Volume 2, No. 2, Mar-Apr 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Artificial Neural Networks for Predicting Cooling Load Reduction using Roof Passive Cooling Techniques in Buildings

Shrikant Pandey*
Mechanical Engineering Department,
Mahakal Institute of Technology,
Ujjain, M.P., India,
shrikantpandey@yahoo.com

D.A. Hindoliya
Mechanical Engineering Department,
Ujjain Engineering College,
Ujjain, M.P., India,
dev_hindoliya@yahoo.com

Ritu Pandey
Electronic and communication Engineering Department,
Mahakal Institute of Technology,
Ujjain, M.P., India,
ritu_mod@gmail.com

Abstract: Two identical prototype rooms having dimension 1m×1m×1m, were constructed using brick work. RCC roofs having thickness of 100 mm were constructed on both rooms. Two roof passive cooling strategies (e.g., roof pond, insulation over the roof) were applied on one of the test rooms one by one. The other room was kept with bare roof. The results show that the average % reduction of roof cooling load was found to be 45 %., 30%, using roof pond, using insulation (thermocol) respectively.

The objective of this work is to train an artificial neural network (ANN) to learn to predict the reduction of cooling load of buildings. Five training algorithms traincgb, traingdx, traingda, trainlm, and trainsc were used to create an ANN model. An ANN has been trained based on number experimental data of cooling load. The network output is reduction in heat gain though roof. The Intelligent model predicts reduction in cooling load with accuracy, more than 90%. The accuracy of the prediction could be improved by more input data. The results show that the predicted data is in good harmony with the experimental data, which indicates artificial neural network is a novel and reliable method to predict reduction in cooling load.

Keywords: Artificial neural network, Cooling-load reduction; Roof cooling; passive cooling, energy saving.

I. INTRODUCTION

Regression models have been established to be effective for building energy predictions in a number of experiments (e.g., [1–3]). Relatively few parameters must be recognized, thus reducing the time required for the model expansion. The regression models do not, however, accurately reflect the hourly or sub-hourly energy demand. They are best suited for predicting the average expenditure over longer periods such as days or months. For different buildings with different environment and weather conditions, much effort and time must be spent on selecting time scales and regresses to find a best fit model. Also, autocorrelation or multi co linearity problems must be considered when evaluating the performance of prediction because they tend to lead to model ambiguity.

The application of artificial neural network (ANN) has reached various fields including mode identification, image processing, nonlinear optimization, expert system, etc. It has also been widely used in HVAC [4]. Load prediction is the Foundation and premises of the optimal control of ice storage system. Kawashima [5] points out that the operating cost of predictive control decreases by 13.5% compared with that of chiller priority. In 1990, Ferrano [6] described an ANN model to predict the next day's total thermal load. In 1993, models of thermal load prediction used in the first building energy prediction competition sponsored by ASHRAE include regression model (linear regression, [1]multiple regression, recursive regression), time-series model (ARIMA, ARMA, AR, MA, etc.), Kalman filter model, fuzzy set model, ANN model, etc. [7].The use of time-series analysis techniques to forecast energy use is logical because the history of energy use can be represented by a time series. Kimbara et al. [8] experimented with the autoregressive integrated moving average (ARIMA) model and found the performance of ARIMA was better than a two-dimensional autoregressive (AR) model. Several models and applications have been implemented based on the autoregressive moving average with exogenous input model (ARMAX) model (e.g., [9]). On one hand, time-series models can capture the relationship between the hourly energy use and time variation given a set of time-series data. On the other hand, both ARMA and AR models work under the hypothesis that the present value is a linear combination of the previous ones. In most cases, this assumption is invalid. The ARIMA and ARMAX models

