

Neural Network Model for Drag coefficient and Nusselt number of square prism placed inside a wind tunnel

Mostafa Y.El-Bakry ,D.M.Habash and Mahmoud Y.El-Bakry

Abstract— The neural network simulation has been designed to simulate and predict the Drag coefficient and Nusselt number of square prism placed inside a wind tunnel . The system was trained on the available data of the two cases. Therefore, we designed the system to work in automatic way for finding the best network that has the ability to have the best test and prediction. The proposed system shows an excellent agreement with that of an experimental data in these cases.

Index Terms— Neural networks, Drag coefficient, Nusselt number, square prism , wind tunnel.



1 INTRODUCTION

Many investigations have been carried out for heat transfer by forced convection between the exterior surface of bluff bodies such as spheres ,cylinder, square ,triangle and rectangular prisms. The important characteristics of flow over a bluff body lie in the nature of the boundary layer .As the streamlines pass over a bluff body , separation takes place due to excessive loss of momentum at adverse pressure gradient from a point , which is not far from the leading edge of the bluff body . The study of heat transfer from a bluff body is important in number of fields such as heat exchanger, gas turbine blades ,hot wire anemometry and cooling of electronic equipments. By extensive search of literature it is revealed that fluid flow over different shaped bluff bodies like square, triangle, circular, rectangular and toroids has been investigated thoroughly [1-8]. Literature is also available for heat transfer from different geometric shaped bluff bodies at various conditions like various angles of attack and Reynolds numbers[9-19].

The experimental investigations are carried out to determine the pressure and Drag coefficients from the measurement of the pressure distributions around the square prism and the average heat transfer rates of the prism as influenced by Reynolds number , angle of attack and ratio of the distance the centroid of the bluff body from the upper wall to the height of the test section in the wind tunnel [20].

The present effort introduce the artificial neural network (ANN) for modelling the Drag

coefficient and the Nusselt number of square prism placed inside a wind tunnel using the data obtained from Chakrabarty D,Brahma [20].Neural networks are widely used for solving many problems in most science problems of linear and non-linear cases [21-30]. Neural network algorithms are always iterative, designed to step by step minimise (targeted minimal error) the difference between the actual output vector of the network and the desired output vector [31-33]. The data obtained by [20] is chosen to be carried out using the neural networks .The present work offers neural network to simulate and predict the unknown data of the Drag coefficient C_d and the Nusselt number Nu_x of square prism placed inside a wind tunnel as a function of angle of attack α at different height ratios .The rest of paper is organized as follows; Sec. 2 describes the Artificial neural network model (ANN). Section 3 presents the proposed ANN. Section 4 shows the obtained results. Finally, Sec. 5 concludes the work.

2 Artificial neural network Model

Bourquin et al. [34,35] and Agatonovic-Kustrin and Beresford[36] described the basic theories of ANN model. An ANN is a biologically inspired computational model formed from several of single units, artificial neurons, connected with coefficients (weights) which constitute the neural structure. They are also known as processing elements (PE) as they process information. Each PE has weighted inputs, transfer function and one output. PE is essentially an equation which balances inputs and outputs.

Figure (1).can be expressed for Neuron Model as follows

*Physics Department, Faculty of Science, Benha University, Egypt

**Physics Department, Faculty of Education Ainshams University

***Physics Department, Faculty of Science University of Tabuke-mail: oelbakre13@yahoo.com

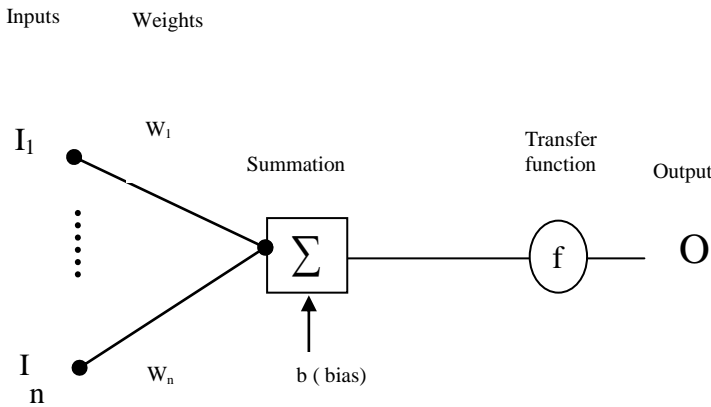


Figure (1). Neuron Model

The neuron transfer function, f , is typically step or sigmoid function that produces a scalar output (O) as follows

$$O = f \left(\sum_i W_i I_i + b \right) \quad (1)$$

where I_i , W_i and b are i th input, i th weight and bias, respectively.

