International Journal of Engineering Research and Development
***e**-ISSN: 2278-067X, **p**-ISSN: 2278-800X*
Volume 4, Issue 2 (October 2012), PP. 35-43

Hybrid Neural Network Model for the Design of Footing

Dr. Ramesh Babu Bandi[1], Dr. Sudarsana Rao Hanchate[2]
[1]*Principal, Ananthalakshmi Institute of Technology & Sciences, Anantapur – 515 002. A.P.(State)*
[2]*Professor of Civil Engineering, J.N.T.U. College of Engineering, Anantapur – 515 002. A.P.(State)*

Abstract:- This paper demonstrates the applicability of Artificial Neural Networks (ANN) and Genetic Algorithms (GA) for the initial design of footing. A hybrid neural network model which combines the features of feed forward neural networks and genetic algorithms have been developed for the design of footing. The network has been trained with design data obtained from design experts in the field. After successful learning, the model predicted the dimensions of footing and areas of reinforcement required for new problems with good accuracy satisfying all design constraints. The various stages involved in the development a genetic algorithm based neural network model are addressed at length in this paper

Keywords:- Footing, Hybridization, Genetic algorithms, Neural networks.

I. INTRODUCTION

Structural Engineering involves understanding and modeling of natural phenomenon, material behavior, laws of mechanics, intuition, past experience or expertise and analysis techniques. The modern computer can bring speed, efficiency and accuracy in analysis of structures. But to computerize the areas such as conceptual design, modeling of natural phenomenon and material behavior, damage assessment etc., is extremely difficult as it requires human expertise. Structural design is an iterative process. The initial design is the first step in design process. Though the various aspects of structural design are controlled by many codes and regulations, the structural engineer has to exercise caution and use his judgment in addition to calculations in the interpretation of the various provisions of the relevant code to obtain an efficient and economic design. After the design process the designer makes an overall guess about the possible optimum solution consistent with designer's experience, knowledge, constraints, and requirements. The analysis of the structure is then carried out using initial design. Based on the results of the analysis a re-design of the structure is carried out if any of the constraints is not satisfied. The efficiency of the design process depends heavily on initial guess. A good initial design reduces the number of subsequent analysis-design cycles. This phase is extremely difficult to computerize, as it needs human intuition. In recent years efforts have been made to computerize the initial design process using artificial neural networks as they can learn from available designs during training process. Artificial neural network is a new technology emerged from approximate simulation of human brain and has been successfully applied in many fields of engineering. Neural networks and genetic algorithms demonstrate powerful problem solving ability. They are based on quite simple principles but take advantage of their mathematical nature in terms of non-linear iteration. Neural networks with back propagation learning showed results by searching for various kinds of functions. However the choice of basic parameters (Network topology, learning rate, initial weights) often already determines the success of the training process. However, there are no clear rules how to set these parameters. Yet these parameters determine the efficiency of training. On the other hand genetic algorithms are global search methods, that are based on principles like selection, cross over, and mutation. By combining genetic algorithm with neural networks, considerable reduction in network parameters can be achieved. Thus, hybridization of neural networks with genetic algorithms considerably improves their efficiency. More details about the principles of neural networks and genetic algorithms can be found in Rajasekharan and Vijayalakshmi Pai [1] and Davis [2]. The scope of this paper is to demonstrate their applicability for the design of footing.

