Journal of Applied Sciences Research, 5(6): 662-670, 2009

Artificial Neural Network Model for Prediction of Friction Factor in Pipe Flow

D.A. Fadare and U.I. Ofidhe

Department of Mechanical Engineering, University of Ibadan, PMB 1, Ibadan, Nigeria

Abstract: Determination of friction factor is an essential prerequisite in pipe flow calculations. The Darcy-Weisbach equation and other analytical models have been developed for the estimation of friction factor. But these developed models are complex and involve iterative schemes which are time consuming. In this study, a suitable model based on artificial neural network (ANN) technique was proposed for estimation of factor to friction in pipe flow. Multilayered perceptron (MLP) neural networks with feed-forward back-propagation training algorithms were designed using the neural network toolbox for MATLAB®. The input parameters of the networks were pipe relative roughness and Reynold's number of the flow, while the friction factor was used as the output parameter. The performance of the networks was determined based the mean on absolute percentage error (MAPE), mean squared error (MSE), sum of squared errors (SSE), and correlation coefficient (*R*-value). Results have shown that the network with 2-20-31-1 configuration trained with the Levenberg-Marquardt '*trainlm*' function had the best performance with *R*-value (0.999), MAPE (0.68%), MSE (5.335×10^{-7}), and SSE (3.414×10^{-4}). A graphic user interface (GUI) with plotting capabilities was developed for easy application of the model. The proposed model is suitable for modelling and prediction of friction to factor in pipe flow for on-line computer-based computations.

Key words: Friction factor, pipe flow, artificial neural network, pressure head, modelling.

INTRODUCTION

The flow of liquid through a pipe is resisted by viscous shear stresses within the liquid and the rough internal walls of the pipe. This resistance, which leads to head loss, is usually measured in terms of the friction factor (*f*). The factors that affect the head loss in pipes are: the viscosity of the fluid being handled; the size of the pipes; the roughness of the internal surface of the pipes; the changes in elevations within the system; and the length of travel of the fluid. Other factors that contribute to the overall head loss are resistance through various valves and fittings. In a well designed system the resistance through valves and fittings will be of minor significance to the overall head loss, many designers choose to ignore the head loss for valves and fittings at least in the initial stages of a design [1].

Much research has been carried out over the years and various formulae for estimation of head loss have been developed based on experimental data. Among these is the Chézy formula which dealt with water flow in open channels. Using the concept of 'wetted perimeter' and the internal diameter of a pipe the Chézy formula could be adapted to estimate the head loss in a pipe [2]. Chezy proposed a relationship of the form:

$$V^2 P \alpha A S \quad (1)$$

where *P* is the wetted perimeter, *S* is the channel slope, and *A* is the area of flow. The velocity term is expressed as:

$$V = C\sqrt{RS} \quad (2)$$

where *C* is the empirical constant, *R* is the radius of the pipe.

The friction factor, *f*, is an artefact of definition, arising from the experimental observation that the pressure drop in a segment of pipe for a type of flow is proportional to the square of velocity. That is;

$$\Delta P = \left(\frac{fL}{D} + k\right)\frac{1}{2}\rho v^2 \quad (3)$$

where *f* is the friction factor, *L* is the pipe length, *D* is the pipe diameter, *k* is the form loss factor (to account for bends, entrance and exit losses, valves, orifices, etc.), ρ is the fluid density and v is the fluid velocity. The Darcy-Weisbach [3] equation is the

Corresponding Author: D.A. Fadare, Department of Mechanical Engineering, University of Ibadan, PMB 1, Ibadan, Nigeria.
Tel.: +234 (0)802 3838593;
E-mail: fadareda@yahoo.com

accepted method to calculate energy losses resulting from fluid motion in pipes and other closed conduits. The factors that influence the friction factor include: Pipe roughness, Pipe diameter, Fluid kinematic viscosity, and Velocity of flow. The complexity in f, which results from boundary layer mechanics, obscures the valid relationship between all the listed parameters, and led to the development of several irrational, dimensionally inhomogeneous, empirical formulas [4]. The Colebrook [5, 6] equation for estimation of friction factor is given in implicit form that can be written in 3 different forms as:

