Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 11, No. 3, September 2023, pp. 854~869
ISSN: 2089-3272, DOI: 10.52549/ijeei.v11i3.4607

Energy Management Analysis of Residential Building Using ANN Techniques

Lohit Kumar Sahoo[1]*, Mitali Ray[2], Sampurna Panda[3], Subash Ranjan Kabat[4], Smita Dash[5]
[1]Electrical and Electronics Engineering, Trident Academy of Technology, Bhubaneswar, Odisha, India
[2]Skill Development and Technical Education Department, Bhubaneswar, Odisha, India
[3]Institute of Technology & Management, Gwalior, M.P, India.
[4]Radhakrishna Institute of Technology and Engineering, Bhubaneswar, Odisha, India
[5]United School of Business Management, Bhubaneswar, Odisha, India

Article Info

Article historys:

Received Mar 17, 2023
Revised Jul 13, 2023
Accepted Sep 23, 2023

Keywords:

HVAC loads
Neural Network
Building Energy

ABSTRACT

The process of limiting the amount of energy that is utilized is known as energy conservation. This can be accomplished by making more effective use of the energy that is available. As a result, there is a requirement for more effective management of the consumption of energy in buildings. It is essential to have an accurate load calculation for a residential building because the loads for heating and cooling add up a significant portion of the total building loads. In this study, the load analysis of the HVAC (Heating, Ventilation, and Air Conditioning) system in a residential building was carried out by taking into consideration three different neural networks. These networks are known as the feed forward network, the cascaded forward back propagation network, and the Elman back propagation network. During the process of conducting a load study of the heating and cooling loads on an HVAC system, performance measurements like MAE (mean absolute error), MSE (mean square error), MRE (mean relative error), and MAPE (mean absolute percentage error) are taken into consideration. It has been discovered that the cascaded forward back propagation method is the most effective method, with MAE, MSE, MRE, and MAPE values of 0.08, 0.0336, 0.0051, and 0.51% respectively for heating load and MAE, MSE, MRE, and MAPE values of 0.0975, 0.0406, 0.0053, and 0.53% respectively for cooling load.

Corresponding Author:

Lohit Kumar Sahoo,
Electrical and Electronics Engineering, Trident Academy of Technology,
Bhubaneswar, Odisha, India,
Email: lohit86@gmail.com

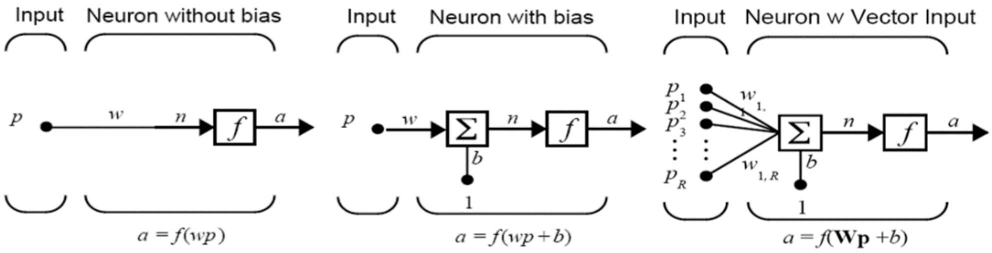
1. INTRODUCTION

The Energy Information Administration (EIA) estimates that buildings account for upto 36% of total world energy use at present [1]. In the realm of energy management and energy savings, the growing awareness of the importance of reducing energy waste and the negative impact of buildings on the environment has become a national sensation. It has been suggested by researchers that the high amount of electrical energy consumed by buildings can be attributed to HVAC systems, which are widely employed to maintain thermal comfort and appropriate air quality inside the building. The development of artificial neural networks paves the way for a wide range of opportunities for the creation of smart built environments and the reduction of overall energy consumption. The formulation of an optimization model for the allocation of retrofit investment funds has a significant impact on the environmental sustainability of buildings. This is made possible by the modification of HVAC systems as a cost-saving opportunity.

Energy simulation programs are widely used to anticipate the energy usage of buildings. However, computational time is increasing and programming skills limiting the usefulness in the building design and energy management [2]. ANNs are utilized in residential buildings for the purpose of forecasting thermal loads by taking into consideration the various physical aspects of the building [3, 4, 5] these approaches are convenient and also give fastest response. Despite being an important characteristic, the glazing area in [3] has no correlation to either heating load (HL) or cooling load (CL). There are many drawbacks to using back propagation learning methods to determine ANN weights and biases, including local minima and a decrease in accuracy. Four metaheuristic methods to construct an ANN model that uses MLP neural networks are used to optimize the HL and CL of the ANN model [6]. Furthermore, the most trustworthy neural network was found to be SSA-MLP. There are two metaheuristic algorithms, i.e. Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO), which are utilized to optimize the MLP. Furthermore, PSO has been proven to outperform ABC in terms of MLP efficiency improvement. This study compares the robustness of ABC and PSO as optimization algorithms. In [8] different ANNs are also used to predict electrical energy consumption of building.

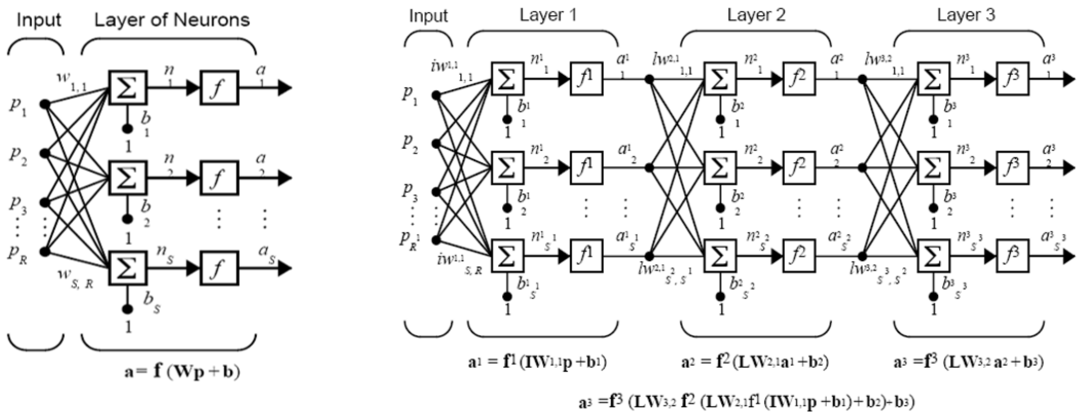
2. NEURAL NETWORKS

A neural network is a collection of algorithms that, by following a predetermined sequence of steps, attempt to determine whether or not a particular group of data points are related to one another. The neural networks basic structures are shown in Figure 1, where p is input, *R* is the number of inputs, *n* is the net input, *w* is weight, *b* is the bias, *a* is output, and *f* is the transfer function. Here, *w* and *b* are the adjustable parameters. Similarly, Figure 2 represents the structural representation of neural network layers.



(a) (b) (c)

Figure 1. Neural Network basic structures with (a) single input (b) single input with bias (c) multi-input with bias [7]



(a) (b)

Figure 2. Layer structure of Neural Network (a) Single Layer (b) Multi-Layer [7]

For the purpose of this research, ANNs with a variety of architectures, such as the Feed Forward (FF) Network, the Cascaded Forward Back propagation (CFBP) Network, and the Elman Back propagation (EBP) Network, are utilized to estimate the heating and cooling loads of residential buildings.

