

# **Estimate of Roomsize for Best Performance of a Cooling System: Neural Network/Matlab Approach.**

**ENGR. DR. EFOSA OBASEKI**

*DEPARTMENT OF MECHANICAL ENGINEERING, FEDERAL POLYTECHNIC NEKEDE, OWERRI.*

*PHONE NUMBER: 07064371681*

**AGULANNA VICTOR OKECHUKWU**

*DEPARTMENT OF MECHANICAL ENGINEERING, IMO STATE POLYTECHNIC.*

**EWURUM TENNISON. I**

*DEPARTMENT OF MECHANICAL ENGINEERING, FEDERAL POLYTECHNIC NEKEDE, OWERRI.*

---

## **ABSTRACT**

*The study, estimate of room size for best performance of a cooling system, using Neural Network/MATLAB approach was successfully carried out. The researchers adopted a survey design approach and randomly selected buildings within Owerri Municipal. Measurements of room dimensions were taken and the values were used to determine room volume, infiltration, ventilation, room area and cooling system capacity. Neural Network/MATLAB was used to analyze the tabular data. Training of room volume and infiltration data using levenberg marquardt algorithm at 70% training data, 15% test data and 15% validation data respectively revealed that the best performance level was 106.0012 at epoch 1 and this matches an approximate room size of 128.479m<sup>3</sup> with infiltration of 2.2489 m<sup>3</sup>. Cooling capacity best performance level was found to be 8.7406Tons at epoch 1 and this corresponds to an approximate room volume of 308.278m<sup>3</sup>. Also, the regression coefficient of 1 from the graphs indicated that there is a close relationship between room size, infiltration and cooling system capacity. Therefore, the estimated value of room size (volume) for best performance of a cooling system ranges from 128.479m<sup>3</sup> to 308.278m<sup>3</sup>. MATLAB script/function was also generated for viewing the neural network model, performance graphs and model simulation.*

**Keywords:** *Cooling system, neural network/MATLAB, algorithm, training data, validation data, test data, cooling capacity, infiltration, room volume, epoch, model, simulation, script.*

---

Date of Submission: 04-11-2022

Date of acceptance: 16-11-2022

---

## **I. INTRODUCTION**

### **Background of the Study**

Jackravut and Nutthaphong (2016) stated that thermal characteristics of a building, to a greater extent depend on room size, infiltrations, building materials, wall orientation, window and door sizes, etc. The energy consumption from a cooling system (air conditioner), also depends on room air volume.

The selection and installation of the most appropriate air handling and processing units to support indoor environment requirements are crucial to maintaining a healthy and comfortable indoor environment. Due to increase energy and environmental concerns, combined with a demand for better indoor thermal comfort and air quality. The maintenance of acceptable indoor air quality is an important consideration in any built environment, as it prevents the occurrence of detrimental health effects in the building's occupants. The maintenance of a healthy and comfortable indoor environment under different outdoor conditions is one of the most energy-intensive requirements of any built environment ( Napoleon, Odinah & Takao, 2020).

Desai(2012) defined infiltration as entering of outside air into the room to be air conditioned. He further stated that infiltration depends on height of the room, length of the room, width of the room and number of air changes per hour. The term one infiltration, means the equivalent volume of outside air that leaks into the room space to be air cooled.

There are no doubts that infiltration increases both sensible and latent heat loads on the cooling system. The study also claimed that the value of load on cooling system due to infiltration/room size is 49.1 times the value of room volume under one unit difference of relative humidity. Hence, this research aimed at studying estimate of room size for best performance of a cooling system, using neural network/matlab approach.

### **Statement of the Problem**

Reviewed studies on performance of cooling system, done by the author, consistently proved that building room size affects the performance of cooling systems. It was also discovered that increasing room volume, tends to increase infiltration and hence, sensible and latent heat loads that increases energy consumption. This is making it very pertinent to determine an estimate value of room size to achieve best performance. Increasing room volume will also increase infiltration which affects people comfort and raises energy consumption of cooling system. As well, decrease in room volume, decreases infiltration, minimizes energy consumption but may not necessarily assured human comfort or best performance. It is on this note that the researcher aims at determining the estimate of room size for best performance of a cooling system, using neural network/matlab approach.

### **Purpose of the Study**

The purpose of this study is to estimate room size for best performance of a cooling system using MATLAB approach.

### **Significance of the Study**

The result of this study will be beneficial to building engineers and air conditioner manufacturing engineers in the following ways:

1. This study will help building engineers to make use of appropriate dimensions of room sizes during design and construction of buildings to ensure effective cooling.
2. Air conditioner manufacturers may deploy the result of the study to manufacture cooling systems that will match a giving room size.

### **Scope of the Study**

The building dimensions used in this work are taken from a buildings within south east of Nigeria and hence, the result of the study may not be applicable to other parts of Nigeria or Western World. Therefore, the study will be applicable to all buildings within South-East of Nigeria.