Nomenclature

Q_R = Heat transferred through roof [watts]
U_R = Overall heat transfer coefficient [its value for a 100mm thick concrete slab is 1.21 W m^2]
L = Thickness of the concrete slab [meter]
A = Surface area of the roof [m^2]
T_0 = Temperature at outer surface [°C]
T_i = Temperature at inner surface [°C]
K = Coefficient of thermal conductivity [W m^{-2} $k^{-1,}$ 1.20 W m^{-2} k^{-1}]

can handle the changes in an unstationary process, but require the judgment of many parameters. Also, the autocorrelation between variables must be considered because it strongly impacts the precision of the prediction. Dhar et al. [10] used a Fourier series model to predict the energy demand in an institutional building. Fourier series models provide better performance compared to the above time-series models; however, they are based on the assumption that energy use in most buildings is periodic. If dramatic changes happen, high-frequency Fourier components must be included in the model, thus considerably increasing the computational cost. An artificial neural network (ANN) is a type of artificial intelligence technique that mimics the behavior of the human brain. It can approximate a nonlinear relationship between the input variables and the output of a complicated system. The main advantage of an ANN model is its self learning capability. The use of ANN in building energy prediction has been investigated by many researchers (e.g., [11, 12]). Their models share some similarities, but each differs significantly in implementation details because each is tailored toward a specific type of energy prediction under a specific building environment. On one hand, ANN models estimate parameters faster by learning from examples automatically. On the other hand, because it is hard to distinguish structure from noise in the data, an ANN tends to memorize noise. Also, an ANN might not be able to adapt to dramatic changes such as unstable behavior in the power load–temperature relationship. Most of the literature focuses primarily on static prediction. In a static prediction, the prediction model is set up in advance using historical data and does not change afterward, when new information become available. It is highly possible that such a model becomes invalid when new patterns emerge and more recent data becomes available. In this case, a dynamic prediction model that can adapt itself to such changes in the energy consumption pattern is desirable. This is especially true for short-term energy prediction. Only a few dynamic prediction systems were found in the literature and all in the field of building load forecasting. Djukanovic et al. [13] used an adaptive system for short-term load forecasting with a moving window consisting of data linked with the 4 previous weeks as well as with 8 weeks at the same time in the previous year. Khotanzad et al. [14] used a combination of three separate models (i.e., weekly, daily and hourly models) for short-term load forecasts. Each model is updated at the end of each period with the data associated with that period (e.g., at the end of each week, the weekly model is updated using the last week data). Mohammed et al. [15] used three ANN adaptive models and a two-stage training algorithm. The first training stage produces a set of initial ANN weights that capture the general, day by day trend of the building load. During the second-stage training, the ANN is refined and enhanced to capture special features of that particular day for which the forecast is made by using a subset of data consisting of those that share similar temperature conditions with the day being forecasted as well as data from the previous 5 days. These approaches have potentials in the area of on-line building energy prediction. The regression and time-series models are based on classical mathematical theory. Thus, the behavior of these models is well understood and the model parameters are straightforward to estimate. However, these models tend to work well only for energy systems that are well behaved. The ANN model generally works better for buildings that exhibit highly nonlinear energy consumption patterns. However, the success of using ANN depends on a number of design issues such as the choice of input and output data, the number of hidden layers, the number of neurons used in each layer and the training algorithms used. A few studies have compared these forecasting methods. Kawashima et al. [16] found that ANN models provide the best performance. An ANN model has the unique advantage that no clear relationship between the input variable and output needs to be defined before the model is used in the prediction process, since this relationship is identified through a self-learning process. Because of this unique feature, the time and effort that are normally required to establish a proper mathematical model in a conventional prediction methodology can be significantly reduced. The ANN-based models appeared to perform well in the two Great Energy Shootout competitions organized by ASHRAE. Therefore, this research adopts ANN and explores ways to develop adaptive ANN models for the prediction of buildings energy demand