2.1 Feed forward neural network

To learn the mapping between the input variables and the output variables, FF network is utilized. FF stands for "feed forward" (outputs). In feed forward networks, there can be several layers of information. The first layer is the one that the network input is connected to. Every subsequent layer is connected to the layer that came before it. As can be seen in Figure 3, the output of the network is generated by the last layer. The network is less complex than other recurrent networks, and it enables pattern reorganization to be carried out in an effective manner. Because there is no feedback in this network, the results of one layer do not have an effect on the next layer.

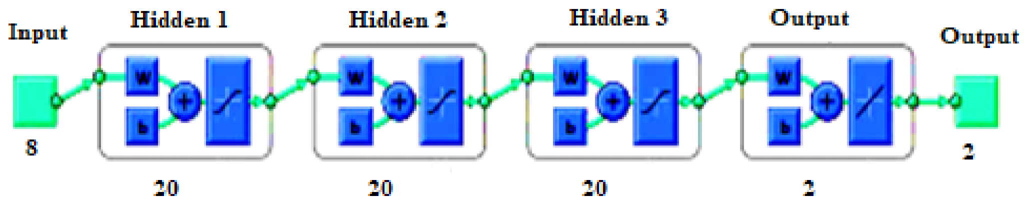


Figure 3. Feed Forward Neural Network [9]

In this case, a 3-hidden layer network with 20-20-20 neurons is taken into consideration. The tansig function serves as the output layer activation function, while the purelin function is used for the transfer function. The following is a mathematical expression that can be used to describe the output of a neural network:

$$Output(x) = purelin\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{3n}(x) + b_3]\right] \quad (1)$$

$$a_3(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{2n}(x) + b_2]\right] \quad (2)$$

$$a_2(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{1n}(x) + b_1]\right] \quad (3)$$

$$a_1(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}p_k(x) + b_0]\right] \quad (4)$$

Where:
a_1 , a_2 , a_3 are the outputs of hidden layers 1, 2, and 3, respectively.
b_1, b_2, b_3 are the biases of hidden layers 1, 2, and 3, respectively.
p_k is the input to first hidden layer
w_{nk} is the weight of neurons

2.2 Cascaded forward back propagation network.

The model of the Cascaded Forward Back Propagation (CFBP) network operates in a manner analogous to feed forward networks. The main distinction is that it incorporates a connection from the input as well as each of the layers that came before it to the layers that came after it, as seen in Figure 4. Any form of mapping from input to output can be accomplished with the help of this network. In addition to this, it is able to take into account the nonlinear relationship that exists between the input and the output without disrupting the linear relationship that exists between the two.

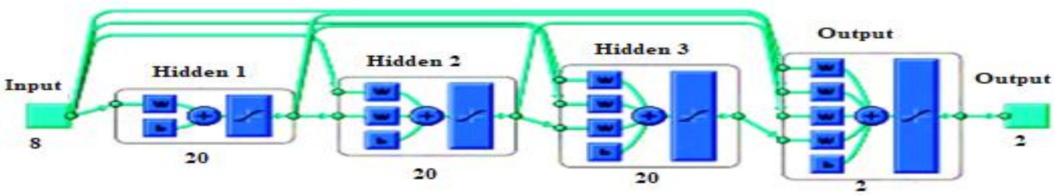


Figure 4. Cascaded Forward Back Propagation Network [9]

This study takes into account a three-hidden layer network that has twenty, twenty, and twenty neurons. The tansig function is regarded to be both the transfer function and the output layer activation function. The following is a possible definition for the mathematical expression of this neural network:

$$Output(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{3n}(x) + b_3 + w_{nk}p_k + w_{nk}a_{1n}(x)w_{nk}a_{2n}(x)]\right] \quad (5)$$

$$a_3(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{2n}(x) + b_2 + w_{nk}p_k(x) + w_{nk}a_{1n}(x)]\right] \quad (6)$$

$$a_2(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{1n}(x) + b_1 + w_{nk}p_k(x)]\right] \quad (7)$$

$$a_1(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}p_k(x) + b_0]\right] \quad (8)$$

2.3 Elman back propagation network

An Elman back propagation network, also known as an EBP network, is a two-layer back propagation network. EBP networks are characterized by the presence of a recurrent link between the output of the hidden layer and the input of the visible layer, together with a tap delay. As can be seen in Figure 5, the recurrent link gives the network the ability to not only recognize but also create patterns that vary over time.

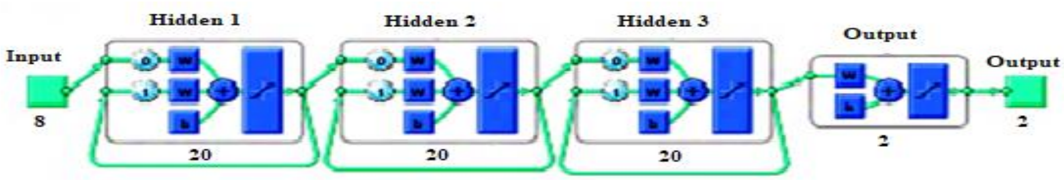


Figure 5. Elman back propagation network [9]

This study takes into account a three-hidden layer network with a total of twenty, twenty, and twenty neurons in each layer. The tansig function is considered to be both the transfer function and the output layer activation function, which is mathematically described here.

$$Output(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{3n}(x) + b_3]\right] \quad (9)$$

$$a_3(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{2n}(x) + b_2 + w_{nk}a_3(x-1)]\right] \quad (10)$$

$$a_2(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}a_{1n}(x) + b_1 + w_{nk}a_2(x-1)]\right] \quad (11)$$

$$a_1(x) = tansig\left[\sum_{k=1}^{8}\sum_{n=1}^{20}[w_{nk}p_k(x) + b_0 + w_{nk}a_1(x-1)]\right] \quad (12)$$

3. PROPOSED METHOD AND INPUT- OUTPUT DATASET

The performance metrics are generally determined by the types of networks, the amount of neurons, the training algorithm, the learning rate, as well as the performance target with 10-fold cross validation. When training and testing classifiers, a 10-fold cross validation is used. This means that the data sample is divided into 10 different groups. This results in a more accurate estimation.

3.1 Proposed ANN method

After that, the output that was generated by selecting an input from the input training pattern is compared with the output that was targeted. The objective of this study is to achieve the lowest possible value for the error function [10], as demonstrated by the following objective function:

$$min(E) = \frac{1}{2}\sum_{i=1}^{u}||b_i - t_i||^2 \quad (13)$$

Where b_i is Network output, t_i is Target output and E is Error function

3.2 Details of Dataset

The proposed method is experimentally validated on the basis of a data set collected from the UCI machine learning [3] repository. Angeliki Xifara was the one who developed the dataset, while Athanasios Tsanas was the one who processed it. To do the energy analysis necessary to gather 768 building shapes, 12 distinct building shapes were simulated in Ecotect [3] software. This allowed for the collection of 768 building shapes. The data set includes a total of two responses along with eight different attributes (X_1–X_8) (Y_1 and Y_2).