## **II. Review of Related Literature**

The review of the related literature is discussed under the following: conceptual framework and empirical framework.

### **Conceptual Frame**

#### **Concept of Cooling System**

Cooling systems can be used to create a healthy indoor thermal environment and provide high-quality air while consuming minimal amounts of energy. Thus, cooling systems are crucial when considering present concerns about energy

use and the environment. Cooling systems process recirculated indoor air and outdoor air to produce an acceptable level of throughout an indoor environment. The conditioning of recirculated air is economical but detrimental to system; meanwhile, the conditioning of fresh outdoor air improves cooling system but consumes a lot of energy. However, a combination of other devices, UV lights, can control air biochemical contents, enable the conditioning of recirculated air to be safely and economically reintroduced in indoor environment. Cooling systems normally either process both recirculated indoor air and fresh outdoor air mixed at proportions based on cooling requirements with the aim to reduce energy consumption. The use of different cooling systems has increased as the demand for quality indoor environment increases.

#### **Concept of Air Conditioner**

Air conditioning (often referred to as air on, AC or A/C) is the process of altering the properties of air (primarily purity, movement, humidity and temperature) to favorable conditions, typically with the aim of distributing the conditioned air to an occupied space is to improve comfort. In the most general sense, air conditioning can refer to any form of technology humidification, de-humidification, heating, cooling, cleaning, ventilation, or air movement that modifies the condition of air. In general, the air conditioner is a device (most commonly a home appliance or automobile system) that lowers the air temperature. The cooling is most done using a simple refrigeration cycle, but sometimes the evaporation is used, commonly for the comfort of cooling in -buildings and motor vehicles. In construction, a complete system of heating, ventilation and air conditioning is referred to as "HVAC" (Desai,2012).

#### **Concept of cooling load**

The two main components of a cooling load imposed on an air conditioning plant operating during hot weather are as follows: Sensible heat gain: When there is a direct addition of heat to the enclosed space, a gain

in the sensible heat is said to occur. Latent heat gain: When there is an addition of water vapor to the air of enclosed space, a gain in latent heat is said to occur (Prathibha, Kodliwad, Busi, & Naga, 2017).

### **Concept of Cooling System components**

The major components of a cooling system are combined based on different designs and specifications, as this improves the performance of specific applications. The simplest system is the simple loop, which consists of a single compressor, a single outdoor heat exchanger, an indoor heat exchanger, and an expansion valve. The single loop cooling system is typical in small units, such as window-type or split-type systems; as such systems must be compact to fit into windows or other small areas. On the other hand, complex cooling systems involve multiple compressors, multiple indoor units, and multiple outdoor units. The cooling system with multiple outdoor units and indoor units is intended for large thermal loading requirements.

### **Empirical Framework**

Jackravut and Nutthaphong (2016) studied a simplified air conditioning systems model with energy management. The study adopted a simulink MATLAB model and an experimental model. They concluded that the result from MATLAB simulink model and experiment model are consistent. Prathibha et al. (2017) studied design of air conditioning system for residential/office building. The classified air conditioning system operation into part A, basic principles of thermodynamics and part B, cooling load calculation. They concluded that accurate cooling load calculations, determine effectiveness of a cooling system. Napoleon et al. (2020) investigated review of the advances and applications of variable refrigerant flow heating, ventilating, and air-conditioning systems for improving indoor thermal comfort and air quality. They concluded that heating, ventilating and air conditioning system can provide a healthy indoor thermal environment with good air quality. Desai (2012) studied refrigeration and air conditioning for engineers. The results of his studies showed that infiltration is 46.1 times the value of room volume under one unit of relative humidity.

## **III. Research Methodology**

The research methodology is discussed under the following headings: research design, area of the study, sample and sampling techniques, method of data collection, validation of data and method of data analysis.

### **Research Design**

The study followed a survey design approach. Survey design is one in which a group of buildings are studied by collecting and analyzing data from only selected buildings that serves as the representative of the entire buildings and findings are generalized. Dimensions from selected buildings would represent the entire buildings within the study area.

### **Area of the Study**

The research was conducted within Owerri Municipal Council which is one the local government areas in Imo State. It is between Owerri North and Owerri West. The representative buildings are located at No. 1036 West layout Prefab Uratta Housing Estate.

### **Method of Data collection**

The researchers personally went to the selected buildings, after obtaining permission from the landlord, measured and recorded the dimensions of room sizes with the aid of measuring tape. Window and door sizes were also recorded. The obtained measurement was used to generate the tabular data that was subjected to MATLAB analysis.

### **Method of Data Analysis**

Neural network/matlab software was used to analyze the tabular data. 15% of the original data serves as a validation data and 15% of the original data also serves as test data and 70% as a train data. Levenberg Marquardt was used as a training algorithm.