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left(\frac{\varepsilon}{3.7D} + \frac{2.51}{\mathrm{Re}\sqrt{f}}\right) \tag{4}$$

$$\frac{1}{\sqrt{f}} = 1.74 - 2\log_{10}\left(\frac{2\varepsilon}{D} + \frac{18.7}{\mathrm{Re}\sqrt{f}}\right) \tag{5}$$

$$\frac{1}{\sqrt{f}} = 1.14 + 2\log_{10}\left(\frac{D}{\varepsilon}\right) - 2\log_{10}\left(\frac{9.3}{\mathrm{Re}\frac{\varepsilon}{D}\sqrt{f}}\right) \tag{6}$$

Where f is the friction factor, D is the diameter of the pipe, Re is the Reynolds number, e is the roughness of the pipe.

These expressions were implicit, complex and involve iteration scheme, while is to which time consuming. As a solution to the iteration scheme of Colebrook's equation, Moody [7] developed the friction factor chart, known today as the Moody chart (Fig. 1) based on the data from the Colebrook equation. Although the chart to moody provides a solution to the implicit Colebrook's equation, the tedious nature of the graphical solution is major setback to its application.

The artificial neural network (ANN) approach provides a viable solution to the problem of prediction of friction factor in pipe flow because it is based on training not on analytical model and statistical assumptions. ANN model can be trained to predict results from examples and once trained can perform predictions at very high speed [8]. ANN is an intelligent data-driven modeling tool that is able to capture and represent complex and non-linear input/output relationships. ANNs are massively parallel, distributed processing systems that can continuously improve their performance via dynamic learning. ANNs are used in many important applications, such as function approximation, pattern recognition and classification, memory recall, prediction, optimization and noise-filtering [9]. They are used in many commercial products such as modems, image-processing and recognition systems, speech recognition software, data mining, knowledge acquisition systems and medical instrumentation, etc.

ANN is inspired after the biological neural network. As in nature, the network function is determined largely by the connections between elements. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output (Fig. 2).

The network weights are adjusted based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are needed to train a network. The power of ANN comes from its collective behaviour where all neurons are interconnected. The network starts *evolving*: neurons continuously evaluate their output by looking at their inputs, calculating the weighted sum and comparing to a threshold to decide if they should fire. This is highly complex parallel process whose features can not be reduced to phenomena taking place with individual neurons in the network. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems. Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings [10-15]. The benefits associated with ANN application include [12]:
Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

Self-organisation: An ANN can create its own organisation or representation of the information it receives during learning time.

Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Negm *et al.* [16] has applied both the multiple linear regression, and ANN for estimation of friction factor external flow over a pile of circular tubes. They established that the ANN gave a better prediction. Azimian [17] has also applied ANN to predict the friction c factor for flow inside a pipe. However, the