Tsanas and Xifara [3] present further information concerning the simulation experiments that were carried out. Ecotect was used to construct a sample data set with eight inputs and two outputs. By combining all these, 768 numbers of buildings were studied (12x3x5x4=720) + (12x4=48) = 768). The components, like walls, floors, clothing, and windows, are used with the lowest U-values. This U-value indicates the thermal transmittance of these materials. The lower the U-vale, the higher the heat resistance and the better the insulation. It can be represented by the following equations.

$$U = \frac{1}{R} \tag{14}$$

$$R = \frac{\Delta T}{\emptyset_q} \tag{15}$$

Where U is U-value, R is R-value, ΔT is temperature difference between the hot and cold surface of the barrier, and Φq is heat flux through the barrier

The building's other internal parameters, like humidity, air speed, lighting level, and occupancy, were kept constant. The humidity level was 60%, the air speed was 0.3 m/sec, the lighting level was 300 lux, and there were seven occupants. Table 1 provides the details of the dataset used in this study. Among these eight-input data, relative compactness (RC) has a great significance on the outputs. It is known as the RC-indicator to show the types of buildings by comparing the building's surface area and volume. This can be derived as,

$$RC = 6V^{\frac{2}{3}}A^{-1} \tag{16}$$

Where V is volume of building and A is surface area of building

Table 1. Input and output data description

Independent and response variables	Scope	Unit	Number of possible values	Minimum value out of all the values	Maximum value out of all the values	Mean of all the values
Relative compactness (X_1)	Surface area to volume ratio	N/A	12	0.62	0.98	0.76
Surface area (X_2)	Heat gain or loss is directly proportional to the area of a surface	m^2	12	514.5	808.5	671.71
Wall area (X_3)	Determine the surface area of the inside walls of a room that is rectangular using a calculator	m^2	7	245	416.5	318.50
Roof area (X_4)	Area of roof	m^2	4	110.25	220.5	176.60
Overall height (X_5)	Height of building	m	2	3.5	7	5.25
Orientation (X_6)	The positioning of a building in relation to seasonal shifts in the path of the sun and the direction from which the predominant winds blow is referred to as the building's orientation in the field of architecture.	N/A	4	2 5	5	3.50
Glazing area (X_7)	Glass used in the construction of the building's exterior or interior surfaces	%	4	0	0.4	0.23
Glazing area distribution (X_8)	Distribution of glazing area	N/A	6	0	5	2.81
Heating load (Y_1)	The amount of heat energy that would have to be provided to a location in order to maintain the temperature within a range that is considered to be acceptable for that location.	kW	586	6.01	43.1	22.31
Cooling load (Y_2)	To keep a space's temperature within an acceptable range, a significant amount of heat energy must be evacuated from it (via cooling).	kW	636	10.9	48.03	24.59

3.3 Statistical analysis of database

Using Spearman rank correlation analysis, the statistical analysis of this eight-input and two-output data set is performed on the data set. It is a measure of the intensity of monotonic association between two variables in a linear relationship. Essentially, it refers to how one variable changes as a result of a change in another variable. The Spearman correlation coefficient has values ranging from +1 to -1. Positive correlation between rankings is shown by a value of +1, whereas negative correlation between ranks is indicated by a value of -1, and zero shows no connection between ranks. According to statistics, a positive correlation indicates that two variables have a direct relationship with one another. On the other hand, a negative correlation suggests that the variables have an inverse association with one or more other variables. Both the positive and negative correlations provide insight into the relationship between the two variables. A positive correlation exists when two variables move in the same direction simultaneously, either up or down. It is claimed that two variables have a negative correlation when a rise in one variable results in a reduction in the other variable. The following is an example of how the formula for the Spearman rank correlation coefficient might be expressed (3).

$$\rho = \frac{1/n[\sum_{i=1}^{n}(R(x_i) - \overline{R(x)}).(R(y_i) - \overline{R(y)})]}{\sqrt{\left(1/n\sum_{i-1}^{n}((R(x_i) - \overline{R(x)})^2\right) \times \left(1/n\sum_{i-1}^{n}((R(y_i) - \overline{R(y)})^2\right)}} \tag{17}$$

Where, $R(x)$ and $R(y)$ are ranks of two variables.

Table 2. Correlation matrix of the input variables and the output variables

Input variables	Correlation coefficient with HL	Correlation coefficient with CL
X_1	0.622	0.651
X_2	-0.622	-0.651
X_3	0.471	0.416
X_4	-0.804	-0.803
X_5	0.861	0.865
X_6	-0.004	0.018
X_7	0.323	0.289
X_8	0.068	0.046

Table 3. Correlation matrix of the input variables

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
X_1	1	-1	-0.256	-0.871	0.869	0	0	0
X_2	-1	1	0.256	0.871	-0.869	0	0	0
X_3	−0.256	0.256	1	−0.193	0.221	0	0	0
X_4	−0.871	0.871	−0.193	1	−0.937	0	0	0
X_5	0.869	−0.869	0.221	−0.937	1	0	0	0
X_6	0	0	0	0	0	1	0	0
X_7	0	0	0	0	0	0	1	0.188
X_8	0	0	0	0	0	0	0.188	1

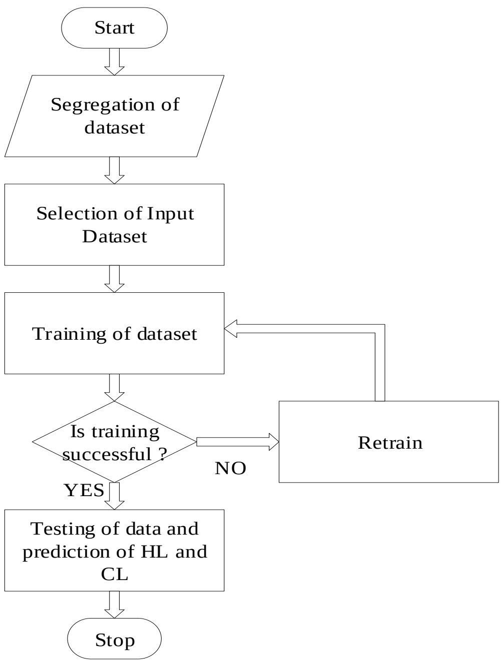


Figure 6. Flow chart of ANN

The correlation coefficient matrix for all of the variables are presented in Table 2. There is a close connection between the first five input variables and the first five output variables. The rank correlations between the input variables are presented in Table 3, which can also be referred to as the covariance matrix. Assuming that the volume of the buildings remains the same, the findings suggest that the relative compactness, denoted by X_1, and the surface area, denoted by X_2, are inversely related to one another. There is a strong correlation between the input variables, such as X_4 (roof area) and X_5 (height). The sign and the magnitude of the correlation coefficient both point to the fact that these variables almost have an inverse proportional relationship with one another (-0.937). All of the values, just like in the correlation table, fall within the range of -1 to +1, which indicates that they are appropriate.

4. SOLUTION METHODOLOGY

After finding out the correlation between the sample data, the dataset is trained, tested and validated with a ratio of 70:15:15 [11,12]. It is essential to separate the data into a training set and a testing set before attempting to understand the dependence of a data set. This will prevent the data from being over fit. Then the outputs such as heating load and cooling load are collected and compared with the target output for error analysis. The flowchart of this algorithm is shown in Figure.6.

5. RESULT ANALYSIS

The learning rate is a hyper parameter that determines how significantly the model should be modified in order to account for the newly estimated mistake each time the weights of the model are changed. Therefore, it is of the utmost importance to have an intuition to know how to analyse its effects on model performance. Mathematical operations known as activation functions are carried out inside an artificial neuron to determine whether or not the neuron should be activated, also known as switched on. An artificial neuron generates the final output of an activation function by first taking a linear combination of the inputs from the neurons in the layer below it, and then applying that activation function to that linear combination. It determines whether or not a certain neuron plays a role for a given collection of inputs (whether or not the neuron should be engaged) and, if it does play a role, how significant of a part it plays in the output if it is active (magnitude value of the output of activation function).