The following definitions are necessary, NI, NH and NO =the number of nodes in the input, hidden and output layers, respectively,

$I(i, r)$ = the i th input value, $i \in [1, NI]$; in the r th input pattern, $r \in [1, P]$,

$W1(i, j)$ = the weight connecting the i th input value to the j th hidden neuron,

$B1(j)$, $j \in [1, NH]$ = the bias associated with the j th hidden neuron,

$H(j, r)$ = the output of the j th hidden neuron, $j \in [1, NH]$, for the r th input pattern,

$W2(j, k)$ = the weight connecting the j th hidden neuron to the k th output neuron,

$B2(k)$, $k \in [1, NO]$ = the bias associated with the k th output neuron,

$O(k, r)$ = output of the k th output neuron, $k \in [1, NO]$, for the r th input pattern,

$T(k, r)$ = target of the k th output neuron, $k \in [1, NO]$, for the r th input pattern,

Y = all weights and biases for the whole NN which starts with the random values.

The output of the j th hidden neuron at the r th input pattern is given by:

$$H(j, r) = f \left[\sum_{i=1}^{NI} W1(i, j) I(i, r) + B1(j) \right] \quad (2)$$

where f , is an approximate transfer function.

Typical transfer functions are the hyperbolic tangent function defined as:

$$f_1(\theta) = \tanh(\theta) \quad (3)$$

and the linear function defined as:

$$f_2(\theta) = \theta \quad (4)$$

Similarly, the output of the k th output neuron is given by:

$$O(k, r) = f \left[\sum_{j=1}^{NH} W2(j, k) H(j, r) + B2(k) \right] \quad (5)$$

The NN output O , is required to mimic a target output T . To achieve that, the NN is trained to find an approximate set of weights and biases Y , which minimizes an index E defined as:

$$E = \sum_{k=1}^{NO} \sum_{r=1}^P [O(k, r) - T(k, r)]^2 \quad (6)$$

There are many types of neural networks designed by now and new ones are invented every week but all can be described by the transfer functions of their neurons, by the training or learning algorithm (rule), and by the connection formula. A single-layer neuron is not able to learn and generalize the complex problems. The multilayer perceptron (MLP) overcomes the limitation of the single-layer perceptron by the addition of one or more hidden layer(s) Fig.(2). The MLP has been proven to be a universal approximator Cybenko [37]. In Fig. (2), a feedforward multilayer perceptron network was presented. The arriving signals, called inputs, multiplied by the connection weights (adjusted) are first summed (combined) and then passed through a transfer function to produce the output for that neuron. The activation (transfer) function acts on the weighted sum of the neuron's inputs and the

most commonly used transfer function is the sigmoid (logistic) function. The way that the neurons are connected to each other has a significant impact on the operation of the ANN (connection formula). There are two main connection formulas (types): feedback(recurrent) and feedforward connection. Feedback is one type of connection where the output of one layer routes back to the input of a previous layer, or to same layer. Feedforward does not have a connection back from the output to the input neurons. There are many different learning rules (algorithms) but the most often used is the Delta-rule or backpropagation (BP) rule.

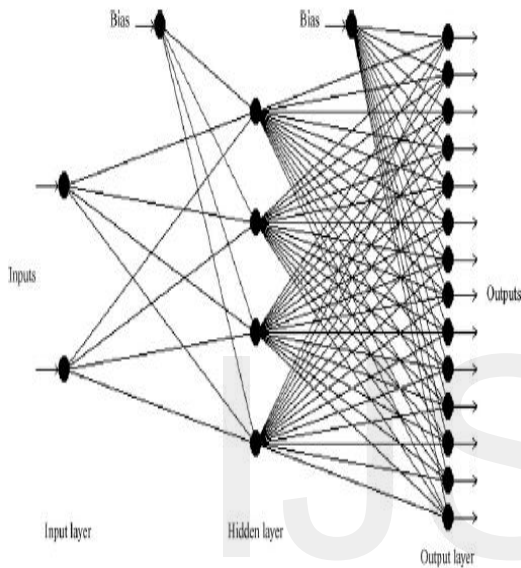


Fig.(2). Schematic representation of a multilayer perceptron feedforward network consisting of two inputs, one hidden layer with four neurons and 14 outputs.