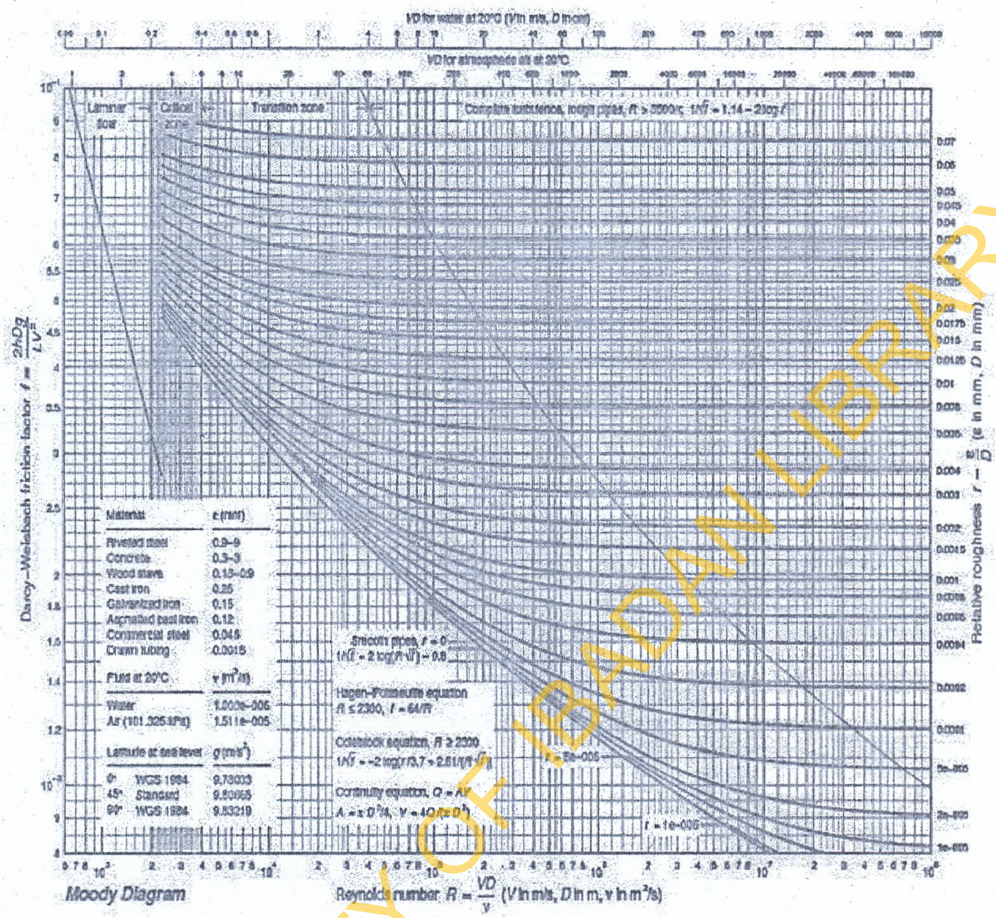

Fig. 1: The Moody chart [7]

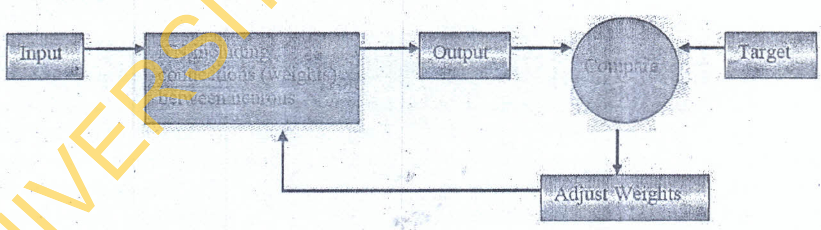

Fig. 2: Schematic of the neural network

training algorithm, performance criteria and predictive accuracy were not reported. The essence of this study was to investigate the feasibility of using ANN to model the non-linear relationship in the friction factor estimation in pipe flow. Hence, the model can be used to predict the friction factor using the pipe relative roughness and Reynolds number as input parameters. The schematic of the proposed model is shown in Fig. 3.

MATERIALS AND METHODS

2.1 Data Source: The data used in this study was generated using Colebrook's equation (equ. 4). The initial guess for the iterative scheme was determined from the expression proposed by [4] as:

$$\frac{1}{\sqrt{f}} = -1.81\log\left[\frac{6.9}{\mathrm{Re}} + \left(\frac{\varepsilon/D}{3.7}\right)^{1.11}\right] \tag{7}$$

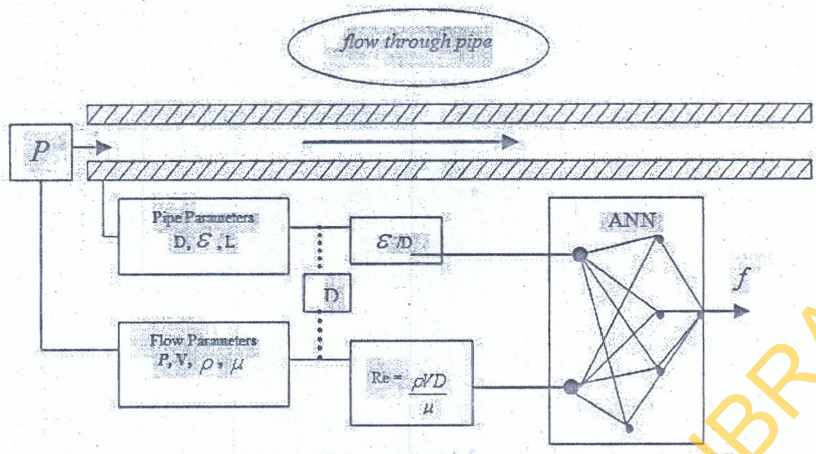

Fig. 3: Schematic of the proposed model for prediction friction factor in pipe flow [17]

Using Eqns. 4 and 7, 2,560 values of friction factor were generated for different values of relative roughness ranging from 5×10^{-6} - 7×10^{-2} and Reynolds number ranging from 2.5×10^{3} - 1.0×10^{8}. The generated 2,560 data constituted the input/output dataset used for training and testing the neural network.