II. BRIEF REVIEW

Lot of research has taken place on applications of artificial neural networks in structural engineering. Hong-Guang and Wang used artificial neural networks for predicting compressive strength of concrete[3]. Sanad and Saka applied artificial neural networks for predicting ultimate shear strength of reinforced concrete deep beams[4]. Cladera and Mari trained an artificial neural network for shear design of reinforced cement concrete beams[5, 6]. Hadi developed a neural network model for the design of fiber reinforced concrete beams[7]. Ghaboussi and Joghatie used artificial neural networks for the active control of structures[8]. Mukherjee and Deshpande applied this principle for developing a neural network model for the structural design of Reinforced concrete beams [9]. Mishra and Akhil Upadhyay developed a simple neural network model for the design of Reinforced cement concrete columns under uni-axial bending[10]. In most of these works the neural networks have been trained by using traditional back propagation algorithm of Rumelhart and McClelland [11]. In this approach the connection weights of neural networks are initially set to some random values. These values are then modified automatically according to the learning algorithm during the process of learning. This type of learning requires huge number of training cycles and also requires higher network configuration. It is reported that these networks trained by back propagation algorithm may get trapped in a local minima. To alleviate this problem, the present paper proposes to use genetic algorithm in conjunction with back propagation neural networks. Genetic algorithms have been successfully used in field of structural engineering. Jenkins applied genetic algorithm for optimum design of trussed beam roof structure[12, 13]. Leite and Topping suggested improved genetic operators for optimization [14]. Topping and Leite have developed parallel genetic models for structural optimization[15]. In the present paper a hybrid neural network, which combines the features of genetic algorithms and back propagation networks is presented as an improved approach for handling structural design problems. The applicability of genetic algorithms for modeling the structural design process of column has been explored. As an example, a genetic algorithm based neural network model has been developed for the design of short column subjected to biaxial bending. The genetic algorithm based neural network model has been developed to satisfy all the requirements of I.S. 456-2000 [16]. The neural network learns the design process in an adaptive fashion through the training examples presented to it. The training examples have been obtained by posing different column problems to various design experts and structural engineers. The weights for the neural network have been obtained by using a genetic algorithm. This alleviates large number of training cycles required for training through back propagation algorithm and also reduces the configuration considerably. Presenting new design problems after successful training has validated the developed neural network model. The hybrid neural network model successfully predicted length, breadth, depth of footing and areas of reinforcement along length and breadth of footing.

III. DEVELOPMENT OF HYBRID NEURAL NETWORK MODEL

The various stages involved in the development of hybrid neural network model are presented below.

Generation of exemplar patterns

The objective of this work is to develop a GA based hybrid neural network model for the design of footing. This requires a comprehensive set of examples that cover various parameters influencing the design of footing. All the training examples should invariably satisfy I.S. 456-2000 code provisions. For the present work, all the training examples have been developed by presenting different footing problems to various design experts. The experts were asked to provide designs satisfying code provisions. The design variables considered are the longer and shorter sides of column, load on column, bearing capacity of soil, grade of concrete and grade of steel. The example designs have been obtained for different combination of variables. M20, M25 and M30 grades of concretes have been considered. Reinforcement steel of three different grades viz. Fe 250, Fe 415 and Fe 500 have been considered. For each set length, breadth, depth of footing and areas of reinforcement along length and breadth of footing are obtained. For the present problem, a total of one hundred training examples have been obtained from different experts such that these examples cover all the possible combinations of design variables considered. Out of these seventy five examples have been used for training and twenty five examples are used for validation.

Selection of input and output

In the present work, it is required to develop a model for the design of footing. Hence, the model should be able to predict the values of length, breadth, depth of footing and areas of reinforcement along length and breadth of footing for given dimensions of column, load on column, bearing capacity of soil, grade of steel and concrete. The input layer for the network has been configured taking in to account the possible parameters that may influence the output. As the network is supposed to map the functional relationship between the input and output parameters, the performance of the network is highly sensitive to the input information. In addition, proper choice of input parameters improves the net performance for unseen problems i.e. the generalization capability. Accordingly the input to the network is chosen as follows: Accordingly the input to the network is chosen as follows:

- Longer side of column (a)
- Shorter side of column (b)
- Load on column (P_u)
- Grade of concrete (f_{ck})
- Grade of steel (f_y)
- Bearing capacity of soil (q_s)

Thus the input vector selected for this model is

IP = $\{a, b, P_u, f_{ck}, f_y, q_s\}$

Although the relationship between the input parameters and macroscopic behavior of the material is highly non-linear, the quantitative degree of non-linearity is not clearly known. Hence, only the linear terms have been induced in the input vector. The network is expected to establish the degree of non-linearity through the training examples in an implicit manner.

The designer would like to know the Length of footing (L), Breadth of footing (B), Depth of footing (D)area of the reinforcement along length of footing (A_L), and area of the reinforcement along breadth in central band of footing (A_B) for any given design problem.

Accordingly, the output vector for the neural network model is selected as

$$OP = \{L, B, D, A_L, A_B\}$$

From the literature available it is learnt that computers work better for the values lying in between 0 and 1. So the input and output parameters have been normalized in the range (0, +1) using suitable normalization or scaling factors. This has been done by dividing the greatest entry at a node by a scale factor slightly greater than it.