In this instance, the training of each ANN is accomplished on the basis of the training algorithm. Tansig is believed to be the output layer activation function for cascaded forward back propagation neural networks as well as Elman back propagation neural networks because it produces fewer error for these two types of neural networks. In a similar vein, purelin is taken into consideration for use as an output layer activation function in feed-forward neural networks due to the fact that this yields a lower error rate. Step-by-step adjustments to the performance goal and learning rate produce a variety of comparative studies that may be compared to one another. The initialization of the weights with arbitrary values and the updating of the bias have been carried out in accordance with the Levenberg-Marquardt (trainlm) training optimization, the Bayesian regularisation (trainbr) training algorithms, and the gradient descent momentum (traingdx) algorithm. To approach second-order training speed without computing, the Levenberg-Marquardt algorithm is utilised. This is accomplished by taking into consideration The following are some of the ways that you can get your weight updated:

$$w^{k+1} = w^k - (J^{k^T} J^k \mu \bar{I})^{-1} J^{k^T} e^k \tag{18}$$

Where,
W^k is weight and bias matrix at the k^th iteration
I is identity matrix
μ is the combination coefficient ($\mu>0$)
J is Jacobian matrix
e is error

Similarly, Bayesian regularization modifies the linear combination to get a resulting network with good generalization qualities. The Bayesian approach can be represented by the posterior distribution of weights (w) in terms of β, D and M is

$$P(w|D,\alpha,\beta,M)\frac{P(D|w,\beta,M)P(w|\alpha,M)}{P(D|\alpha,\beta,M)} \tag{19}$$

Where,
$P(w|D,\alpha,\beta,M)$ is posterior probability of w
$P(D|w,\beta,M)$ is likelihood function of w
$P(w|\alpha,M)$ is probability of observing the data given w
$P(D|\alpha,\beta,M)$ is normalization factor

Gradient descent is an optimization algorithm for finding local minima of a differentiable function and determining values of function parameters that minimizes the cost function. The gradient descent algorithm can be represented as follows:

$$X = X - lr * \frac{d}{dX} f(X) \tag{20}$$

Where X is input, $F(X)$ is output based on X and lr is learning rate

The accuracy of these models is investigated with the help of performance indicators MAE , MSE , MRE and MAPE .

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{21}$$

$$MSE = \frac{\sum_{i=1}^{n} |y_i - x_i|^2}{n} \tag{22}$$

$$MRE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n x_i} \tag{23}$$

$$MAPE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n x_i} \times 100 \tag{24}$$

Where y_i is Target value, x_i is True Value and n is number of samples

5.1 Performance metrics by varying types of neural networks

Here the number of hidden layers is assumed to be three for all of the 3 proposed networks, with twenty neurons distributed throughout each layer. The Trainlm function is not only the most efficient back propagation algorithm, but it also modifies the weights and biases in accordance with the Levenberg-Marquardt optimum. Training is considered to be the standard training algorithm that applies to all of the networks. For all of the networks, the performance target is set to 10e-5, and the learning rate is set to 0.01. Analysis has been carried out based on performance metrics such as MAE, MSE, MRE, and MAPE. A feed-forward network is optimal for using the linear transfer function known as purelin, which produces accurate results. Purelin is used as an output layer activation function for feed forward because of this reason. In a similar vein, tansig is a hyperbolic tangent sigmoid transfer function, and it is recommended for usage in situations in which speed is a primary consideration. Therefore, for both cascaded forward back propagation and Elman back propagation, the tansig function is used as the output layer activation function. Table 4 presents an overview of the activities carried out by each of these three networks. The tansig–tansig–tansig transfer function is selected as the optimum network with the fewest errors out of all of the cascaded forward-back propagation networks with 20–20–2 neurons, since it has the highest likelihood of producing accurate results.

Table 4. Performance by varying neural networks

ANN	Heating Load				Cooling Load			
	MAE	MSE	MRE	MAPE (%)	MAE	MSE	MRE	MAPE (%)
Feed forward	0.1248	0.033	0.0075	0.75	0.1686	0.0611	0.0087	0.87
Cascaded forward back propagation	0.1023	0.0326	0.0058	0.58	0.1406	0.0473	0.0072	0.72
Elman back propagation	0.2023	0.0819	0.0106	1.06	0.2471	0.1193	0.0117	1.17

Table 5. Performance by varying number of neurons

ANN	Number of neurons	Heating Load				Cooling Load			
		MAE	MSE	MRE	MAPE (%)	MAE	MSE	MRE	MAPE (%)
Cascaded forward back propagation	30-30-30	0.1294	0.0436	0.0073	0.73	0.1805	0.1076	0.0092	0.92
	20-20-20	0.1023	0.0326	0.0058	0.58	0.1406	0.0473	0.0072	0.72
	10-10-10	0.1599	0.0541	0.0086	0.86	0.1784	0.0734	0.0086	0.86

5.2 Performance metrics by varying number of neurons

In order to evaluate the suggested strategy, the number of neurons in each layer is varied on a trial and error basis manner keeping the learning rate, training algorithm, number of layers and performance goal same as before. It is observed that the hidden layers with 20–20–20 neurons give more accurate results, which are clearly shown in Table 5. So, it is considered the optimal network for further analysis.

5.3 Assessment of result by changing no of hidden layers

The proposed method is then put to the test by incorporating a variable number of hidden layers on a trial and error basis manner, keeping the learning rate, training algorithm, number of neurons and performance goal constant as before. The number of hidden layers in a neural network is critical in terms of the network's speed as well as the network's effectiveness. On the other hand, we can observe in Table 6 that the network with three hidden layers produces more accurate results than the network with two hidden layers. So it is considered as the optimal network for further analysis.

Table 6. Performance by varying number of hidden layers

Number of neurons in each hidden layer	Number of hidden layers	Heating Load				Cooling Load			
		MAE	MSE	MRE	MAPE (%)	MAE	MSE	MRE	MAPE (%)
20	1	1.1378	2.0532	0.098	0.98	1.169	3.0956	0.0086	0.86
	2	0.1817	0.0681	0.0088	0.88	0.2016	0.1040	0.0093	0.93
	3	0.1023	0.0326	0.0058	0.58	0.1406	0.0473	0.0072	0.72

5.4 Assessment by altering training algorithm

The proposed method is then tested by changing the training algorithm without changing the other parameters. Trainbr is based on the Bayesian regularization algorithm, trainlm is the Levenberg Marquardt back propagation algorithm, and traingdx is the gradient descent momentum and adaptive learning rate back propagation algorithm. By changing the training algorithm, the result analysis is done, and it is clearly shown in Table 7. Here in this analysis, it is observed that the training algorithm is giving a more accurate result. So, it is considered the optimal algorithm for cascaded forward-back propagation in the 20-20-20-2 network.