2.2 Design of the ANN Model; Neural Network Toolbox for MATLABÒ [18] was used to design the neural network.

The basic steps involved in designing the network were: Collection/generation of input/output dataset; Pre-processing of data (normalization and partitioning of dataset); Design of the neural network objects; Training and testing of the neural network; simulation and prediction with new input data sets; and Analysis and post-processing of predicted result.

2.2.1 Pre-processing of Data: Prior to the training of the network, the input/output dataset was normalized using the '*premnmx*' MATLAB^{0} function. The dataset was scaled to range between 0 and 1. The normalised input/output dataset was then partitioned into two subsets consisting training dataset, 75% (1920 data), and the test dataset, 25% (640 data).

2.2.2 Design of the Network Object: Multi-layer feed-forward back-propagation hierarchical networks with different architecture were designed using the 'Neural Network Toolbox' version 4.0.2 for MATLAB [18]. The networks consisted of three layers: input layer; hidden layer; and output layer. There were two input parameters into the network: relative roughness and Reynolds number and one output parameter corresponding to the friction factor. Different networks with single or double hidden layer topologies were used. The schematic of typical network architecture is depicted in Fig. 4.

2.2.3 Training of the Neural Network: The network was trained by feeding in some teaching patterns (training dataset) and letting it change its weights according to some learning rule. Four different back-propagation training algorithms were used in training the different networks: Levenberg-Marquardt '*trainlm*', Bayesian regularization '*tranibr*', BFGS Quasi-Newton '*trainbfg*', and Cyclical order incremental '*traininc*'. The neurons with tan-sigmoid transfer function '*tansig*' were used in the hidden layer(s), while neurons with linear transfer function '*purelin*' were used in the output layer. The '*purelin*' transfer function was used so that the output would not be limited like the '*tansig*' function which generates output between 0 and +1. If linear output neurons were used, the output can take on any value.

2.2.4 Testing of the ANN Model: The training was terminated when the threshold of MSE = 0.001 or when the number of iterations is equal to 1000. The test dataset, 25% (640 data) was used to test the validity of the proposed model. The mean square error (MSE), sum of square error (SSE) and mean absolute percentage error (MAPE), and correlation coefficient (R-value) between the network predicted outputs and the desired outputs were used as the performance parameters to determine the network structure with optimal predictive capability.

RESULTS AND DISCUSSIONS

3.1 Network Optimization: The performance parameters of the different network structures trained with four different training algorithms are presented in Tables 1-4. The table shows the mean square error (MSE), sum of square error (SSE) and mean absolute percentage error (MAPE), and correlation coefficient (R-value) between the networks predicted outputs and