Selecting a suitable network configuration

As mentioned earlier, the network configuration is defined in terms of the number, size, nodal properties, etc, of the input/output vectors and the intermediate hidden layers, once the input and output vectors are decided to cater the present investigation requirements, the task of selecting a suitable configuration has been taken up. There is no direct method to select number of nodes in hidden layers. Generally a trial and error method is adopted for arriving at the network configuration. After doing a few trials, it is observed that the network with 6 neurons in one hidden layer is behaving well. Accordingly a configuration of (6-6-5) has been selected for this network model. The architecture is depicted in figure 1

Training of the network

The training of the present network has been accomplished using the back propagation algorithm. Conventionally, a Back Propagation Network (BPN) determines its weights based on a gradient search technique and hence runs the risk of encountering local-minima. Genetic Algorithm (GA) on the other hand is found to be good at finding 'acceptably good' solutions. The idea to hybridize the two networks has been successful to enhance the speed of training [1] In the present work, the weights for the BPN have been obtained by using GA. Genetic algorithms which use a direct analogy of natural behavior work with a population of individual strings, each representing a possible solution to the problem considered. Each individual string is assigned a fitness value, which is an assessment of how good a solution is to a problem. The high-fit individuals participate in "reproduction" by cross-breeding with other individuals in the population. This yields new individual strings as offspring, which share some features with each parent. The least-fit individuals are kept out from reproduction and so they "die out". A whole new population of possible solutions to the problem is generated by selecting the high-fit individuals from the current generation. This new generation contains characteristics, which are better than their ancestors. The parameters which represent a potential solution to the problem, *genes*, are joined together to form a string of values referred as a *chromosome*. A decimal coding system has been adopted for coding the *chromosomes* in the present work. The network configuration chosen for the present work is 6-6-5. Therefore, the number of weights (*genes*) that are to be determined are 6X6+6X5= 66. With each *gene* being a real number, and taking the *gene* length as 5, the string representing the *chromosomes* of weights will have a length of 66X5=330. This string represents the weight matrices of the input-hidden layer-output layers. An initial population of *chromosomes* is randomly generated. Weights from each *chromosome* have been extracted then using the procedure suggested in reference. A constant learning rate of 0.6 and a momentum factor of 0.9 have been adopted during the training. Satisfactory training has been obtained after just 1500 training cycles.

The learning of the network model is presented in Figures 2(a-e). These figures are presented only for 10 data sets only. However, the author has verified all the hundred data sets used for training and found that the network has learned the beam design problem satisfactorily. From the figures 2(a-e), it can be observed that the hybrid neural network model is able to predict the depth, reinforcement, spacing of stirrups correctly for the problems in the training set.

Validation of the hybrid network model

Validation of the network is to test the network for the parameters that are not used in training of the network. The GA/BPN model was asked to predict the length, breadth, depth of footing and areas of reinforcement along length and breadth of footing for ten new problems, which are not included in the training set. It can be seen that from Figs.3(a-e), that the values predicted by GA/ BPN model for new sets match satisfactorily with results of design experts.

IV. CONCLUSION

In this paper, the application of genetic algorithm based hybrid neural networks for the design of footing problem has been demonstrated. The hybrid network model has been trained using one hundred examples obtained from different design experts. The training examples are so chosen that they will cover all the design variables involved in the problem. The weights for the network have been obtained using a genetic algorithm. The network could learn the column design problem with just 1500 training cycles. After successful training, the neural network model is able to predict the length, breadth, depth of footing and areas of reinforcement along length and breadth of footing satisfactorily for new footing problems. Thus, it is concluded that the developed neural network model can provide a safe and economical design for the design of footings.

REFERENCES

[1]. Rajasekharan, S., & Vijayalakshmi Pai,G.A., *Neural Networks, Fuzzy Logic and genetic Algorithms*, pp. 34-56,305-314, Prentice Hall of India , New Delhi,2003 .

[2]. Davis, L. *Hand book of genetic algorithms*. Van Nostrand Reinholt, New York. 1991.

[3]. Ni Hong-Guang , & Wang Ji-Zong., Prediction of Compressive Strength of Concrete by Neural Networks, *Cement and Concrete Research*, Vol.30, 2000, pp. 245-1250.

[4]. Sanad, A., & Saka, M.P., Prediction of Ultimate Strength of Reinforced Concrete Deep Beams by Neural Networks, *ASCE Journal of Structural Engineering*, Vol. 127, No. 7, 2001, pp. 818-828,

[5]. Cladera, A., & Mari, A.R., Shear Design Procedure for Reinforced Normal and High Strength Concrete Beams using Artificial Neural Networks Part II: Beams With Stirrups, *Engineering Structures*. Vol. 26, 2004, pp. 917-926.

[6]. Cladera, A., & Mari, A.R., Shear Design Procedure for Reinforced Normal and High Strength Concrete Beams using Artificial Neural Networks. Part I: Beams Without Stirrups, *Engineering Structures*, Vol. 26, 2004, pp. 927-936.