### **Design Analysis**

$$\text{Infiltration} = \frac{H \times L \times W \times G}{60} \text{ m}^3/\text{min} \dots\dots(1.0) \text{ Desai (2012)}$$

Where H = height of the room;

L = length of room;

W = width of room;

G = number of air changes/hr

Load due to outside air will be both sensible as well as latent. It is given as below:

$$\text{outside air sensible heat} = 20.43 \times Q_m (t_o - t_i) \dots(1.1)$$

$$\text{outside air latent heat} = 50Q_m(W_o - W_i) \dots(1.2)$$

W and t means relative humidity and temperature respectively and Q is outside air volume flow rate.

$$\text{sensible heat gain, } Q = U \times A \times CLTD \text{ watts} \dots(1.3)$$

U = overall heat transfer coefficient; A = wall area and CLTD = cooling load temperature difference.

$$\text{heat gain through windows} = a \times GLF \dots(1.4)$$

where a = glass area and GLF = glass load factor.

### Presentation of Data Analysis

The tabular data gotten from the study is presented as shown below.

**Table 1.0: below shows room volume, total infiltration, ventilation and room area.**

NAME OF ROOM/SPACE	ROOM	TOTAL INFIL	VENT.	ROOM
	VOL	Q <sub>MT</sub>	Q <sub>vent</sub>	AREA
	(m <sup>3</sup> )	(m <sup>3</sup> )	(m <sup>3</sup> )	(m <sup>2</sup> )
Room 1	61.238	30.896	42	200.364
Room 2	5180.175	16.29	26.46	82.225
Room 3	308.278	6.638	3.36	90.67
Room 4	206.244	14.263	25.2	60.66
Room 5	678.912	24.472	70	199.68
Room 6	40.426	1.424	2.5	11.89
Room 7	52.935	2.779	2.52	15.569
Room 8	73.824	4.361	3.5	21.713
Room 9	54.539	1.653	3.5	16.041
Room 10	660.8308	27.42	112	194.365
Room 11	112.047	2.057	3.5	32.955
Room 12	128.479	2.2489	3.5	37.788
Room 13	208.06	1.456	2.356	61.194

### Matlab Script for neural network model and training performance graph is shown below.

```
% Solve an Autoregression Time-Series Problem with a NAR Neural Network
% Script generated by Neural Time Series app
% Created 11-Jun-2021 12:23:49
%
% This script assumes this variable is defined:
%
% data - feedback time series.

T = tonndata(data,false,false);

% Choose a Training Function
% For a list of all training functions type: help nntain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Nonlinear Autoregressive Network
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narnet(feedbackDelays,hiddenLayerSize,'open',trainFcn);

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer
```

```
% states. Using PREPARETS allows you to keep your original time series data
% unchanged, while easily customizing it for networks with differing
% numbers of delays, with open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,{},{},T);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotresponse(t,y)
%figure, ploterrcorr(e)
%figure, plotinerrcorr(x,e)

% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the outout layer.
netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
view(netc)
[xc,xic,aic,tc] = preparets(netc,{},{},T);
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(net,tc,yc)

% Step-Ahead Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is
% given y(t+1). For some applications such as decision making, it would
% help to have predicted y(t+1) once y(t) is available, but before the
% actual y(t+1) occurs. The network can be made to return its output a
% timestep early by removing one delay so that its minimal tap delay is now
% 0 instead of 1. The new network returns the same outputs as the original
% network, but outputs are shifted left one timestep.
nets = removedelay(net);
nets.name = [net.name ' - Predict One Step Ahead'];
view(nets)
[xs,xis,ais,ts] = preparets(nets,{},{},T);
ys = nets(xs,xis,ais);
stepAheadPerformance = perform(nets,ts,ys)
```