A neural network is trained to map a set of input data by iterative adjustment of the weights. Information from inputs is fed forward through the network to optimize the weights between neurons. Optimization of the weights is made by backward propagation of the error during training or learning phase. The ANN reads the input and output values in the training data set and changes the value of the weighted links to reduce the difference between the predicted and target (experimental) values. The error in prediction is minimized across many training cycles (iteration or epoch) until network reaches specified level of accuracy. A complete round of forward-backward passes and weight adjustments using all input-output pairs in the data set is called an epoch or iteration. In this study, we focused on the learning situation known as supervised learning, in which a set of input/output data patterns is available. Thus, the

ANN has to be trained to produce the desired output .

An algorithm is employed to minimize the index E over Y, employing gradients estimated using the partial derivatives of E with respect to Y. The gradients are determined, employing the backpropagation technique which involves performing computations backward in the network [38]. The training is performed employing the Levenberg - Marquardt algorithm (LMA) [39]. The LMA employs a Newton - like update in the form,

$$Y_v = Y_{v-1} - \left[J^t J - \mu \tilde{I} \right]^{-1} J^t e, \quad (7)$$

where J is the Jacobian matrix which contains the first derivatives of the NN errors with respect to the weights and biases, e is a vector of NN errors, μ is a scalar changed adaptively by the algorithm (the adaptation constant), V is the iteration number, t denotes transposition and \tilde{I} is the identity matrix. The term between brackets in the right hand side of Eq.(7) is an approximation of the Hessian matrix, while the term after it is the gradient.

In order to perform a supervised training we need a way of evaluating the ANN output error between the actual and the expected output. A popular measure is the mean squared error (MSE) or root mean squared error (RMSE):

$$MSE = \frac{\sum(y_i - o_i)^2}{n} \quad (8)$$

$$RMSE = (MSE)^{1/2} \quad (9)$$

where y_i is the predicted value, O_i the observed value, and n is the number of data set.

3 The proposed ANN model for Drag coefficient and Average Nusselt Number

Drag coefficient C_D and Average Nusselt Number N_u can be simulated and predicted at different inputs using ANN. authors choose to internally model the problem with two individual neural networks trained separately using experimental data. The first ANN was configured to have angle of attack α and height ratio Y/h (0.5,0.167 and 0.217) as inputs while the output is Drag coefficient C_D . The second

ANN was configured to have angle of attack α and height ratio Y/h (0.5,0.35 and 0.217) as inputs while the output is Average Nusselt Number Nu_x . Fig.(3) represent a block diagram of the two ANN based modeling.

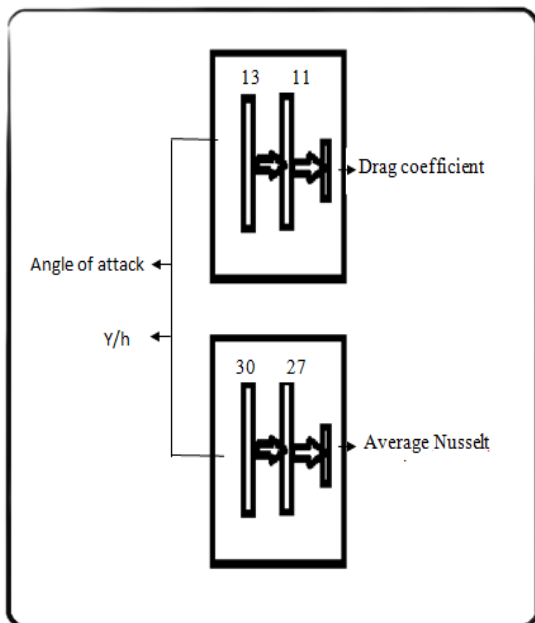


Fig.(3): Block diagram of the two ANN based modeling.

The performance of the previous two models is examined by using the mean square error (MSE) and the mean absolute error (MAE) which are defined as follows.

$$MSE = \frac{1}{N} \sum_{i=1}^N (E_{i_{measured}} - E_{i_{predicted}})^2 \tag{10}$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |E_{i_{measured}} - E_{i_{predicted}}| \tag{11}$$

In Equations. (10) and (11), the E denotes the RMS Drag coefficient C_d in first case and Nusselt Number in second case and N is the total number of data.