[7]. Hadi, N.S., Neural Networks Applications in Concrete structures, *Computers and Structures*. Vol. 81**,** 2002, pp. 373-381.

[8]. Ghaboussi, J., & Joghatie., Active Control of Structures Using Networks, *A.S.C.E Journal of structural engineering*. Vol. 121, No. 4, 1995, pp.555-567.

[9]. Mukharjee, A., & Deshpande., Application of Artificial Neural Networks in Structural Design Expert Systems, *Computers and Structures*, Vol. 54, No. 3,1995,pp. 367-375.

[10]. Mishra, A.K., & Akhil Upadhyay., Column Design using ANN, *ICI Journal*, 2004, pp. 17-19.

[11]. Rumel hart, D.E., Hinton, G.E., & Williams, R.J., Learning into representations by error propogation. In D.E Rumelhart. & McClelland (Eds), Parallel distributed processing: explorations in microstructure of cognition *Cambridge. MA: Press*. , 1986, pp. 318-362.
[12]. Jenkins, W.M., Plane Frame Optimum Design Environment Based On Genetic Algorithm, *Journal of Structural Engineering*. Vol. 118**,** No.11, 1992, pp.3103-3112.
[13]. Jenkins, W.M., Towards Structural Optimization Via The Genetic Algorithm, *Computers and Structures*.Vol. 40, 1991, pp. 1321-1327.
[14]. Leite, J.P.B., & Topping, B.H.V., Improved Genetic Operators For Structural Engineering Optimization *Advances in engineering software,* Vol. 29, No. 7-9, 1998, pp. 529-562.
[15]. Topping, B.H.V., & de Barros Leite, J.P., Parallel genetic Models for Structural Optimization. *International Journal of Engineering Optimization*, Vol. 31**,** No.1, 1998, pp. 65-99.
[16]. Code of Practice for *Plain and Reinforced Concrete* I.S 456 – 2000.