**The matlab function is as below:**

```
[Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 11-Jun-2021 11:35:55.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
% X = 0xTS cell, 0 inputs over TS timesteps
%
% and returns:
% Y = 0xTS cell of 0 outputs over TS timesteps.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

%#ok<*RPMTO>

% ===== NEURAL NETWORK CONSTANTS =====

% ===== SIMULATION =====

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;

% Dimensions
TS = size(X,2); % timesteps

% Allocate Outputs
Y = cell(0,TS);

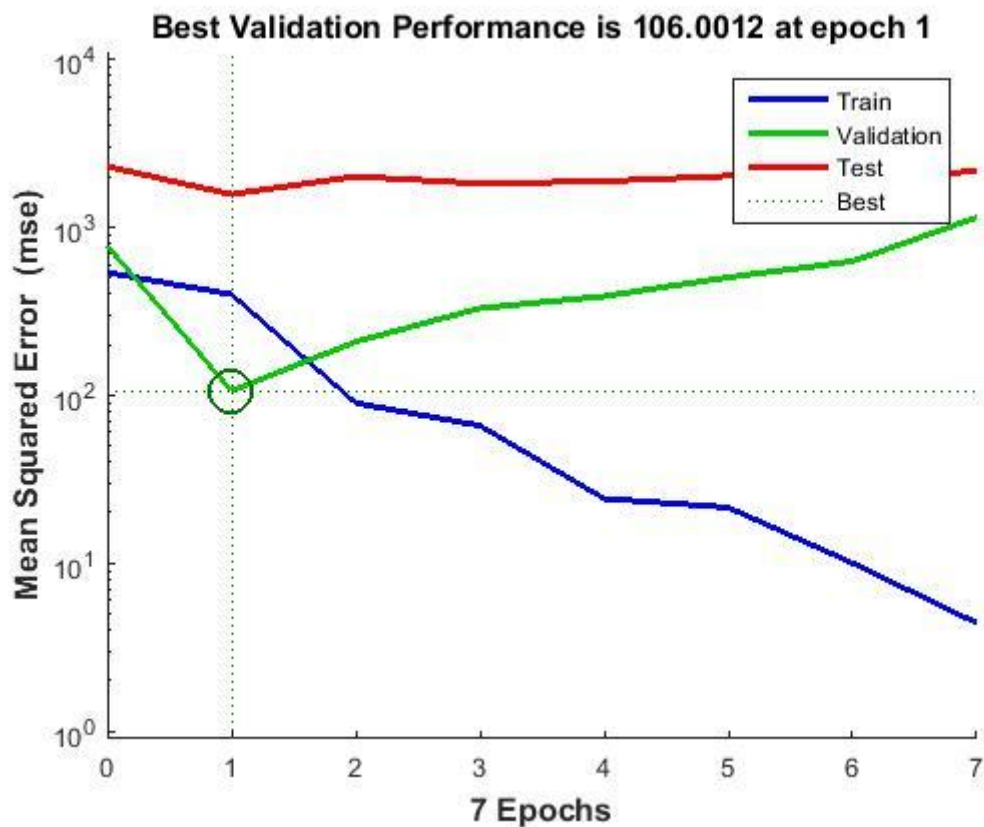
% Time loop
for ts=1:TS

end

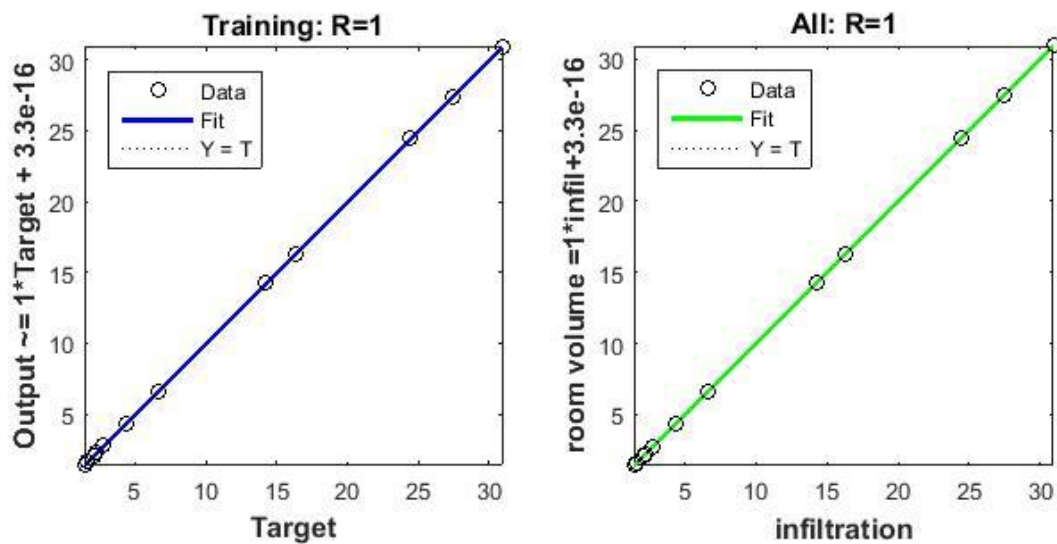
% Final Delay States
Xf = cell(0,0);
Af = cell(0,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end

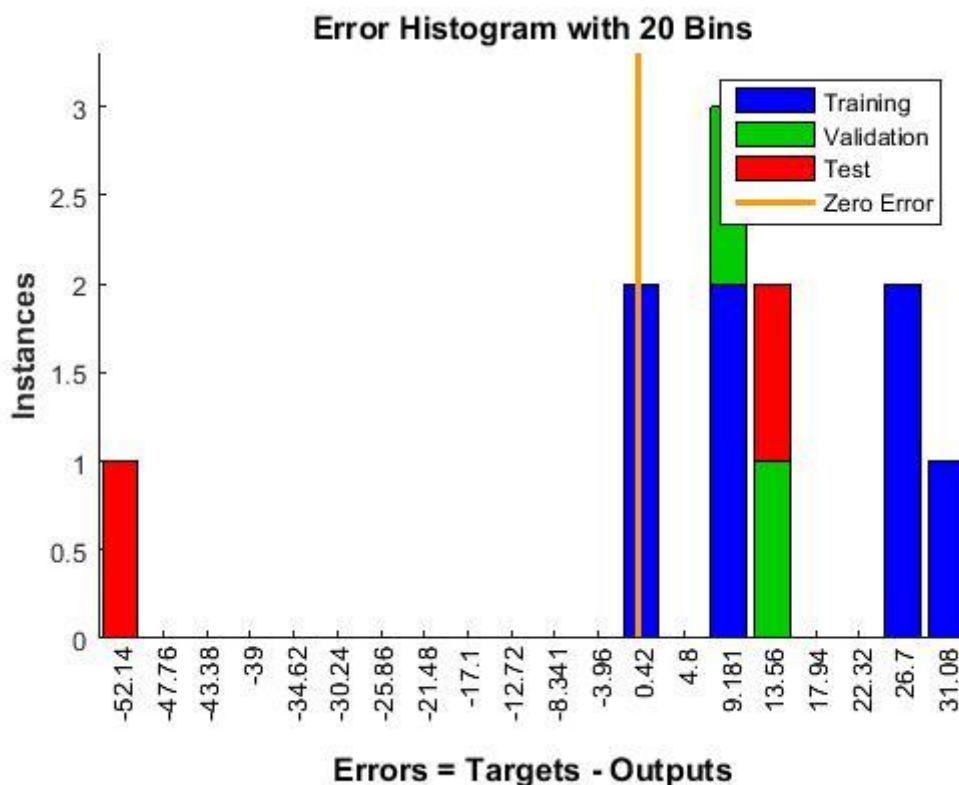
% ===== MODULE FUNCTIONS =====
```