Table 7. Performance by varying training algorithm

No of neurons	Training algorithm	Heating Load				Cooling Load			
		MAE	MSE	MRE	MAPE (%)	MAE	MSE	MRE	MAPE (%)
20-20-20	Trainbr	0.1366	0.0532	0.008	0.80	0.169	0.0956	0.0086	0.86
	Trainlm	0.1023	0.0326	0.0058	0.58	0.1406	0.0473	0.0072	0.72
	Traingdx	0.4391	0.3320	0.0247	0.24	0.6702	0.9900	0.0267	2.67

5.5. Performance by varying learning rates

The proposed method is tested to have more accuracy with 3-layer, 20-20-20-2 neurons and a trainable training algorithm. Then the error analysis was done by varying the learning rates between 0.01 and 0.001, without changing the other parameters, which is shown in Table 8. It was observed that the CFBP network converges faster at a learning rate of 0.001 and has a minimum error. So it is considered the optimal network for further analysis.

Table 8. Performance varying learning rate

Training algorithm	Learning rate	Heating Load				Cooling Load			
		MAE	MSE	MRE	MAPE (%)	MAE	MSE	MRE	MAPE (%)
trainlm	0.01	0.1023	0.0326	0.0058	0.58	0.1406	0.0473	0.0072	0.72
	0.001	0.1378	0.0532	0.008	0.8	0.1690	0.0956	0.0086	0.86

5.6 Performance by varying performance goal

From the previous analysis, the cascaded forward-back propagation (CFBP) network was found to be more accurate than the other proposed networks with less MAE, MSE, MRE, and MAPE at a learning rate of 0.01. Then again, this method is tested by taking different performance goals without varying other parameters, as given in Table 9. Tansig is the output layer activation function in this network, which has three hidden layers and twenty-two neurons. It was observed that for performance goal 10e-7, the proposed network shows more accuracy than others.

Table 9. Network performance by varying performance goal

Learning Rate	Performance goal	Heating Load				Cooling Load			
		MAE	MSE	MRE	MAPE (%)	MAE	MSE	MRE	MAPE (%)
0.01	10e-7	0.0800	0.0336	0.0041	0.41	0.0955	0.0406	0.0043	0.43
	10e-6	0.0865	0.0367	0.0046	0.46	0.0956	0.0587	0.0045	0.45
	10e-5	0.1023	0.0326	0.0058	0.58	0.1406	0.0473	0.0072	0.72

6. ERROR ANALYSIS

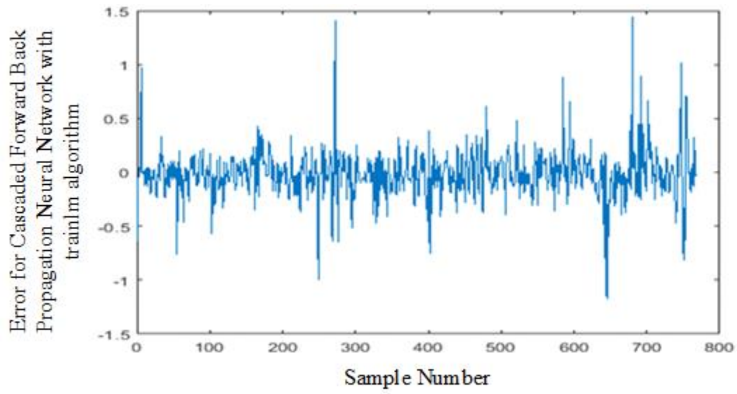
The result analysis is done in terms of different performance indices for all three neural networks, and the relationship between absolute error and the number of sample data is represented here. A comparative analysis of the absolute error and sample numbers for all the proposed ANNs is shown in the following figures. From Figure 7, it can be observed that the maximum heating load error for the cascaded forward back propagation network trained with the trainlm algorithm occurs at instance 680 with an error of 0.7474 and the minimum at instance 645 with an error of -0.5877. For the cascaded forward back propagation network trained with the trainbr algorithm, the maximum occurs at instance 272 with an error of 0.9275 and the minimum at instance 268 with an error of -2.4160. For the feed forward network trained with the trainlm algorithm, maximum occurs at instance 272 with an error of 2.4461 and the minimum at instance 752 with an error of -1.8467 and for Elman back propagation network trained with the trainlm algorithm, maximum occurs at instance 272 with an error of 2.4108 and the minimum at instance 161 with an error of -1.8286.

Similarly, from Figure 8, it can be observed that the maximum cooling load error for a cascaded forward back propagation network trained with the trainlm algorithm occurs at instance 680 with an error of 0.8441 and the minimum at instance 100 with an error of -0.4946. For a cascaded forward back propagation network trained with the trainbr algorithm, the maximum occurs at instance 684 with an error of 1.9115 and the minimum at instance 113 with an error of -1.5888, for Feed forward network trained with trainlm algorithm maximum occurs at instance 596 with an error of 1.3706 and the minimum at instance 113 with an error of -2.1765 and for Elman back propagation network trained with the trainlm algorithm maximum occurs at instance 136 with an error of 1.7626 and the minimum at instance 593 with an error of -1.6566.

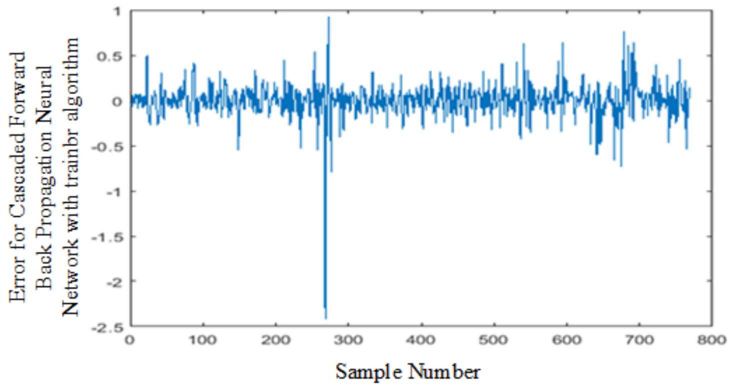
From the graphical analysis of Figure 8, cascaded forward-back propagation is found to be the best method among the three proposed methods. It has fewer errors than the others in terms of MAE, MSE, MRE, and MAPE. For MAE, MSE, MRE and MAPE, the values were obtained as 0.0800, 0.0336, 0.0041, and 0.41% for heating load, and 0.0955, 0.0406, 0.0043, and 0.43% for cooling load, respectively.

In Figure 9, the variation of target output and network output for both the heating load and cooling load for the 768 dataset is given, where we can see that the variation of output is very small.

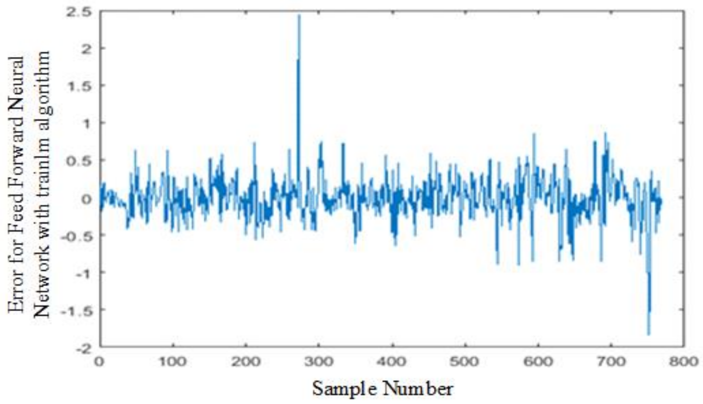
In this analysis, the cascaded forward-back propagation network with the trainlm algorithm is found to be the most efficient neural network with very little error in the heating and cooling load analysis of buildings. Figure.10 represents the performance of the network in terms of gradient, validation check and mu. It is the training state of the network. In this case gradient is 0.0010582 at 140 epochs, which means finding the best fit for the given set of training dataset, and number of validation check is 100 at 140 epochs with Mu=1e-07, at 140 epoch. When we talk about having 100 validation errors here, we imply that when we have simultaneously produced 100 validation check mistakes, then we will halt training. Validation check errors indicate that the dataset in question has certain flaws—perhaps some instances are not comprehensible by the training algorithm, for example—which means that validation check errors can be generated as a direct result of dataset flaws. In this context, "mu" refers to the momentum constant or momentum parameter that is incorporated into the weight update expression in order to circumvent the issue of reaching a local minimum. Between 0 and 1, mu can take on any value.