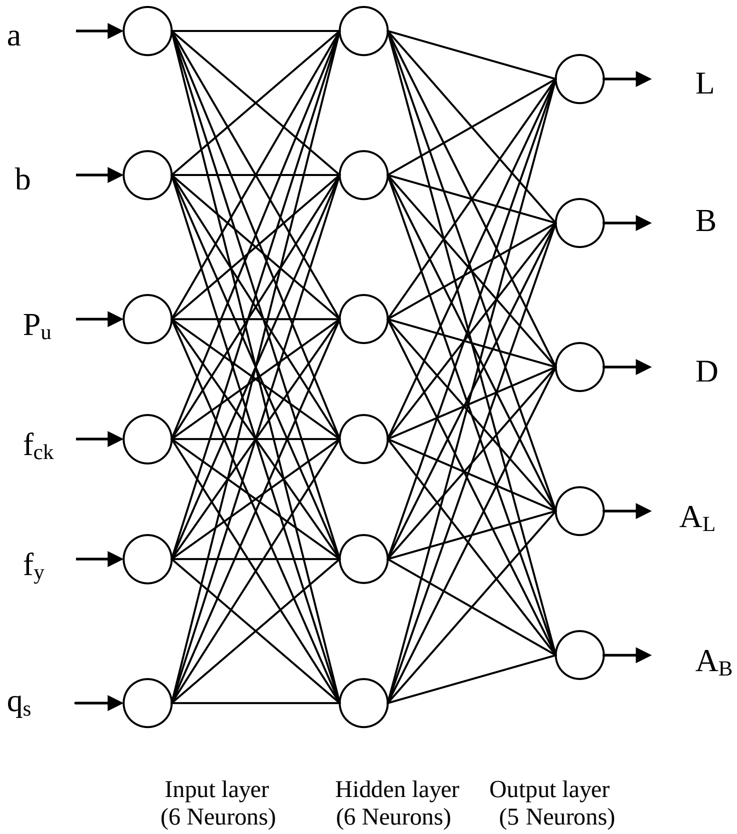

Fig. 1. Configuration of GA/BPN Model

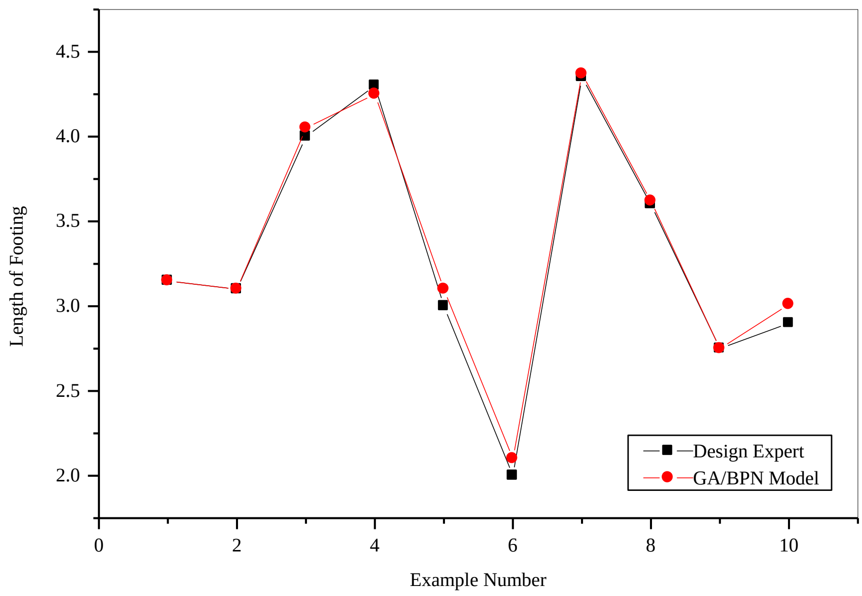

Fig. 2(a). Learning of the GA/ BPN Network Model for Length of Footing

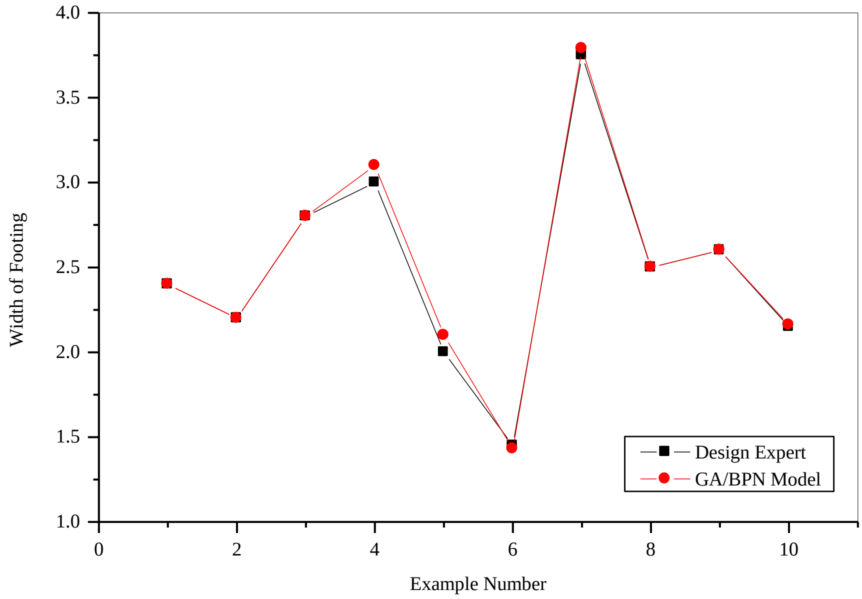

Fig. 2(b). Learning of the GA/ BPN Network Model for Width of Footing

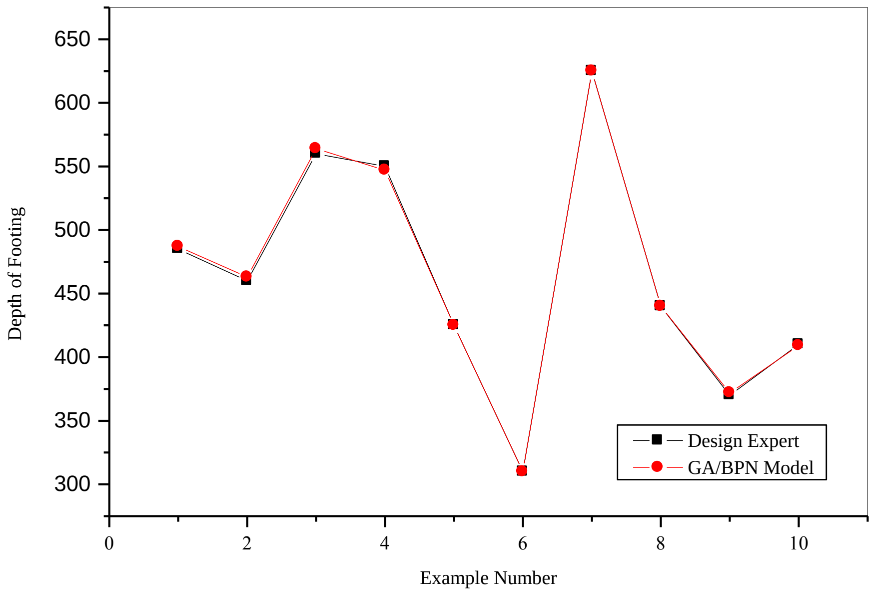