Experimental Procedure

Figure 1: Experimental setup

Two identical prototype rooms each having dimensions 1 m×1m× 1m were fabricated. All the walls of rooms were constructed using brick work. Both rooms have one door of 0.5m×0.4 m and one window of 0.4 m×0.3 m. The roofs were constructed using RCC slab of 100 mm thickness. Two passive techniques (e.g., roof pond and insulation over the roof) were applied on one of the test rooms (Room 1) while other test room was kept with bare RCC roof (Room 2). In first technique, water was filled to a depth of 10 mm to build a roof pond, and net cooling load was calculated. In second technique a 20 mm thick thermocol sheet was covered on the roof surface of the test room. Sufficient numbers of experiments were performed under different climatic conditions using the above mentioned techniques one by one. Other climatic parameters such as ambient temperature, intensity of solar radiation and wind velocity, relative humidity were same for both the test structures as they were built side by side. Numbers of thermocouples were fixed inside and outside surfaces of roof on both the structures for temperature measurement. Temperatures were recorded at a regular interval of 15 minutes using PC based data logger. The provision was also made to measure the intensity of solar radiation using pyranometer with shading ring. The heat transfer through the roof was calculated by the following equation

$Q_R = U_R \times A\ (T_o — T_i)$

Following constants values were used in calculation.

L = 0.1 m

A =1 x 1= 1 m^2

K = 1.20 w/m-k.

Artificial neural network (ANN)

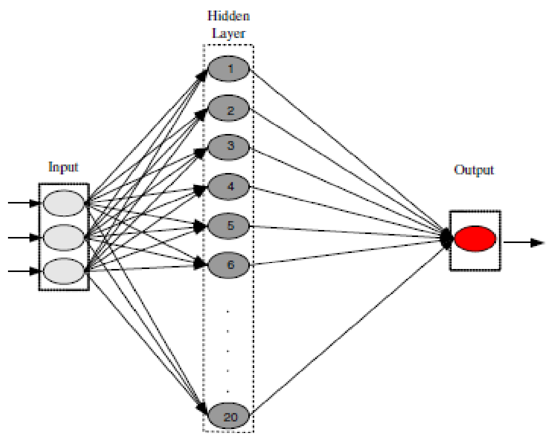

Figure 2: Multi-layer feed forward neural network

Artificial neural network imitates the working principles of human brain and performs learning and prediction. Learning of a network shortly can be determined as the adjustment of the weights and the variables of the activation and transfer functions in order to perform a desired function. The advantage of ANN from other methods is its accomplishment in modeling the complex problems having many variables easily. The structure of a multi-layered feed forward, backpropagation network used in this study. ANNs imitate the learning process of a biological brain. The neural network, through the learning process, understands the underlying functional relationships in the loaded data and the same are stored as inter-neuron connection strengths or synaptic weights. A schematic diagram (figure2) of a typical multi-layer feed forward neural network. The network consists of an input layer, one or more hidden layers and one output layer. Neurons in one layer are connected to all the neurons of previous and subsequent layer. Each connection between two neurons is associated with an adaptable synaptic weight. Information is processed at each neuron. Information received from all the connected neurons is summed up and passed through an activation function and the activation outcome is sent out to the subsequently connected neurons. The network needs to be trained to give the desired output using a training set. Training set is a group of input sets and corresponding desired output set. Training involves the revision of the synaptic weights. The training set should be self-sufficient to train the network. The network reads each set of input data and produces an output. This output is then compared with the desired output. Before the training is completed, there would obviously be a difference between the network output and the desired output. Then the synaptic weights are adjusted such that the error function is decreased. This way, the network adjusts its synaptic weights, while running through all the input and desired output sets. When the network has run through all the input patterns, root mean square error (RMSE) is compared with the maximum desired tolerance. If it is greater than the maximum desired tolerance, a new 'epoch' (a run through all training input–output sets) is started after the completion of the current one, and the synaptic weights are further adjusted towards reducing the error function. This process is repeated until the network achieves an error function less than the desired tolerance. This is called as the back propagation algorithm.

The neural network models were developed using Matlab [11]. Several training algorithms available in Matlab used in this research have basic variations in training the model. In an overall description, these algorithms are classified in four categories:

1. Steepest descent: Traingd, Traingdm, Traingda, Trained, Trainer
2. Conjugate gradient: Tracing, Traincgp, Traincgb
3. Newton's method: Trainbfg, Trainoss
4. Levenberg-Marquardt: Trainlm, Trainbr.