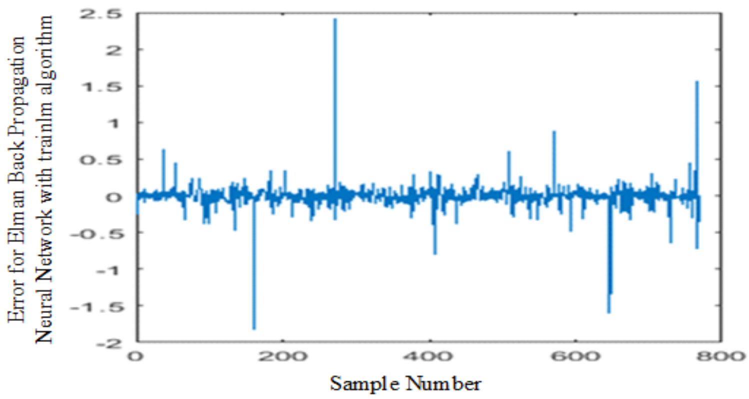
(a)



(b)

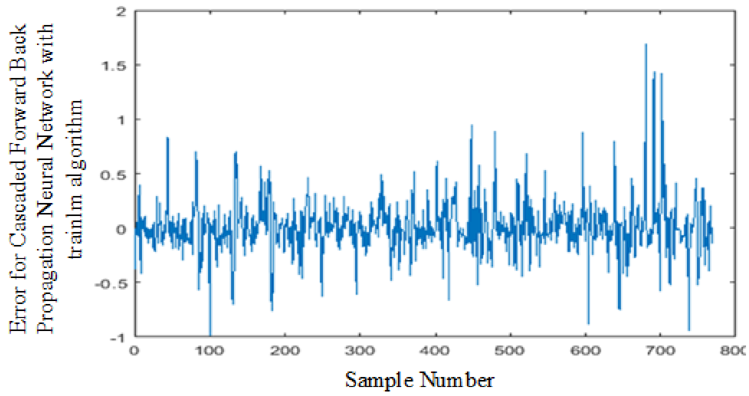


(c)

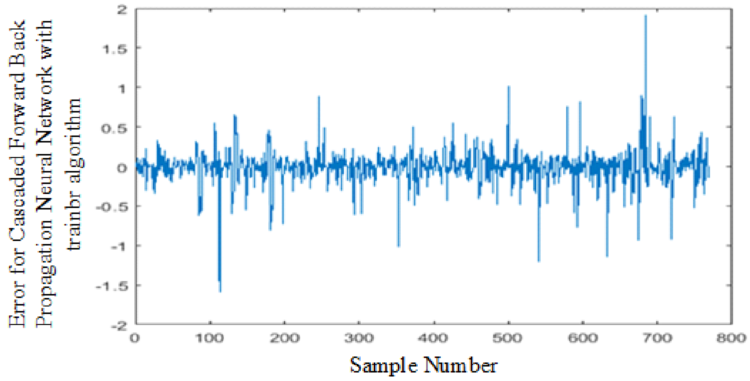


(d)

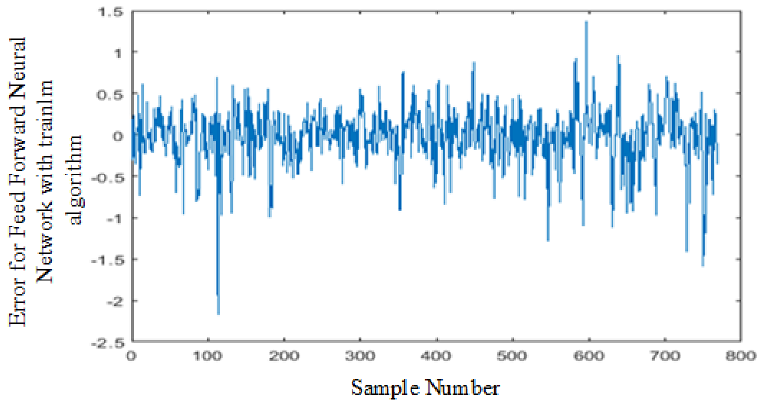
Figure .7 Absolute error vs Sample number of different ANNs for Heating Loads



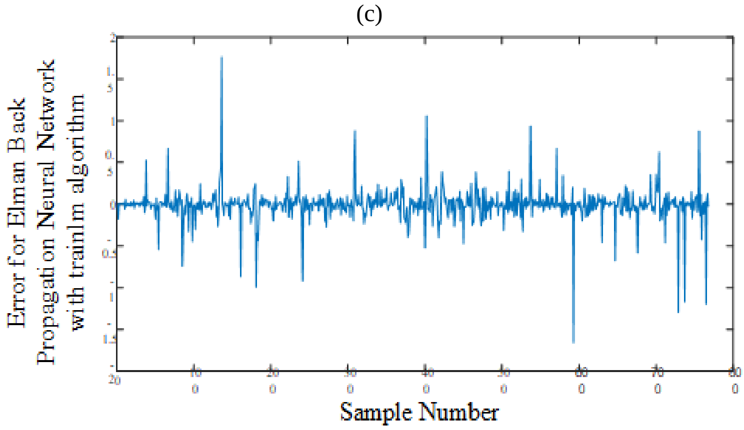
(a)



(b)

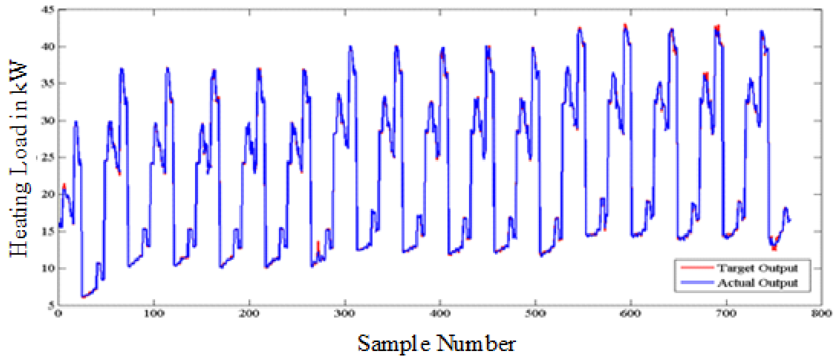


(c)