Fig. 2(c). Learning of the GA/BPN Network Model for Depth Footing

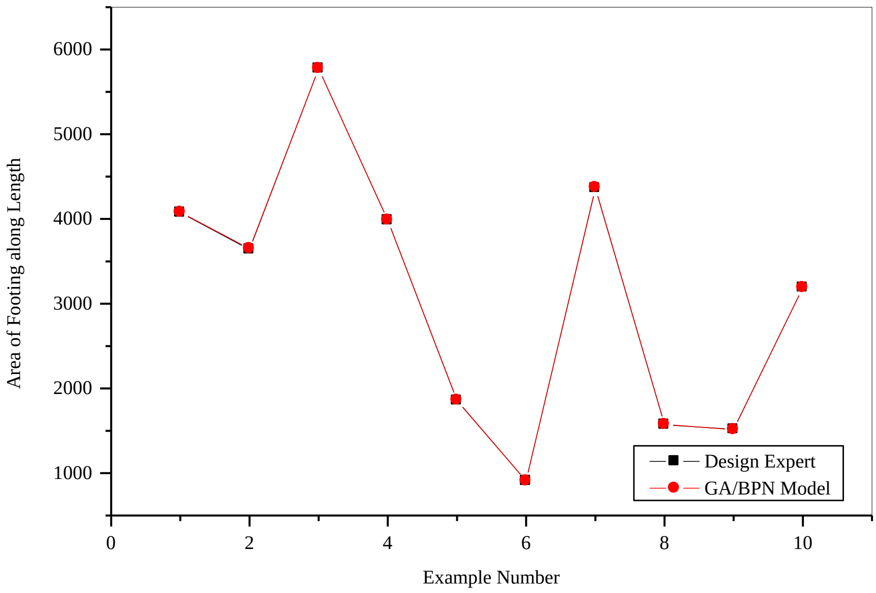

Fig. 2 (d). Learning of the GA/ BPN Network Model for Area of Footing along Length

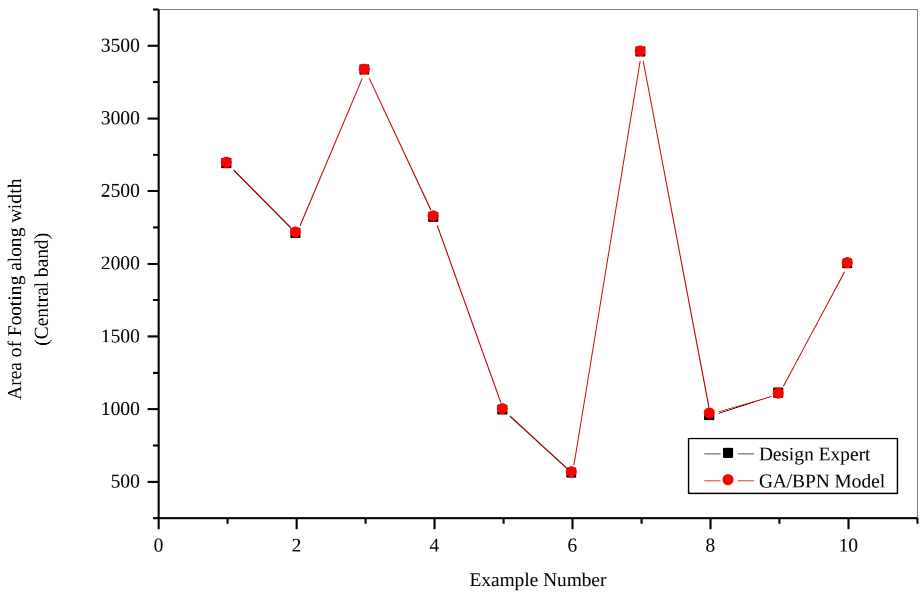

Fig. 2 (e). Learning of the GA/ BPN Network Model for Area of Footing along width central band