The hourly data set consists of outdoor temperature, solar intensity, wind velocity and relative humidity collected at constant interval of time i, for i = 1, 2 . . . . n. These variables will become inputs to an ANN to predict the cooling load reduction .A numbers of runs of the software were done to generate numbers of data for training and validation of the network. Also a number of different network sizes and learning parameters were tried aiming at finding the one that could result in the best overall performance. The architecture of the network used is quite simple. The network is a feed forward back propagation neural network with 25 neurons in the input layer, 10 in the hidden layer (can be adjusted by user) and one in the output. Adding more layers or neurons can potentially improve the prediction accuracy, but also adds complexity and also.ANN training time. The MATLAB ANN toolbox is used to build, and train the network. The ANN toolbox offers several training algorithms for training the network. Five algorithms were tested in this study. The activation function used in the input layer is pure linear while for the other two layers logarithmic sigmoid function was used. We have tried five following different training algorithms

Table I

Name of training algorithms	Description	Syntax
TRAINGDX	Gradient descent with momentum and adaptive learning rate backpropagation traingdx is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate. traingdx(net,TR,trainV,valV ,testV)	[net,TR] = traingdx(net,TR,trainV,valV, testV) info = traingdx('info')
TRAINGDA	Gradient descent with	[net,TR] =

	adaptive learning rate backpropagation. Traingda is a network training function that updates weight and bias values according to gradient descent with adaptive learning rate. traingda(net,TR,trainV,valV ,testV)	Traingda(net,TR,trainV,valV, testV) info = traingda('info')
TRAINSCG	Scaled conjugate gradient backpropagation trainscg is a network training function that updates weight and bias values according to the scaled conjugate gradient method. trainscg(net,TR,trainV,valV, testV)	[net,TR,Ac,El] = trainscg(net,TR,trainV,valV, testV) info = trainscg('info')
TRAINCGB	Conjugate gradient backpropagation with Powell-Beale restarts traincgb is a network training function that updates weight and bias values according to the conjugate gradient backpropagation with Powell-Beale restarts. traincgb(net,TR,trainV,valV , testV)	[net,TR] = traincgb(net,TR,trainV,valV, testV) info = traincgb('info')
TRAINLM	Levenberg-Marquardt backpropagation trainlm is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization. Trainlm is often the fastest backpropagation algorithm in the toolbox, and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms. trainlm(net,TR,trainV,valV, testV)	[net,TR] = trainlm(net,TR,trainV,valV, testV) info = trainlm('info')

Where, net: Neural network, TR= Initial training record created by train, trainV =Training data created by train, valV= Validation data created by train, testV= Test data created by train

II. TRAINING AND TESTING OF PREDICTION MODELS

For the training of an artificial neural network hundred of data are used. Data are collected experimentally after every 15 minutes reading is taken they are feed to train artificial neural network. Training stops when any of these conditions occurs:

- The maximum number of epochs (repetitions) is reached.
- The maximum amount of time is exceeded.
- Performance is minimized to the goal.
- The performance gradient falls below min_grad.
- Validation performance has increased more than max_fail times since the last time it decreased (when using validation).

III. RESULTS / VALIDATION

The optimization of neural networks often results in different networks, dependent on the initial random values of the synaptic weights. Therefore, the outcome will, in general, not be the same in two different trials even if the same training data have been used. In this article, we have only presented the best result obtained after about 20-30 trials with the same input–output data for different model. MAE-value for test data has been used as criterion when comparing the final performance of different networks. Typical test results for cooling load prediction are shown in different graphs and tables.

Figure 3, 4, 5 , 6 and 7 shows % reduction in heat gain from roof by roof pond for one day sample run for traincgb, traingdx, traingda, trainlm, and trainsc algorithms respectively. Figure 8,9,10,11 and 12 shows % reduction in heat gain from roof by insulator for one day sample run for traincgb, traingdx, traingda, trainlm, and trainsc algorithms respectively.