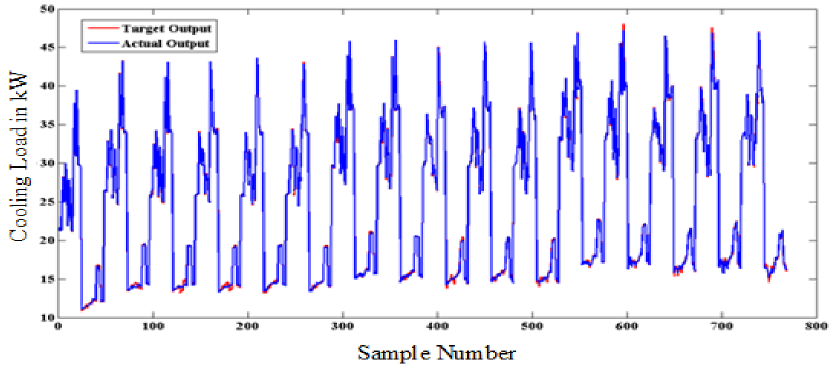


(d)

Figure 8. Absolute error vs Sample number of different ANNs for Cooling Load



(a)



(b)

Figure 9. Target output and network output variation for Cascaded forward back propagation network (a) Heating load (b) Cooling load

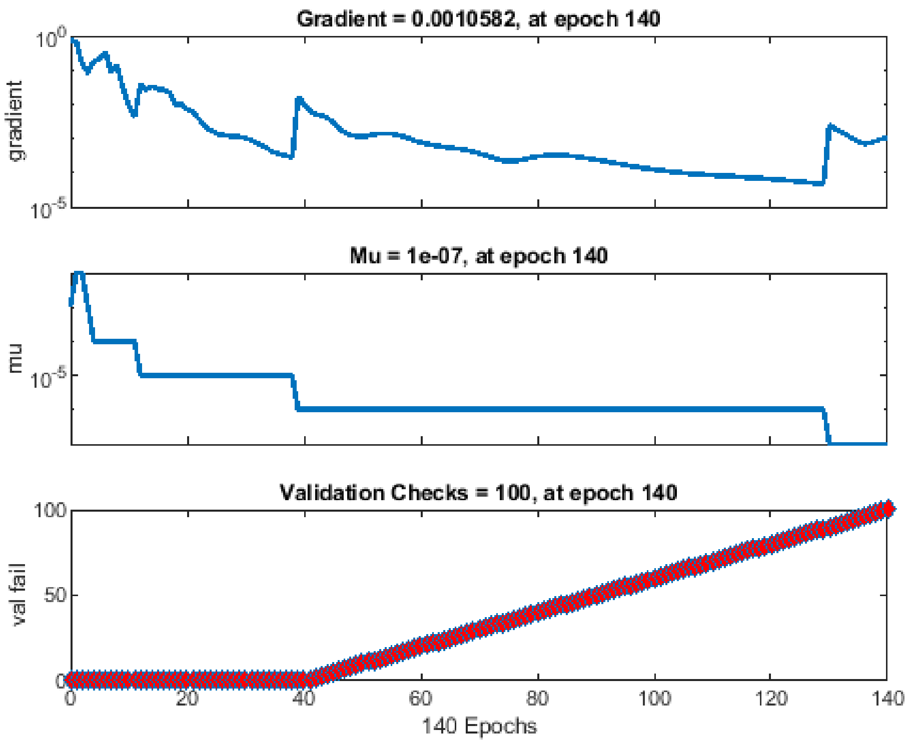


Figure 10. Performance plot of training

According to the findings presented in Figure 11, the best validation that could be attained was 0.0005969 at epoch 40. In order to monitor the level of training that has been completed, it is necessary to plot the training, validation, and test mistakes. It is clear to see that the training was terminated after 140 iterations or epochs due to a rise in the validation error after that point in time. The fact that the test error and the validation set error share comparable features, in addition to the absence of any major over-fitting in this instance, both of the relevant figures point to the fact that the result that was produced is plausible. When the other line in the figure, which is represented by the best line, lies on or very close to this line, we will know for certain that the training has been carried out effectively. If any of the three lines (training, validation, and testing) meet or pass near the best line (dotted), this indicates that convergence has been achieved; if this is not the case, then the network has to be retrained. The total number of mistakes that have occurred at any given instant is presented in Figure.12 and Figure.13, respectively. These figures pertain to the heating load and the cooling load.