In general, a good model will produce less error measurements (in our case MSE and MAE). The MSE and MAE were calculated for both the training and testing (prediction) data sets for different number of hidden nodes to avoid the over-fitting problem. The error calculated using the testing data set often provides a better measurement of the predictive capability of a fitted model. Thus, computing the error over the

testing data set becomes more helpful to validate the model and avoid over-fitting.

4 Results

The proposed ANN models were applied to simulate the experimental data [20] of the Drag coefficient C_d (referred to as model1) and Nusselt Number Nu_x (model 2). By employing the above mentioned proposed models with different values of the ANN parameters we have obtained different numbers of hidden neurons for the ANN models. The results obtained by the two models are discussed in the following :

The first ANN having two hidden layers of 13 and 11 neurons (model1) and second network having 30 and 27 neurons(model2) respectively with one neuron in the output layer. Network performance was evaluated by plotting the ANN model output against the experimental data and analyzing the percentage error between the simulation results and the experimental data Fig. (4). In the training process, 74 and 461 epochs was found to be sufficient, Fig. (4), with respect to the minimum error of 9.6×10^{-6} and 3.1×10^{-7} errors (mean sum of square error MSE) respectively. For all networks, the function which describes the nonlinear relationship is given in appendix.

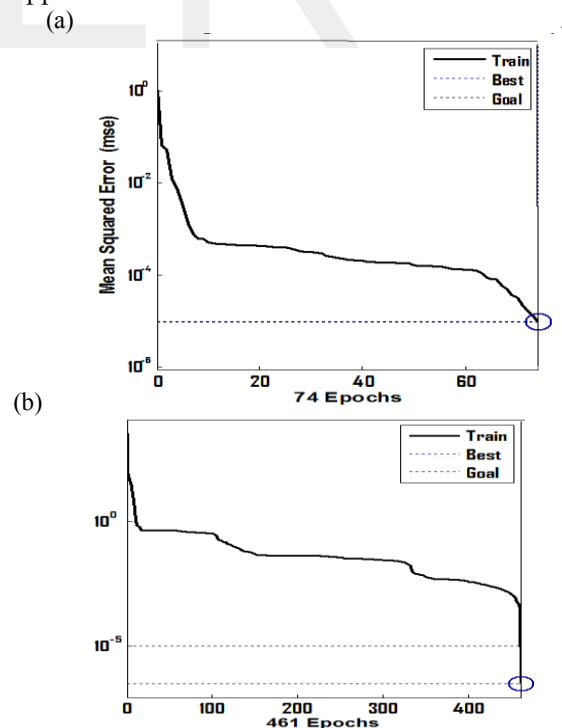


Fig. (4): Performance of the obtained ANN model:

- (a) Drag coefficient (model1) ,
- (b) Nusselt number (model2).

The above mentioned details of the proposed ANN model (model1) are carried out and simulated two the experimental data of the angle of attack α and Drag coefficient C_d using the obtained function which is given in appendix. the proposed C_d is trained using ANN model on three cases of height ratio Y/h . The values of these cases are 0.167, 0.217 and 0.5 Fig.(5). After the training, the obtained system is predicted the behavior of height ratio $Y/h=0.083$ Fig(5). It is found that as shown in Fig (5), the obtained results (simulated and predicted) are provided to demonstrate good agreement with the experimental data[20].