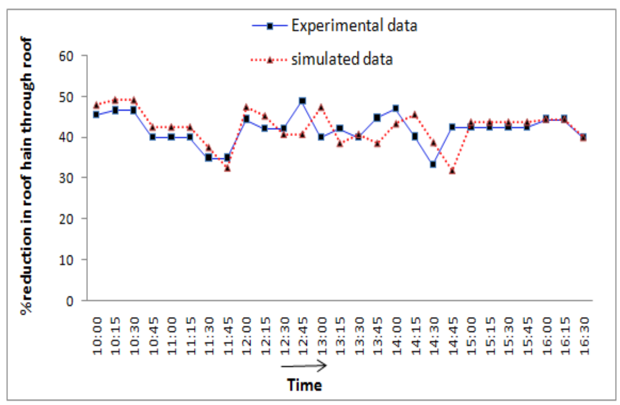

Figure 3: comparison between experimental and ANN data (traincbg) for roof pond

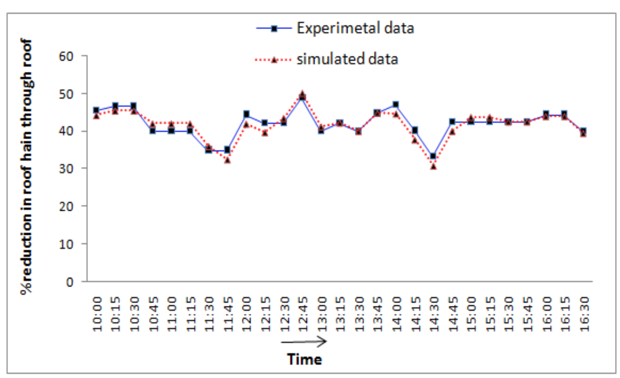

Figure 4: comparison between experimental and ANN data (traingdx) for roof pond

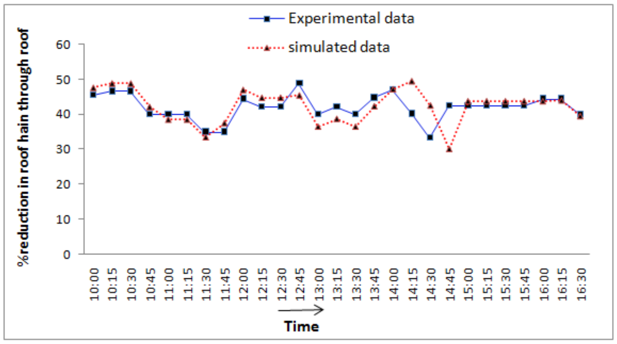

Figure 5: comparison between experimental and ANN data (traingda) for roof pond

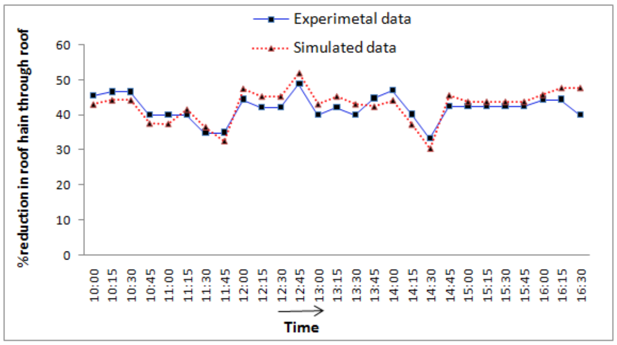

Figure 6: comparison between experimental and ANN data (trainlm) for roof pond

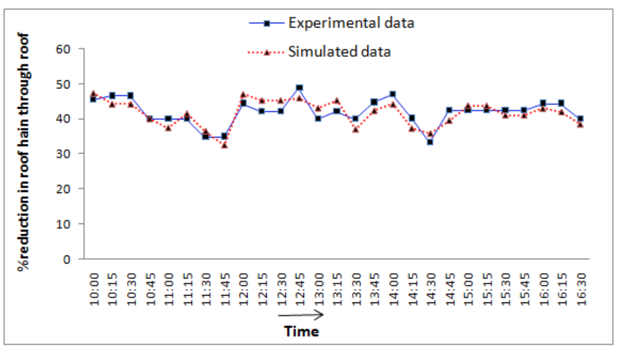

Figure 7: comparison between experimental and ANN data (trainscg) for roof pond

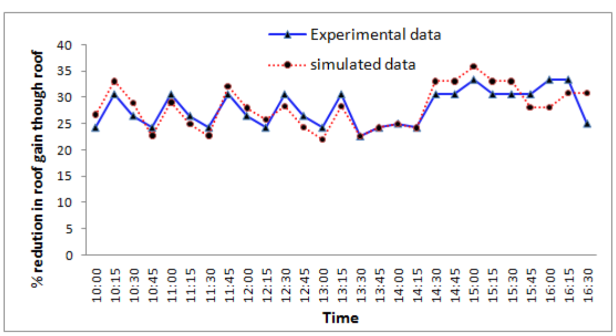