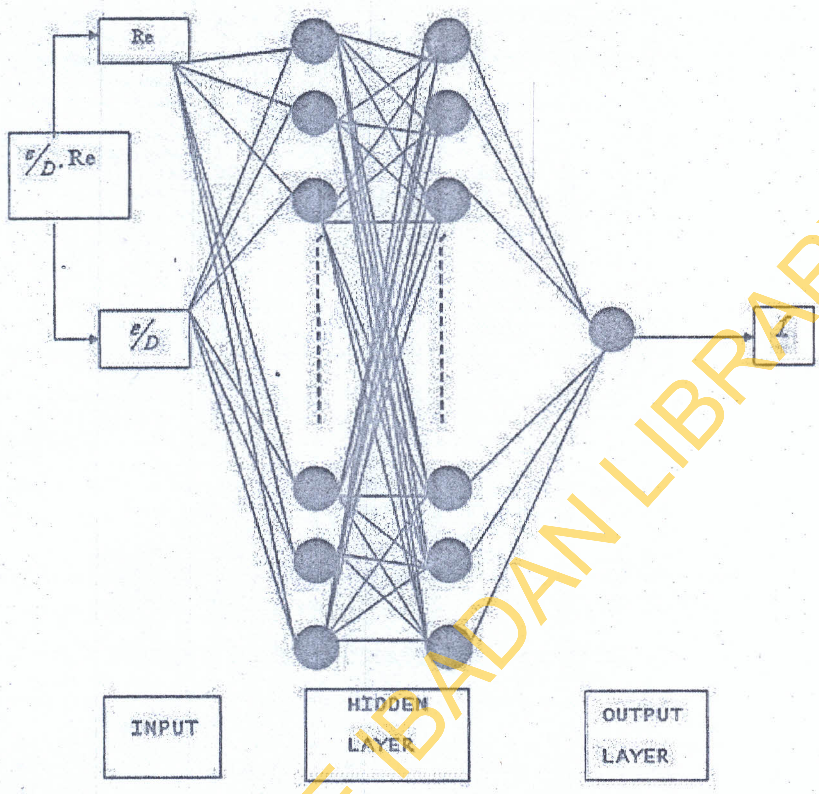

Fig. 4: A typical design of a multilayer neural network used for prediction of friction factor

Table 1: Performance parameters for different network structures trained with Bayesian regularization *'tranibr'* algorithm

Network Structure	Network performance parameters							
	Training dataset				Test dataset			
	Correlation Coefficient (*R*-value)	MSE	MAPE (%)	SSE	Correlation Coefficient (*R*-value)	MSE	MAPE (%)	SSE
2-5-1	0.996	3.566e-6	4.34	0.006	0.998	2.192e-6	5.40	0.001
2-10-1	0.997	2.534e-6	4.40	0.005	0.998	1.669e-6	3.97	0.001
2-15-1	0.992	6.981e-6	6.61	0.013	0.995	4.892e-6	6.42	0.003
2-5-5-1	1.000	1.469e-7	1.28	2.821e-4	0.999	1.483e-7	1.31	9.493e-5
2-5-10-1	1.000	1.097e-7	0.89	2.106e-4	0.999	9.429e-8	0.78	6.035e-5
2-5-15-1	1.000	4.568e-8	0.61	8.771e-5	0.999	3.522e-8	0.49	2.254e-5
2-10-15-1	1.000	1.987e-8	0.38	3.815e-5	0.999	1.840e-8	0.30	1.177e-5
2-15-20-1	1.000	4.010e-8	0.48	7.698e-5	0.999	3.059e-8	0.36	1.958e-5
2-20-30-1	1.000	3.720e-7	1.46	7.143e-4	0.999	1.881e-7	1.03	1.204e-4
2-20-31-1	1.000	1.532e-7	0.99	2.942e-4	0.999	1.265e-7	0.86	8.099e-5

Table 2: Performance parameters for different network structures trained with Cyclical order incremental '*trainc*' algorithm

Network Structure	Network performance parameters							
	Training dataset				Test dataset			
	Correlation Coefficient (*R*-value)	MSE	MAPE (%)	SSE	Correlation Coefficient (*R*-value)	MSE	MAPE (%)	SSE
2-5-1	0.935	2.743e-4	75.26	0.527	0.945	2.921e-4	74.39	0.187
2-10-1	0.840	2.057e-4	72.60	0.357	0.848	1.905e-4	71.87	0.282
2-15-1	0.906	2.099e-4	66.10	0.403	0.939	1.954e-4	60.94	0.125
2-5-5-1	0.875	3.515e-4	85.62	0.675	0.883	3.578e-4	82.35	0.229
2-5-10-1	0.911	8.950e-4	133.5	1.718	0.893	8.853e-4	126.9	0.567
2-5-15-1	0.918	1.882e-4	62.50	0.836	0.922	2.017e-4	61.41	0.328
2-10-15-1	0.865	3.705e-4	88.51	0.361	0.885	3.611e-4	83.35	0.129
2-15-20-1	0.961	1.705e-5	60.12	0.711	0.930	1.921e-5	62.25	0.231
2-20-30-1	0.942	1.221e-5	57.12	1.718	0.950	1.321e-5	57.25	0.567
2-20-31-1	0.927	2.614e-4	73.28	0.502	0.921	2.998e-4	75.37	0.192

Table 3: Performance parameters for different network structures trained with BFGS Quasi-Newton '*trainbfg*' algorithm