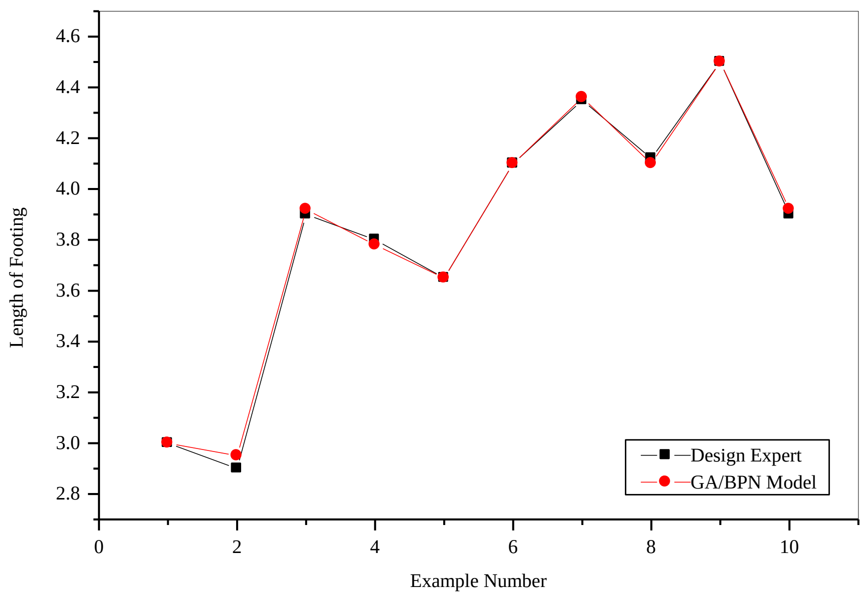

Fig. 3 (a). Validation of the GA/ BPN Network Model for Length of Footing

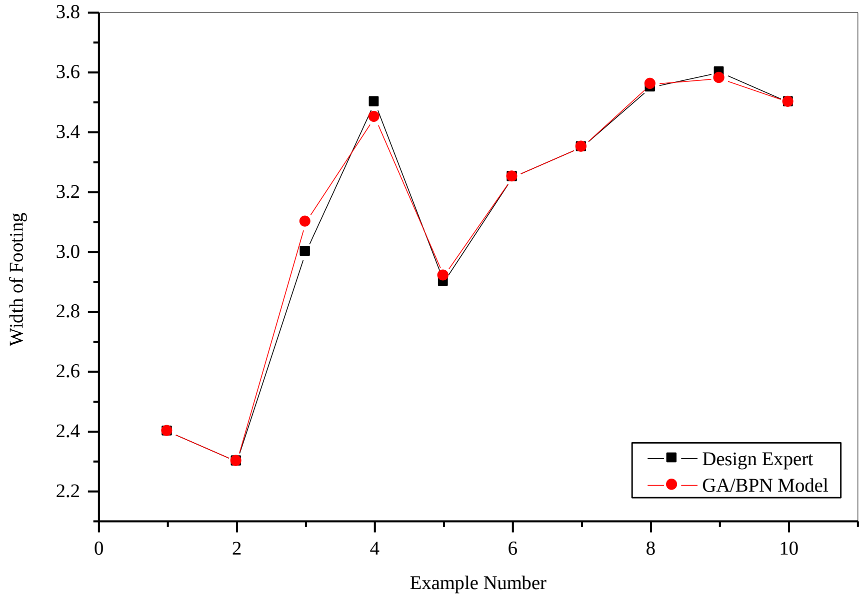

Fig. 3(b). Validation of the GA/ BPN Network Model for Width of Footing

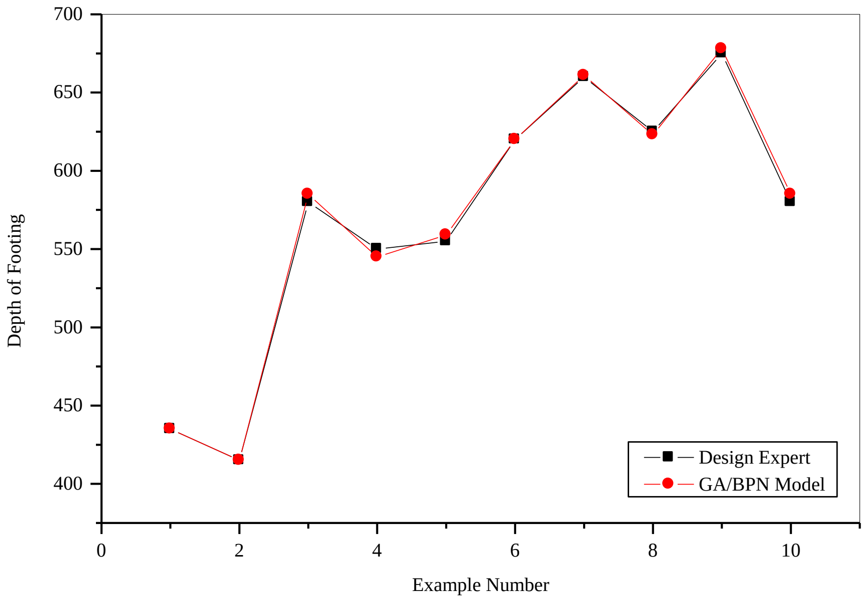