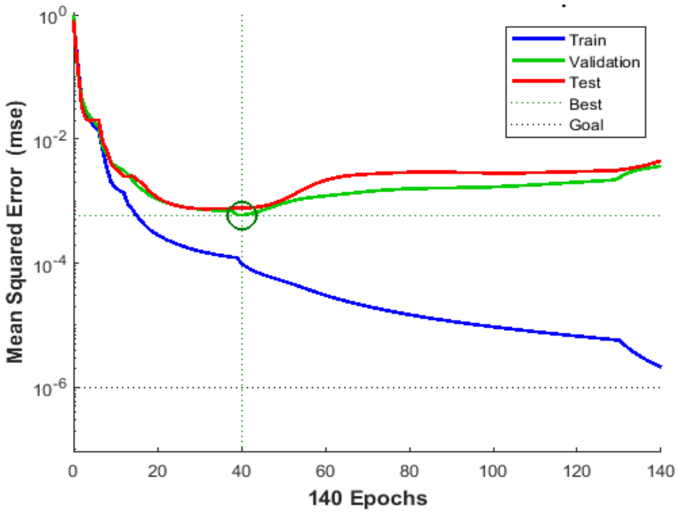


Figure 11. Validation Plot

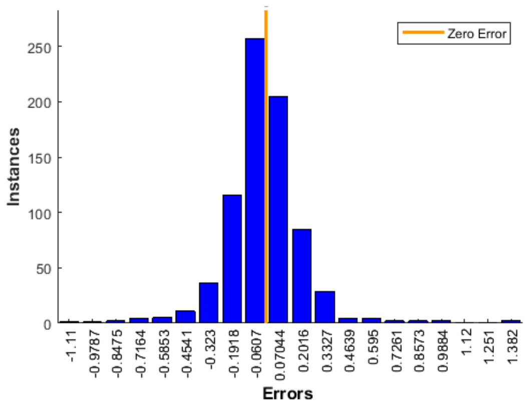


Figure 12. Error plot for heating load

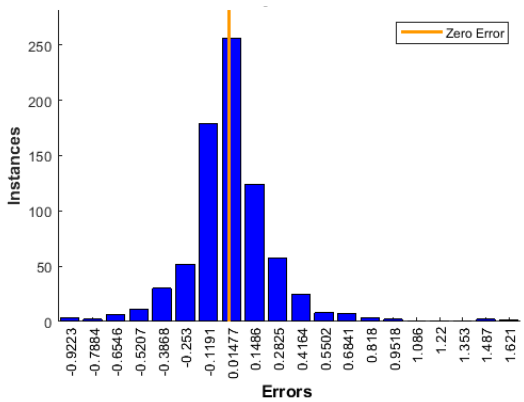


Figure 13. Error plot for cooling load

7. VALIDATION OF RESULT

The method that has been proposed can also be contrasted with several other methods. For the purpose of error analysis, a variety of performance evaluators such as MAE, MSE, MRE, and MAPE are taken into consideration for application. For their forecast of heating and cooling loads, the authors of [3] utilised a technique called iteratively reweighted least squares (IRLS), and they also used a technique called random forests (RF). It has been discovered that the MAE, MSE, and MRE for heating load when using IRLS is 2.14, 9.87, 10.09, while when using RF it is 0.51, 1.03, 9.41, and that the MAE, MSE, and MRE for cooling load when using IRLS is 2.21, 11.46, 2.18, while when using RF it is 1.42, 6.59, 4.62. These results can be found in the table below. The authors have used SVR and MLP [13] to predict the amount of heating and cooling load required for a building using the same data. Both the MAE and the MSE will be used as performance evaluators throughout this article. For the heating load, the results found here are 0.778, 0.7838 in support vector regression (SVR) and 0.4118, 0.2335 in multilayer perceptron (MLP). For the cooling load, the results found here are 1.4762, 3.024 in SVR and 2.0973, 6.896 in MLP. According to [14], the back propagation approach provides MAE, MSE, and MRE values of 0.1001, 0.0335, and 0.0053 for the heating load, and it provides values of 0.1254, 0.763, and 0.0062 for the cooling load. In [11], three different machine learning algorithms, namely MLR, KNN, and SVR, were put through their paces using the same dataset that was utilised in the present investigation's analysis. After applying MLR and SVR, the values of RMSE for both output variables Y1 and Y2 range from 3.13 to 4.22, however when KNN is applied, the values are lower, coming in at 2.86 for Y1 and 2.25 for Y2. In a similar vein, the MSE values for KNN are significantly lower than those for MLR and SVR. Also, while using MLR and SVR, the MAE values range between 2.25 and 3.19, however when using KNN, the values are between 1.96 and 1.54. R The squared values for KNN are higher (0.90 and 0.94), compared to those for MLR (0.87 and 0.83) and SVR (0.83 and 0.87). (0.76 and 0.84). The results obtained by using the KNN method are superior to those obtained using either of the other two algorithms. When using multiple linear regression, the accuracy of forecasting the heating load and the cooling load, respectively, was 88.59% and 85.26%. When using support vector regression, the accuracy was 82.38% and 89.32%. When using K-nearest neighbours, the accuracy was 91.91% and 94.47%. When compared to this, the answer that we obtain using the suggested method is 95% more accurate for the heating load, and it is 94% more accurate when calculating the cooling load. The error analysis is presented in Table 10, and it was discovered that the proposed method of cascaded forward-back propagation has an improved MAE, MSE, and MRE when compared to the methods that were discussed previously. Specifically, the MAE for the heating load is 0.0800, and the MSE and MRE for the cooling load are 0.0041 and 0.0955, respectively.

Table 10. Comparison with other methods

Proposed algorithm and reference	MAE		MSE		MRE	
	HL	CL	HL	CL	HL	CL
IRLS [3]	2.14	2.21	9.87	11.46	10.09	2.18
RF [3]	0.51	1.42	1.03	6.59	9.41	4.62
SVR[13]	0.778	1.4762	0.7838	3.024	-	-
MLP[13]	0.4118	2.0973	0.2335	6.896	-	-
Back propagation [14]	0.1000	0.1254	0.0335	0.763	0.0053	0.0062
KNN [11]	1.96	1.54	8.2	5.07	8.17	5.06
Cascaded forward back propagation	0.0800	0.0955	0.0336	0.0406	0.0041	0.0043

8. CONCLUSION AND FUTURE SCOPE

In this paper, an ANN model has been proposed to forecast the heating and cooling load of an HVAC system in a residential building [3] using eight input datasets with a total of 768 different building samples. The MAE, MSE, MRE, and MAPE are found to be 0.0800, 0.0336, 0.0041, and 0.41% for heating load and 0.955, 0.0406, 0.0043, and 0.43% for cooling load respectively, when the learning rate is 0.01 and the performance goal is $10e^{-7}$. It is revealed that the cascaded forward-back propagation network with three layered 20-20-20-2 neurons is the most efficient network and achieves the best results. In future this proposed network can be implemented for analysis of energy consumption and cost minimization in real time data set.

REFERENCES

[1] www.iea.org
[2] A. Yezioro, B. Dong and F. Leite, "An applied artificial intelligence approach towards assessing building performance simulation tools", *Energy and Buildings*, Vol. 40, pp. 612– 620, 2008.
[3] A. Tsanas and A. Xifara "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools" *Energy and Buildings*. Vol. 49, pp. 560–567,2012.
[4] Y. Sonmez, U. Guvenc, H. T. Kahraman and C. Yilmaz, "A comperative study on novel machine learning algorithms for estimation of energy performance of residential buildings," *Smart Grid Congress and Fair (ICSG)*, Vol. 22, pp. 178–188, 2015.
[5] R. Yao, B. Li and K. Steemers, "Energy policy and standard for built environment in China", *Renewable Energy*, Vol. 30, pp. 1973–1988, 2005.

[6] Z. Guo , H. Moayedi , L. K. Foong and M. Bahiraei , "Optimal Modification of Heating, Ventilation, and Air Conditioning System Performances in Residential Buildings Using the Integration of Metaheuristic Optimization and Neural Computing", *Energy & Buildings*, Vol.214, 2020.
[7] https://in.mathworks.com/
[8] S. M. Hong, G. Paterson, D. Mumovic, and P. Steadman, "Improved benchmarking comparability for energy consumption in schools," *Build. Res. Inf.*, vol. 42, pp. 47–61.
[9] https://www.researchgate.net/figure/Schematic-of-the-multilayer-feed-forward-neural-network-proposed-to-model-the-chaotic_fig1_275334508
[10] U.S. Department of Energy, E.E.a.R.E.O., ''Building Technology Program, Net- Zero Energy Commercial Building Initiative",Commercial Building Benchmark Models. Available from: http://www1.eere.energy.gov/buildings/commercial_initiative/benchmarkmodels.htmli. 2009.
[11] M. Goyal, M.Pandey, "A Systematic Analysis for Energy Performance Predictions in Residential Buildings Using Ensemble Learning", *Arabian Journal for Science and Engineering*. Vol. 46(4), pp. 3155-68, 2021.
[12] https://scholarworks.utep.edu/cgi/viewcontent.cgi?article=2202&context=cs_techrep
[13] J.-S. Chou and D.-K. Bui, "Modeling heating and cooling loads by artificial intelligence for energy-efficient building design," *Energy Build*., Vol. 82, pp. 437–446, 2014.
[14] S. Das, A. Swetapadma, C. Panigrahi and A.Y. Abdelaziz, "Improved Method for Approximation of Heating and Cooling Load in Urban Buildings for Energy Performance Enhancement", *Electric Power Components and Systems*, 48:4-5, 436-446, 2020.

BIOGRAPHY OF AUTHORS

Lohit Kumar Sahoo is a presently working as Assistant Professor at Trident Academy of Technology in the department of Electrical and Electronics Engineering. The author has completed his M.tech from IIT-Kharagpur in 2013. The author has 14 years of experience in the field of education and consultancy work.

Mitali Ray has completed her Ph.D in Energy Science at KIIT deemed to be University in the year of 2022. Presently she is working in the department of Skill Development & Technical Education Odisha, India.

Sampurna Panda has completed her Ph.D in Solar Energy Technology at Poornima University in the year of 2023. Presently she is working as Assistant Professor at ITM University, Gwalior, Madhya Pradesh, India.

Subrat Ranjan Kabat) has completed his Ph.D in Power System at KIIT deemed to be University in the year of 2022. Presently he is working as Assistant Professor at Radhakrishna Institute of Technology and Engineering, Bhubaneswar, Odisha, India.

Smita Dash has completed her Ph.D in Management in 2022 from Kalinga University,Naya Raipur,Chhatisgara in Topic-Importance of E-banking and CRM in banks. Presently she is working as Assistant Professor (Finance) at United School of Business Management, Bhubaneswar, Odisha, India