Network Structure	Network performance parameters							
	Training dataset				Test dataset			
	Correlation Coefficient (*R*-value)	MSE	MAPE (%)	SSE	Correlation Coefficient (*R*-value)	MSE	MAPE (%)	SSE
2-5-1	0.997	2.011e-5	11.49	0.012	0.979	1.935e-5	12.84	0.039
2-10-1	0.976	2.162e-5	15.32	0.042	0.975	2.289e-5	12.35	0.015
2-15-1	0.977	2.056e-5	12.47	0.040	0.978	2.017e-5	14.07	0.013
2-5-5-1	0.969	2.726e-5	14.57	0.052	0.967	3.088e-5	18.70	0.020
2-5-10-1	0.983	1.569e-5	10.64	0.030	0.984	1.381e-5	11.20	0.009
2-5-15-1	0.929	2.105e-5	12.20	0.040	0.976	2.255e-5	15.21	0.014
2-10-15-1	0.929	1.811e-5	10.42	0.035	0.980	1.792e-5	12.51	0.012
2-15-20-1	0.945	7.555e-6	6.52	0.015	0.985	5.539e-6	6.22	0.004
2-20-30-1	0.988	4.814e-6	5.67	0.009	0.996	3.169e-6	4.93	0.002
2-20-31-1	0.991	8.478e-6	7.58	0.016	0.992	7.366e-6	7.86	0.005

the desired outputs for the training and test datasets. The network with 2 neurons in the input layer, 10 neurons in one hidden layer, and one neuron in the output layer was designated by 2-10-1.

The network structure and the training algorithm with the best predictive performance were determined based on the performance parameters of the test dataset. Results have shown that, for networks trained with the Bayesian regularization '*tranibr*' algorithm (Table 1) the correlation coefficient (*R*-value) and mean absolute percentage error (MAPE) for the test dataset ranged between 0.995 – 0.999, and 0.30 – 6.42%, respectively. Corresponding values for networks trained with Cyclical order incremental '*trainc*' (Table 2), BFGS Quasi-Newton '*trainbfg*' (Table 3), and Levenberg-Marquardt '*trainlm*' (Table 4) were 0.848 – 0.950 and 57.25 – 83.35%, 0.967 – 0.996 and 4.93 – 18.70%, and 0.996 – 0.999 and 0.68 – 5.77%, respectively. The network structure (2-20-31-1) trained with the Levenberg-Marquardt '*trainlm*' algorithm with

Table 4: Performance parameters for different network structures trained with Levenberg-Marquardt '*trainlm*' algorithm

Network Structure	Network performance parameters							
	Training dataset				Test dataset			
	Correlation Coefficient (R-value)	MSE	MAPE (%)	SSE	Correlation Coefficient (R-value)	MSE	MAPE (%)	SSE
2-5-1	0.993	6.064e-6	6.39	0.012 e-0	0.996	3.875e-6	5.77	0.003 e-0
2-10-1	0.995	4.476e-6	5.41	0.008 e-0	0.996	3.288e-6	5.39	0.002 e-0
2-15-1	0.992	3.176e-6	5.10	0.003 e-0	0.995	2.208e-6	5.24	0.001 e-0
2-5-5-1	0.997	4.137e-7	1.94	7.943e-4	0.999	3.720e-7	2.02	2.382e-4
2-5-10-1	0.995	4.112e-6	4.12	0.008 e-0	0.997	2.719e-6	3.77	0.002 e-0
2-5-15-1	0.999	6.959e-7	2.38	0.001 e-0	0.999	4.538e-7	2.06	2.904e-4
2-10-15-1	0.999	5.599e-7	2.16	0.001 e-0	0.999	3.379e-7	1.76	2.162e-4
2-15-20-1	1.000	1.384e-7	0.98	2.657e-4	0.999	1.649e-7	0.94	1.055e-4
2-20-30-1	0.999	1.010e-7	0.82	1.940e-4	0.999	1.645e-7	0.80	1.053e-4
2-20-31-1	0.999	4.275e-8	0.51	8.208e-5	0.999	5.335e-7	0.68	3.414e-4

highest R-value (0.999) and lowest MAPE (0.68%) gave the best predictive performance compared with other network structures and training algorithms investigated. The comparison between the actual and the predicted values for the best network structure and training algorithm is shown in Fig. 5.