Figure 8: comparison between experimental and ANN data (traincbg) for insulator

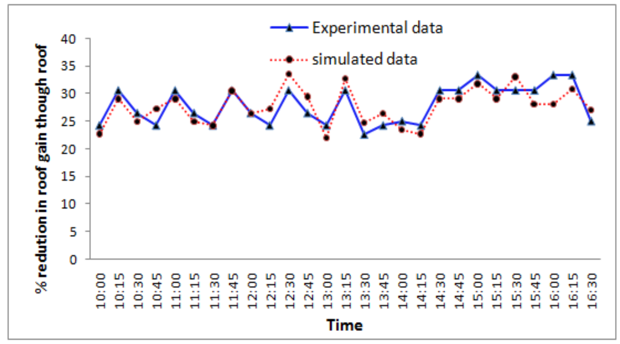

Figure 9: comparison between experimental and ANN data (traingdx) for insulator

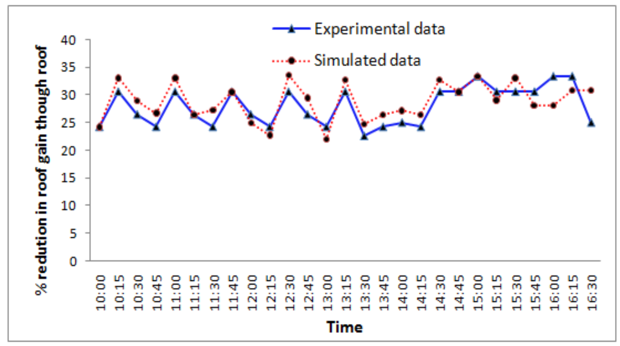

Figure 10: comparison between experimental and ANN data (traingda) for insulator

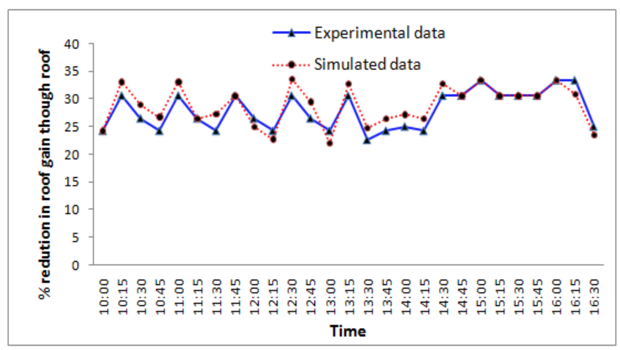

Figure 11: comparison between experimental and ANN data (trainlm) for insulator

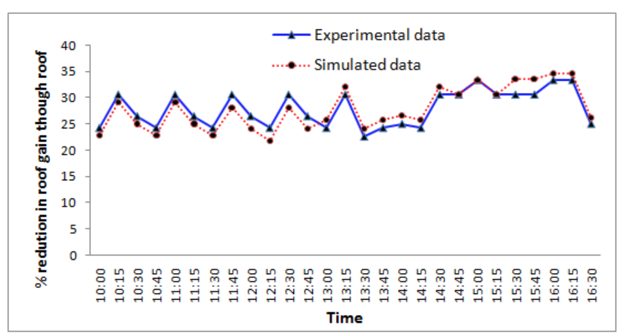

Figure 12: comparison between experimental and ANN data (traingda) for insulator

Table II: comparison of different training algorithm of ANN modal for roof pond method