Regression graph of room volume against infiltration is shown below.



The Error histogram is shown below.



**Table 1.1, below shows room volume and capacity of cooling system.**

NAME OF ROOM/SPACE	ROOM	CAPACITY =
	VOL	RTH/3500
	(m <sup>3</sup> )	(Tons)
Room 1	61.238	29.68386
Room 2	5180.175	22.48338
Room 3	308.278	7.278687
Room 4	206.244	20.29576
Room 5	678.912	51.67699
Room 6	40.426	3.412411
Room 7	52.935	2.896382
Room 8	73.824	3.7785
Room 9	54.539	3.573547
Room 10	660.8308	71.50495
Room 11	112.047	4.135946
Room 12	128.479	4.479082
Room 13	208.06	5.632464

**Matlab Script for neural network model and performance training is shown below.**

```
% Solve an Autoregression Time-Series Problem with a NAR Neural Network
% Script generated by Neural Time Series app
% Created 11-Jun-2021 13:29:56
%
% This script assumes this variable is defined:
```



```
%
% data - feedback time series.

T = tonndata(data,false,false);

% Choose a Training Function
% For a list of all training functions type: help nntain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Nonlinear Autoregressive Network
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narnet(feedbackDelays,hiddenLayerSize,'open',trainFcn);

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer
% states. Using PREPARETS allows you to keep your original time series data
% unchanged, while easily customizing it for networks with differing
% numbers of delays, with open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,{},{},T);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotresponse(t,y)
%figure, ploterrcorr(e)
%figure, plotinerrcorr(x,e)

% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the outout layer.
netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
view(netc)
[xc,xic,aic,tc] = preparets(netc,{},{},T);
```

```
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(net,tc,yc)

% Step-Ahead Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is
% given y(t+1). For some applications such as decision making, it would
% help to have predicted y(t+1) once y(t) is available, but before the
% actual y(t+1) occurs. The network can be made to return its output a
% timestep early by removing one delay so that its minimal tap delay is now
% 0 instead of 1. The new network returns the same outputs as the original
% network, but outputs are shifted left one timestep.
nets = removedelay(net);
nets.name = [net.name ' - Predict One Step Ahead'];
view(nets)
[xs,xis,ais,ts] = preparets(nets, {}, {}, T);
ys = nets(xs,xis,ais);
stepAheadPerformance = perform(nets,ts,ys)
```