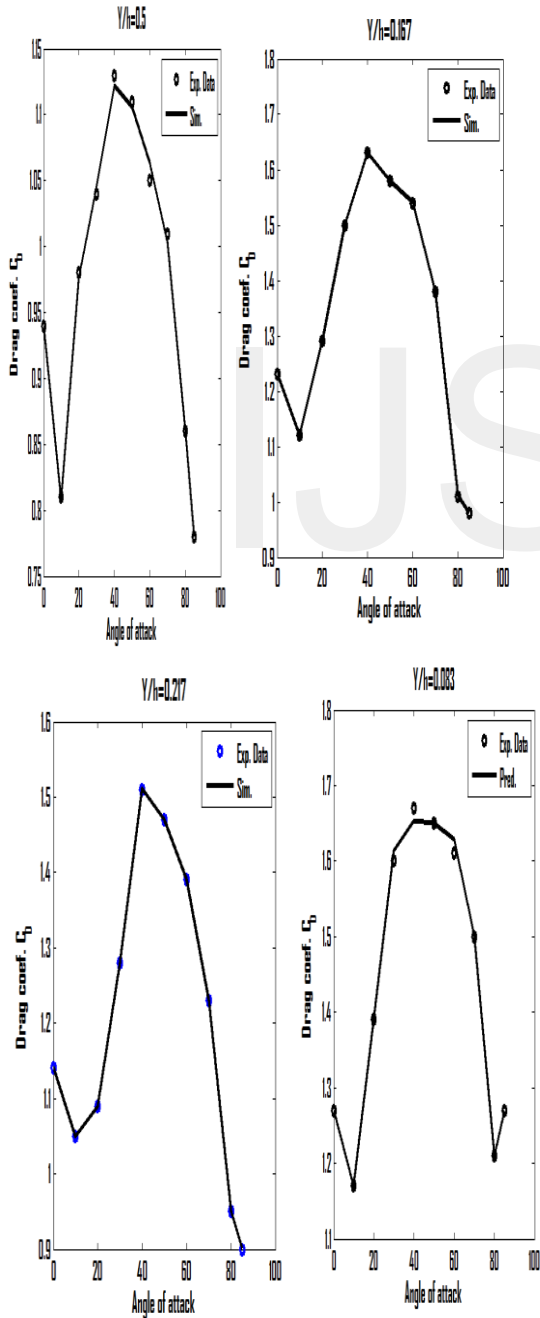


Fig. (5). ANN simulation and prediction of Drag coefficient.

Also, ANNs are chosen to be applied on the average Nusselt number Nu_x (model2) at different values of height ratio Y/h . The training values Y/h are 0.217, 0.35 and 0.5 as shown in Fig(6). The predicted value of $Y/h=0.167$, is in Fig(6). The simulation and predicted results from the obtained function which is given in appendix are best fitting with the experimental data. It is noted (Fig(5) Fig(6)) that the proposed ANN model shows excellent results matched well the experimental data.

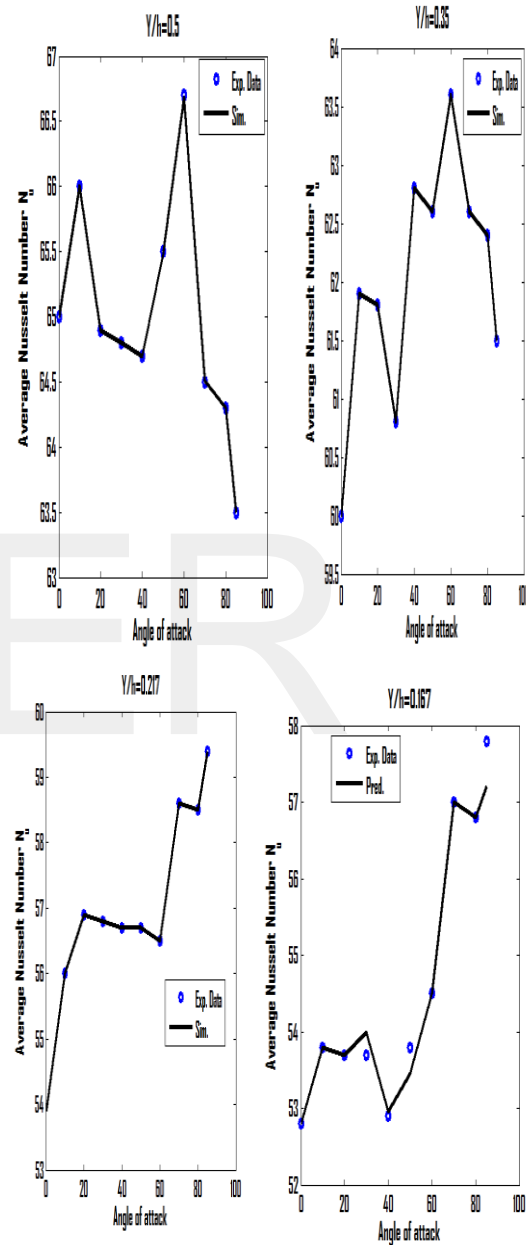


Fig. (6). ANN simulation and prediction of Nusselt Number.