Fig. 3(c). Validation of the GA/ BPN Network Model for Depth Footing

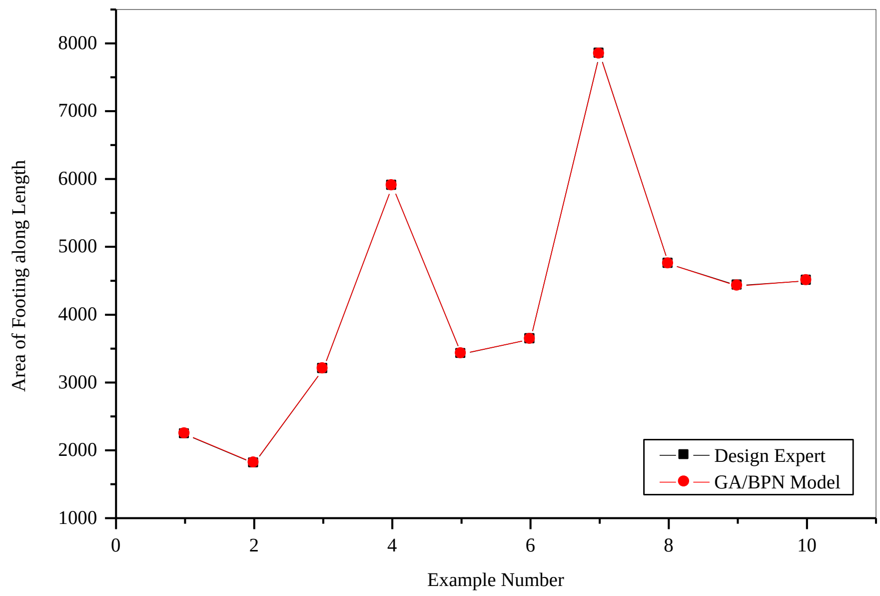

Fig. 3(d). Validation of the GA/BPN Network Model for Area of Footing along Length

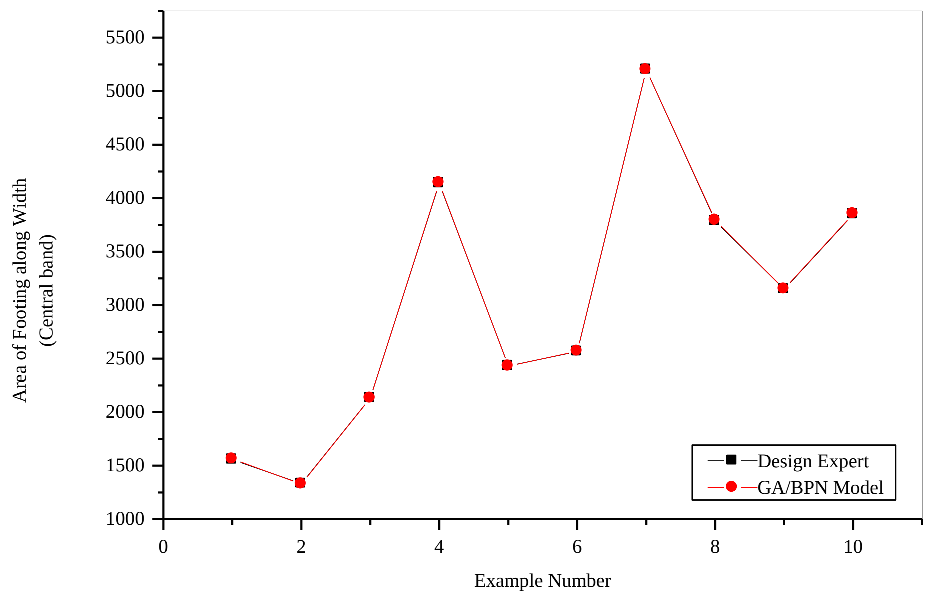

Fig. 3(e). Validation of the GA/BPN Network Model for Area of Footing along width central band