Training algorithm	No of neurons	No of epoch	Learning rate	RMSE	Max. Error	Performance	gradient	Validation check	No of iteration	Time (sec)
Traingdx	25	500	0.1	1.6	5.6	0.9	1.4	200	281	9
Trainlm	25	500	0.1	0.5	2.9	1.4	0.6	1	5	1
Traingda	25	500	0.1	1.8	5.5	0.9	1.2	36	100	2
Trainscg	25	500	0.1	2.1	3.5	0.7	1.5	396	460	9
Traincgb	25	500	0.1	0.6	5.0	0.5	0.9	450	462	5

Table III: comparison of different training algorithm of ANN modal for insulator

Training algorithm	No of neurons	No of epoch	Learning rate	RMSE	Max. Error	Performance	gradient	Validation check	No of iteration	Time (sec)
Traingdx	25	500	0.1	1.1	4.6	1.1	1.5	250	300	9
Trainlm	25	500	0.1	0.6	2.9	2.1	1.5	1	9	3
Traingda	25	500	0.1	1.8	2.5	0.9	1	32	100	5
Trainscg	25	500	0.1	2.1	4.5	0.8	4.5	456	460	9
Traincgb	25	500	0.1	0.8	3.5	1.1	1.8	400	462	7

IV. ANN MODAL FOR ROOF POND

Table 2 shows the optimization of neural networks for estimation of reduction in heat gain from roof with different numbers of inputs for test structure. It shows the performance numbers for different training algorithms. The following observations can be made from the results of the identification experiment: All estimation models presented in this report give estimates of the reduction of heat gain from roof. The lowest RMSE-value for test data was obtained using a trainlm algorithm. This network gave RMSE = 0.5 for training data The Levenberg–Marquart (LM) algorithm appeared to be the fastest training algorithm however, because the LM method must solve a linear system of equations in order to obtain the search direction, the computation becomes expensive when the number of input elements and the volume of the training data increase. Therefore, when the volume of the data is large, the standard gradient descent algorithm is used for training the ANN. This roof cooling system gives stable indoor temperature and lower heat flux flowing out through the roof than bare RCC roof. Even in the afternoon, when the air temperature is relatively higher and solar radiation is intense, there is still a lower amount of heat flux flows through roofs into the room. Figure 3-7 shows comparative performance for different training algorithms for the sample data of roof pond. All the five algorithms give excellent performance and fairly accurate prediction so this modal can adopted for perdition of reduction in heat gain from roof for roof pond cooling system. Once a satisfactory degree of input - output mapping has been reached, the network training is freeze and the set of completely unknown test data were applied for verification. The difference between predicted and actual indoor temperature, for the testing set, was 3% to 6% for different network.

V.ANN MODAL FOR INSULATED ROOF

Table 3 shows the optimization of neural networks for estimation of reduction in heat gain from roof with different numbers of inputs for test structure. It shows the performance numbers for different training algorithms. The following observations can be made from the results of the identification experiment: All estimation models presented in this report give estimates of the reduction in heat gain from roof. The lowest RMSE-value for test data was obtained using a trainlm algorithm. This network gave RMSE = 0.6 for training data. The Levenberg–Marquart (LM) algorithm appeared to be the fastest training algorithm. This roof cooling system has stable indoor temperature and lower heat transfer from roof than bare RCC roof. Therefore better thermal satisfaction can be achieved in insulated roof. Figure 4-12 shows comparative performance for different training algorithms for the sample data of passive cooling by insulated roof. All the five algorithms give excellent performance and fairly accurate prediction so this modal can adopted. Once a satisfactory degree of input - output mapping has been reached, the network training is freeze and the set of completely unknown test data were applied for verification. The difference between predicted and actual indoor temperature, for the testing set, was 4% to 6% for different network.