5 Conclusion

In this paper, a method was proposed to model the the Drag coefficient C_d and the Nusselt number Nu_x of square prism placed inside a wind tunnel as a function of angle of attack α at different height ratios using ANN

approach. The trained ANN network shows excellent results matched with the experimental data in the two cases of the Drag coefficient C_d and the Nusselt number Nu_x . The designed ANN introduce a powerful model and shows a good match to the experimental data. Then, the capability of the ANN techniques to simulate and predict the experimental data with almost exact accuracy recommends the ANN to dominate the modelling techniques in physics of fluid.

Appendix

The equation which describe Drag coefficient and Nusselt Number is given by:

C_D and $Nu_x = \text{pureline}[\text{net.LW}\{3,2\} \text{logsig}(\text{net.LW}\{2,1\} \text{logsig}(\text{net.IW}\{1,1\}A + \text{net.b}\{1\}) + \text{net.b}\{2\}) + \text{net.b}\{3\}]$

Where



A is the input

net . IW{1,1} : linked weights between the input layer and first hidden layer.

net . LW{2,1} : linked weights between the two hidden layers.

net . LW{3,2} : linked weights between the second hidden layer and output layer.

net . b{1} : is the bias of the first hidden layer.

net . b{2} : is the bias of the second hidden layer.

net . b{3} : is the bias of the output layer.

References:

1. Bearman PW (1965) Investigation of flow behind a two dimensional model with blunt trailing edge and fitted with splitter plates. *J Fluid Mech* 21(2):241-255.
2. Bearman PW, Trueman DM (1972) An investigation of the flow around rectangular cylinders. *Aeronautical Q* 23:229-237.
3. Aplet CJ et al (1973), The effects of wake splitter plates on the flow past a circular cylinder in the range $10^4 < Re < 5 \times 10^4$. *J Fluid Mech* 61(1):187-198.
4. Bearman PW, Zdravkovich MM (1978) Flow around a circular cylinder near a plane boundary. *J Fluid Mech* 89(1):53-73.
5. Igarashi T (1984) Characteristics of the flow around a square prism. *Bull JSME* 27(231):1858-1864.
6. Igarashi T (1985) Characteristics of flow around rectangular cylinders. *Bull JSME* 28(242):1690-1696.
7. Mansingh V, Oosthuizen PH (1990) Effects of splitter plates in the wake flow behind a bluff body. *AIAA J* 28:778-783.
8. Simpson RL (1991) The structure of the near wall region of two dimensional turbulent separated flow. *Philos Trans R Soc Lond* 336:5-17.
9. Igarashi T (1974) Heat transfer in separated flows. In: 5th International heat transfer conference, Tokyo, pp 300-304.
10. Igarashi T, et.al (1975) Heat transfer in separated flows (part1, experiments on local heat transfer from the rear of a flat plate to an air steam). *Heat Transf Jpn Res* 4(1):11-32.
11. Igarashi T, Hirata M (1977) Heat transfer in separated flows (part 3, the case of equilateral triangular prisms). *Heat Transf Jpn Res* 6(4):13-39.
12. Igarashi T (1977) Heat transfer in separated flows, part 2: theoretical analysis. *Heat Transf Jpn Res* 6(3):60-78.
13. Igarashi T (1982) Fluid flow and heat transfer in the separated region of a circular cylinder with wake control. *Heat Transf Jpn Res* 11(3):1-16.
14. Igarashi T (1984) Correlation between heat transfer and fluctuating pressure in separated region of a circular cylinder. *Int J Heat Mass Transf* 27(6):927-937
15. Igarashi T (1985) Heat transfer from a square prism to an air stream. *Int J Heat Mass Transf* 28(1):175-181.
16. Aiba S, Tsuchoda H (1985) Heat transfer around a circular cylinder near a plane boundary. *Trans JSME*, 51-463, 866-873.
17. Igarashi T (1986) Local heat transfer from a square prism to an air stream. *Int J Heat Mass Transf* 29(5):777-784.
18. Igarashi T (1987) Fluid flow and heat transfer around rectangular cylinders. *Int J Heat Mass Transf* 30(5):893-901
19. Yovanovich MM, Refai, Ahmed G (1997) Experimental study of forced convection from isothermal circular and square cylinders and toroids. *J Heat Transf Trans ASME* 119:70-79
20. Chakrabarty D, Brahma R (2008) Fluid flow and heat transfer characteristics for a square prism (blockage ratio = 0.1) placed inside a wind tunnel, *J Heat and Mass Transf* 44(3):325-330
21. Mostafa Y. El-Bakry (2011) Radial Basis Function Neural Network Model for Mean velocity and Vorticity of Capillary Flow, *Inter. J. for numerical methods in fluids* 67: 1283-1290
22. El-Bakry Mostafa Y., El-Bakry Mahmoud Y. and Elhelly M. (2009) Neural Network Representation For the Forces and Torque of The Eccentric Sphere Model, *Trans.on Comput.Sci.III, LNCS 5300, pp171-183.*

