side management-A simulation of household behavior under variable prices. Energy Policy, 39(12), 8163-8174. https://doi.org/10.1016/j.enpol.2011.10.016

[9] McPherson, M., & Karney, B. (2014). Long-term scenario alternatives and their implications: LEAP model application of Panama's electricity sector. Energy Policy, 68, 146-157. https://doi.org/10.1016/j.enpol.2014.01.028

[10] Kavousi-Fard, A., Samet, H., & Marzbani, F. (2014). A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate short term load forecasting. Expert Systems with Applications, 41(13), 6047-6056. https://doi.org/10.1016/j.eswa.2014.03.053

[11] Ahmad, A. S., Tahar, R. M., & Azeez, S. M. A. (2014). A review on applications of ANN and SVM for building electrical energy consumption forecasting. Renewable and Sustainable Energy Reviews, 33, 102-109. https://doi.org/10.1016/j.rser.2014.01.069

[12] Uykan, Z. (2013). Fast-convergent double-sigmoid Hopfield neural network as applied to optimization problems. IEEE Transactions on Neural Networks and Learning Systems, 24(6), 990-996. https://doi.org/10.1109/TNNLS.2013.2244099

[13] Amjady, N. (2006). Day-ahead price forecasting of electricity markets by a new fuzzy neural network. IEEE Transactions on Power Systems, 21(2), 887-896. https://doi.org/10.1109/TPWRS.2006.873409

[14] Pinto, T., Praca, I., Morais, H., Santos, G., Vale, Z., & Silva, M. (2016). Adaptive portfolio optimization for multiple electricity markets participation. IEEE Transactions on Neural Networks and Learning Systems, 27(8), 1720-1733. https://doi.org/10.1109/TNNLS.2015.2461491

[15] Muralitharan, K., Sakthivel, R., & Vishnuvarthan, R. (2018). Neural network based optimization approach for energy demand prediction in smart grid. Neurocomputing, 273, 199-208. https://doi.org/10.1016/j.neucom.2017.08.017