**Matlab Function for the neural network model and training performance is shown below.**

```
function [Y,Xf,Af] = myNeuralNetworkFunction(X,Xi,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 11-Jun-2021 13:36:37.
%
% [Y,Xf,Af] = myNeuralNetworkFunction(X,Xi,~) takes these arguments:
%
% X = 1xTS cell, 1 inputs over TS timesteps
% Each X{1,ts} = 1xQ matrix, input #1 at timestep ts.
%
% Xi = 1x2 cell 1, initial 2 input delay states.
% Each Xi{1,ts} = 1xQ matrix, initial states for input #1.
%
% Ai = 2x0 cell 2, initial 2 layer delay states.
% Each Ai{1,ts} = 10xQ matrix, initial states for layer #1.
% Each Ai{2,ts} = 1xQ matrix, initial states for layer #2.
%
% and returns:
% Y = 1xTS cell of 1 outputs over TS timesteps.
% Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
% Xf = 1x2 cell 1, final 2 input delay states.
% Each Xf{1,ts} = 1xQ matrix, final states for input #1.
%
% Af = 2x0 cell 2, final 0 layer delay states.
% Each Af{1ts} = 10xQ matrix, final states for layer #1.
% Each Af{2ts} = 1xQ matrix, final states for layer #2.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

%#ok<*RPMTO>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1_xoffset = 2.8963815;
x1_step1_gain = 0.0291508774279283;
x1_step1_ymin = -1;
```

```
% Layer 1
b1 = [4.4271887242357311;-3.4433690077389021;-2.4595492912420731;-
1.4757295747452439;0.49190985824841477;-0.49190985824841477;-
1.4757295747452435;2.4595492912420731;-3.4433690077389016;4.4271887242357311];
IW1_1 = [-3.3721107357915532 -2.8686005622183348;4.1116811365949237
1.6413647464745551;4.4267488985408319 -0.062403439548895663;3.2223014062604727
3.0359139723008912;-3.9022291646603144 2.0910780823475883;-2.2492062683591918
3.813275647310816;-1.7183935128979801 4.0800886920298858;3.2537069569169561 -
3.0022310101839587;-3.8843513705091453 2.1241032061610658;2.6151670899169344 -
3.5722403463114563];

% Layer 2
b2 = -0.63415506117017206;
LW2_1 = [-0.93891810739072668 0.48814852073492476 4.4871180401750976e-05 -0.040155717707879068
0.80944447613472548 0.21973329684511689 0.23533277917690931 0.71888461129242454
0.61097884905937128 0.1534430312293702];

% Output 1
y1_step1_ymin = -1;
y1_step1_gain = 0.0291508774279283;
y1_step1_xoffset = 2.8963815;

% ===== SIMULATION =====

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;
if (nargin < 2), error('Initial input states Xi argument needed.');
```