23. El-Bakry Mostafa Y., El-Harby A. A. and Behery G. M. (2008) Automatic Neural Network System for Vorticity of Square Cylinders With Different Corner Radii, *Journal of Applied Mathematics And Informative(JAMI)*, Vol.26, No.5-6, pp911-923.
24. Elbakry Mostafa Y. and Radi A., (2007), Genetic Programming Approach For Flow Of Steady State Fluid Between Two Eccentric Spheres, *Appl. Rheol.* 17:6, 68210.
25. El Bakry M.Y., and El-Metwally K. A. (2003) Neural Network for Proton- Proton Collision at High Energy, *Chaos, Solitons and Fractals*, vol. 16, No. 2, pp. 279-285.
26. El Bakry M.Y. (2003) Feed Forward Neural Networks Modeling for K-P Interactions, *Chaos, Solitons and Fractals*, vol. 16, No.2, pp. 279-285.
27. Latafat A. Gardashova, (2014), Application of DEO Method to Solving Fuzzy Multiobjective Optimal Control Problem, *Applied Computational Intelligence and Soft Computing*, Article ID 971894, 7 pages, Volume 2014 .
28. Scalabrin G., Corbetti C. and Cristofoli G., (2001) A Viscosity Equation of state for R123 in the form of a Multilayer feed forward Neural Network, *Inter.J. of Thermophyscis*, vol. 22, No.5.
29. El-dahshan E., Radi A. and El-Bakry M. Y. (2008) artificial Neural Network and Genetic Algorithm Haybrid Technique for Nucleus-Nucleus Collisions; *Int. J. Mod Phys C* 19, 1787.
30. El-dahshan E., Radi A. and El-Bakry M. Y. (2009) Genetic Programming Modeling for Nucleus-Nucleus Collisions; *Int. J. Mod Phys C* 20, 1817.
31. Yi-Chung Hu, and Jung-Fa Tsai (2006) Backpropagation multi-layer perceptron for incomplete pairwise comparison matrices in analytic hierarchy process, *Applied Mathematics and Computation*, vol. 180, No. 1, pp. 53-62.
32. Curry B., and Morgan P.H. (2006) "Model selection in Neural Networks: Some difficulties", *European Journal of Operational Research*, vol. 170, No. 2, pp. 567-577.
33. Jochen J. Steil (2006) Online stability of Backpropagation-decorrelation recurrent learning Neurocomputing, vol. 69, No. 7-9, pp. 642-650.
34. Ghaffari A. et al. ,*International Journal of Pharmaceutics* (2006) 327,pp 126-138.
35. Bourquin, J., Schmidli, H., Hoogevest, P.V., Leuenberger, H. (1997) Basic concepts Of artificial neural networks (ANN) modeling in the application to pharmaceutical development. *Pharm. Dev. Tech.* 2,pp 95-109.
36. Agatonovic-Kustrin, S., Beresford, R. (2000) Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *J. Pharm. Biomed. Anal.* 22, pp 717-727.
37. Cybenko, G. (1989) Approximation by superposition of a sigmoidal function. *Math. Control Signals Syst.* 2,pp 303-314.
38. Lippmann,R. (1987) An Introduction to Computing With Neural nets, *IEEE Trans. Acoust., Speech and signal processing* 4(2), pp4-22.
39. Hagan, M.T. and Menhaj, M.B. (1994) Training feed forward network with the Marquardt algorithm , *IEEE Trans. on Neural Networks*, 5(6), pp.861-867.