V. CONCLUSIONS

The article presents an intelligent design tool of a building in terms of passive environmental performance. The model proposed was first validated with measured experimental data sets. Results proved highly satisfactory and provided enough confidence for the process to be extended to a larger solution space for which there is uneconomical and time consuming way of calculating the solution. This model incorporates greater accuracy, accounting for two different passive roof cooling technique. The results indicated very good agreement between the experimental results and model predictions. A neural network model based on a back propagation algorithm was used for estimation of hourly cooling load reduction. Training and testing values of cooling load reduction were compared with the experimental data it was found that the neural network could successfully simulate the cooling load reduction. The Intelligent model predicts reduction in cooling load with accuracy more than 90%. The results also show that the average % reduction of heat gain from roof was found to be 45 %, 30%, using roof pond, using insulation (thermocol) respectively. The lowest RMSE-value for test data was obtained using a trainlm algorithm. The network gave lowest RMSE, and fastest result for trainlm algorithm, so most situations; we recommend that you try the Levenberg-Marquardt algorithm first. If this algorithm requires too much memory, then try one of the conjugate gradient methods.

VI. REFERENCES

[1] M. Fels, Special Issues devoted to measuring energy savings: the scorekeeping approach, Energy and Buildings 9 (1–2) (1986) 5–18.

[2] D. Ruch, D.E. Claridge, A four–parameter change-point model for predicting energy consumption in commercial buildings, ASME Journal of Solar Energy Engineering 114 (2) (1992) 77–83.

[3] S. Katipamula, T.A. Reddy, D.E. Claridge, Multivariate regression modeling, ASME Journal of Solar Energy Engineering 120 (1998) 177–184.

[4] Y. Y. Li, "Artificial neural network (ANN) research in HVAC field", *Journal of HV& AC*, (31): 38-41, 2001.

[5] M. Kawashima, Charles E Dorgan, et al, "Optimizing system control with load prediction by neural networks for an ice-storage system", ASHRAE Transactions, (102): 1169- 1178, 1996.

[6] F. J. Ferrano, K. V. Wong, "Prediction of thermal storage loads using a neural network", ASHRAE Transactions, (96): 723-726, 1990.

[7] J. F.Kreider, Jeff S. Haberl, "Predicting hourly building energy use: The great energy predictor shootout---overview and discussion of result", ASHRAE Transactions, (100): 1104-1118, 1994.

[8] A. Kimbara, S. Kurosu, R. Endo, K. Kamimura, T. Matsuba, A. Yamada, On-line prediction for load profile of an air-conditioning system, ASHRAE Transactions 101 (2) (1995) 198–207.

[9] Y. Hong-Tzer, H. Chao-Ming, H. Ching-Lien, Identification of ARMAX model for short term load forecasting: an evolutionary programming approach, IEEE Transaction on Power Systems 11 (1) (1996) 403–408.

[10] A. Dhar, T.A. Reddy, D.E. Claridge, Modeling hourly energy use in commercial buildings with Fourier series functional form, ASME Journal of Solar Energy Engineering 120 (3) (1998) 217–223.

[11] M. Anstett, J.F. Kreider, Application of neural networking models to predict energy use, ASHRAE Transactions 99 (1) (1993) 505–517.

[12] P.S. Curtiss, J.F. Kreider, M.J. Brandemuehl, Energy management in central HVAC plants using neural networks, ASHRAE Transactions 100 (1) (1994) 476–493.

[13] M.R.S. Djukanovic, B. Babic, D.J. Sobajic, Y.H. Pao, A neural-based short term load forecasting using moving window procedure, Electric Power and Energy System 17 (6) (1995) 391–397.

[14] A. Khotanzad, R.C. Hwang, A. Abaye, A.D. Maratukulam, An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities, IEEE Transactions on Power System 10 (3) (1995) 1716–1722.

[15] O. Mohammed, D. Park, R. Merchant, T. Dinh, C. Tong, A. Azeem, J. Farah, C. Drake, Practice experience with an adaptive neural networks short-term load forecasting system, IEEE Transactions on Power Systems 10 (1) (1995) 254–265.

[16] M. Kawashima, C.E. Dorgan, J. Mitchell, Hourly thermal load prediction for the next 24 h by ARIMA, EWMA, LR, and an artificial neural network, ASHRAE Transactions 101 (1) (1995) 186–200.