---

```
end

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
elseif ~isempty(Xi)
    Q = size(Xi{1},2);
else
    Q = 0;
end

% Input 1 Delay States
Xd1 = cell(1,3);
for ts=1:2
    Xd1{ts} = mapminmax_apply(Xi{1,ts},x1_step1_gain,x1_step1_xoffset,x1_step1_ymin);
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Rotating delay state position
    xdts = mod(ts+1,3)+1;

    % Input 1
    Xd1{xdts} = mapminmax_apply(X{1,ts},x1_step1_gain,x1_step1_xoffset,x1_step1_ymin);

    % Layer 1
    tapdelay1 = cat(1,Xd1{mod(xdts-[1 2]-1,3)+1});
```

```

a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*tapdelay1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
Y{1,ts} = mapminmax_reverse(a2,y1_step1_gain,y1_step1_xoffset,y1_step1_ymin);
end

% Final Delay States
finalxts = TS+(1: 2);
xits = finalxts(finalxts<=2);
xts = finalxts(finalxts>2)-2;
Xf = [Xi(:,xits) X(:,xts)];
Af = cell(2,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end

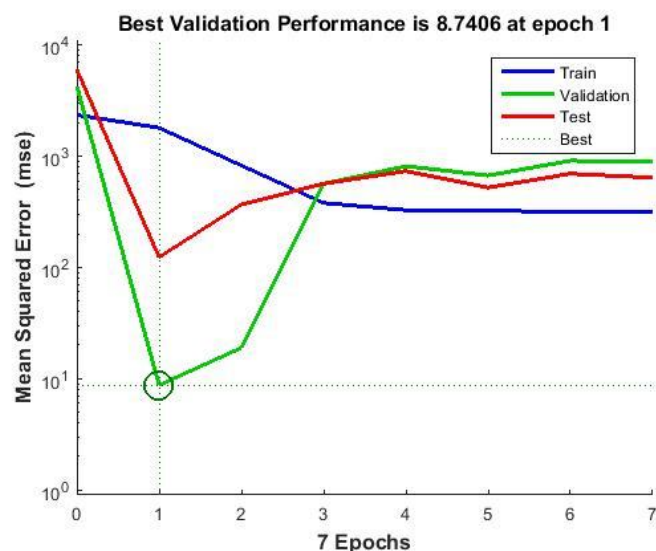
% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings_gain,settings_xoffset,settings_ymin)
y = bsxfun(@minus,x,settings_xoffset);
y = bsxfun(@times,y,settings_gain);
y = bsxfun(@plus,y,settings_ymin);
end

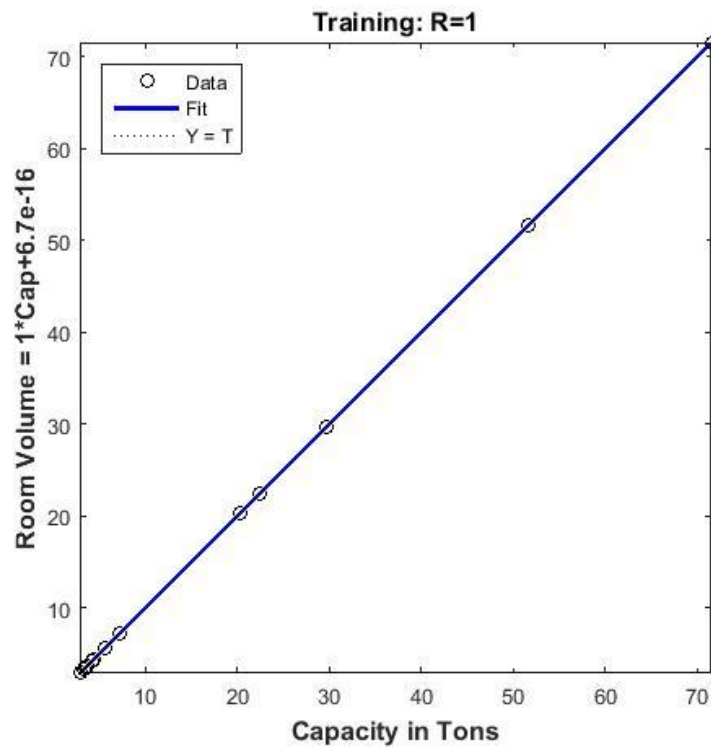
% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings_gain,settings_xoffset,settings_ymin)
x = bsxfun(@minus,y,settings_ymin);
x = bsxfun(@rdivide,x,settings_gain);
x = bsxfun(@plus,x,settings_xoffset);
end

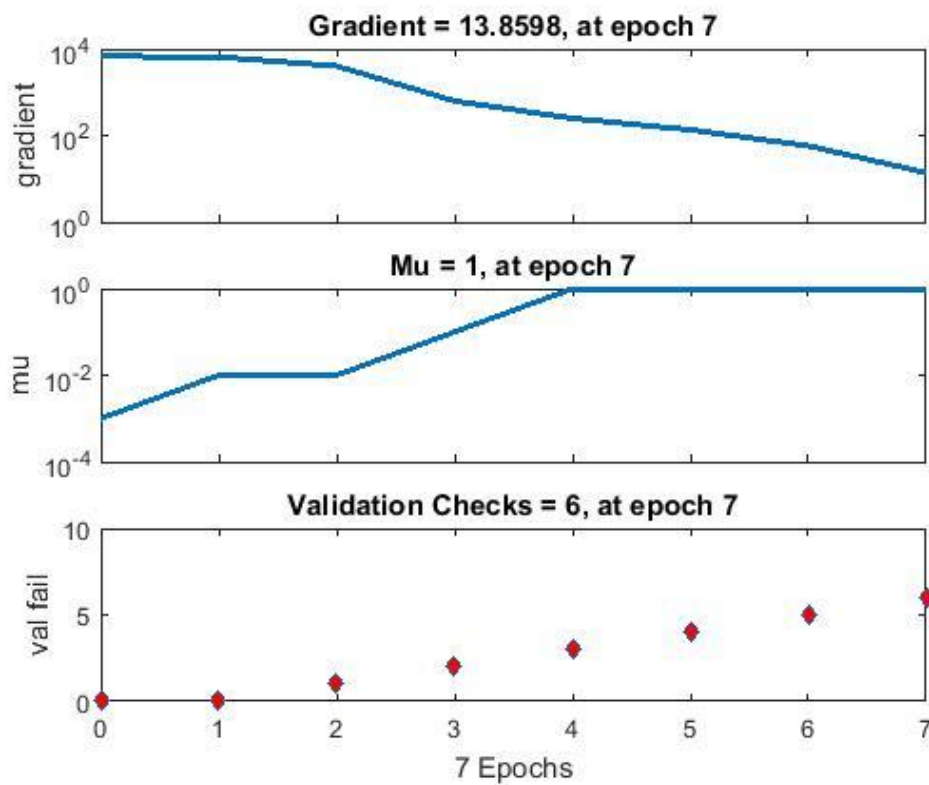
```



The fig below shows the regression graph of room volume against capacity of cooling system.



The fig below shows the state train of cooling system capacity



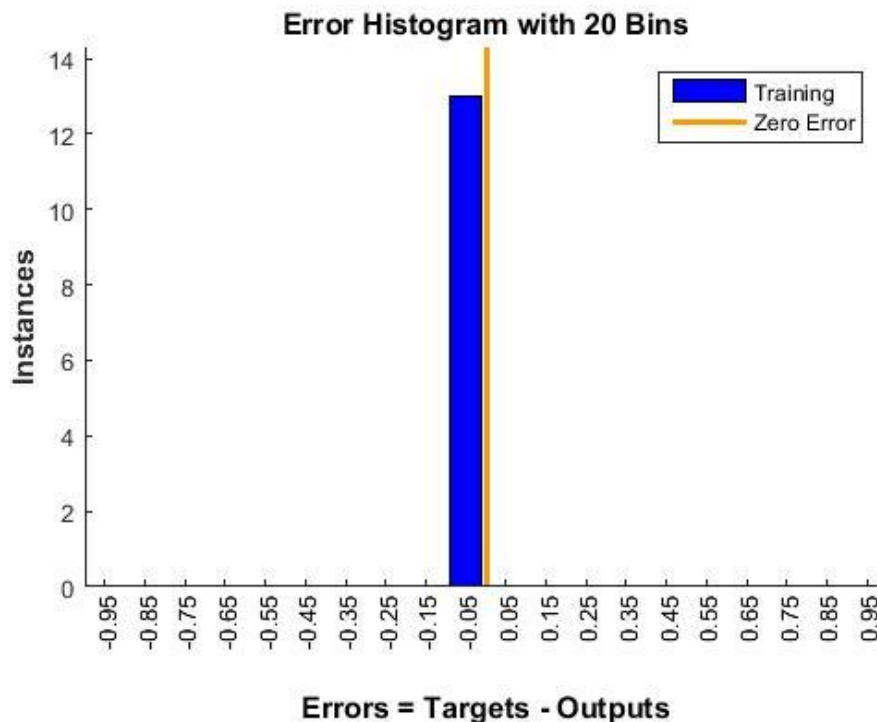


Table 1.3 below shows ventilation, sensible heat factor and cooling capacity.

NAME OF ROOM/SPACE	VENT.	SHF =	CAPACITY
	Q <sub>vent</sub>	RSH/RTH	RTH/3500
	(m <sup>3</sup> )		(Tons)
Room 1	42	0.267015	29.68386
Room 2	26.46	0.423957	22.48338
Room 3	3.36	0.718772	7.278687
Room 4	25.2	0.364505	20.29576
Room 5	70	0.376324	51.67699
Room 6	2.5	0.483043	3.412411
Room 7	2.52	0.535557	2.896382
Room 8	3.5	0.53313	3.7785
Room 9	3.5	0.506353	3.573547
Room 10	112	0.302095	71.50495
Room 11	3.5	0.573479	4.135946
Room 12	3.5	0.601593	4.479082
Room 13	2.356	0.690887	5.632464

#### IV. Discussion of Findings

The outcome of the study of estimate of room size for best performance of a cooling system, using neural network/matlab approach was discussed here. According to table 1.0, the original data was trained to achieve a network performance test level of 266.0675. A closed loop network yielded performance test level of 641.4498, which is reasonably higher and proves improvement/refinement. Training of room volume and infiltration data using levenberg marquardt algorithm at 70% training data, 15% test data and 15% validation data respectively revealed that the best performance level was 106.0012 at epoch 1 and this matches an approximate room size of 128.479m<sup>3</sup> with infiltration of 2.2489 m<sup>3</sup>. The regression coefficient of 1 from the graph indicated that there is a close relationship between room size and infiltration.

Training of room volume and capacity of cooling system from table 1.1, using the same training conditions and algorithm, yielded a forward performance network test level of  $1.1769e+03$  with close loop network performance of  $1.8887e+03$  and that of step ahead performance being equal to forward network performance level. Cooling capacity best performance level was found to be 8.7406Tons at epoch 1 and this corresponds to an approximate room volume of  $308.278m^3$ . Also, the regression coefficient of 1 from the graph indicated that there is a close relationship between room size and cooling system capacity.

#### **V. Conclusion**

Obviously, results from the study revealed that the estimated value of room size (volume) for best performance of a cooling system ranges from  $128.479m^3$  to  $308.278m^3$ . Room size within the estimated value, will give minimum infiltration with optimal cooling load capacity.

#### **VI. Recommendations**

1. Building construction engineers should ensure that rooms that will carry cooling system are constructed based on the estimated value to avoid overloading of the cooling system.
2. Air conditioner manufacturers should design cooling systems whose capacity is within the estimated room sizes.
3. This study can also be done in future using other training algorithms.
4. This research can also be done using multiple room volumes and other advanced program for generalization.

#### **Limitation of the Study**

In the course of carrying out this research, estimate of room size for best performance of a cooling system, the researcher encountered many encumbrances which might cause deviations from actual results. Some of the hindrances includes: data collection, cost of data analysis, lack of electricity, etc.

#### **REFERENCES**

- [1]. Desai, P.S.(2012). Modern air condition and refrigeration for engineers. New Dehi: Khanna Publishers.
- [2]. Jackravut, D. & Nutthaphong, T. (2016). A simplified air conditioning system model with energy management. Journal of International Electrical Engineering Congress,86(1),361-364.
- [3]. Napoleon, E., Odinah,C.E. & Takao, S. (2020). Review of the advances and application of ventilation and air conditioning systems for improving indoor thermal comfort and air quality. International Journal of Energy and Environmental Engineering,2(1).
- [4]. Prathibha, B.V.V., Ajay,K. , Busi, A.K & Naga, D. (2017). Design of air conditioning system for residential/office building. International Journal of Emerging Research in Management and Technology, 6(3).