3.2 Graphical User Interface (GUI): A graphical user interface (GUI), was designed based on the best network structure and training algorithm, to enhance the users' friendliness application of the model. The GUI (Fig. 6) was designed using the GUI toolbox for MATLAB$^{\circ}$. On input of the pipe diameter, pipe roughness and Reynolds number of the flow in the respective data input windows, the relative roughness is computed internally as the ratio of the pipe roughness and pipe diameter and the friction factor is predicted for the given relative roughness and Reynolds number by clicking the 'calculate friction factor' button. The calculated relative roughness and friction factor are displayed in the respective data output windows. Graphical display of the predicted friction factor for given range of Reynolds number was also incorporated in the GUI. The minimum and maximum values of the Reynolds number are inputted in the respective data input windows and chart is displayed in the graphing window by clicking the 'plot' button. The GUI also allows easy comparison of friction factor charts for two or more plots by plotting them together using the 'hold' button. The facilities for zooming in and out on the chart were incorporated by the 'zoom in' and 'zoom out' buttons. It is however important to note that the GUI will work with appropriate accuracy only when the input parameters are within the range of the dataset used in this study. A pop-up window showing a warning message is displaced as feedback when inputted data is out of range.

An illustrative example (Figure 6) for typical pipe and flow parameters (D = 0.56, ε = 0.009, Re = 45600) shows the relative roughness and predicted friction factor as 0.01607 and 0.04578, respectively, while the graphical display shows plots for relative roughness of 0.0161 and 0.0107 for the Reynolds number ranging from 3.4×10^3 and 1.04×10^7.

Conclusions: In this paper, a suitable method for predicting friction factor in pipe flow using an artificial neural network is described. The friction factor prediction is done in a simple way with no need for neither analytical nor empirical equation. This model can predict friction factor using relative roughness and Reynolds number as input parameters. The validation of the model was performed with previous data, which has not been used in the training of the network. The network structure with 2-20-31-1 configuration trained with the Levenberg-Marquardt '*trainlm*' algorithm gave the best prediction performance with highest R-value (0.999) and lowest MAPE (0.68%). This accuracy is within the acceptable level used by design engineers.

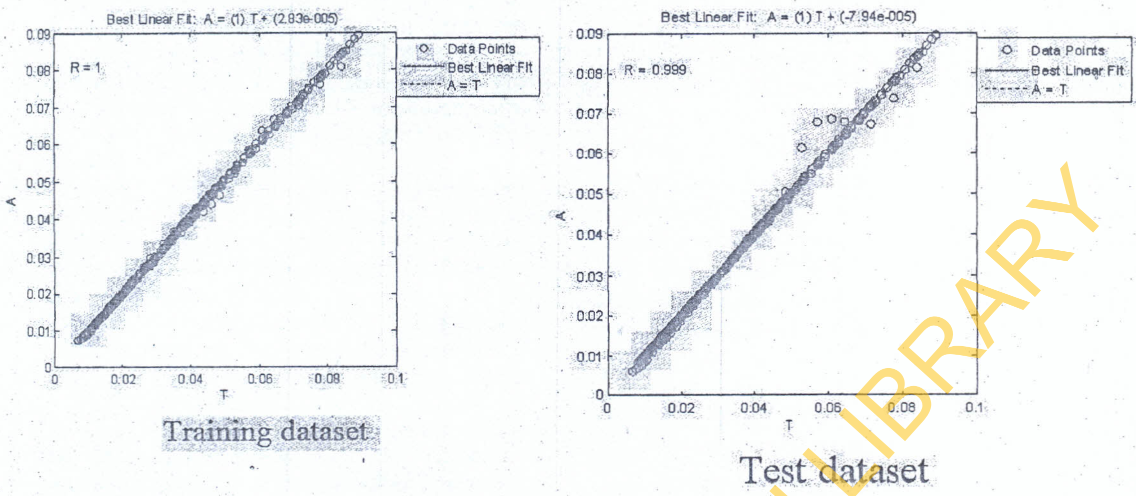

Fig. 5: Comparison between the actual and ANN predicted friction factor using 2-20-31-1 network trained with Levenberg-Marquardt '*trainlm*' (T=actual values; A= ANN predicted values)

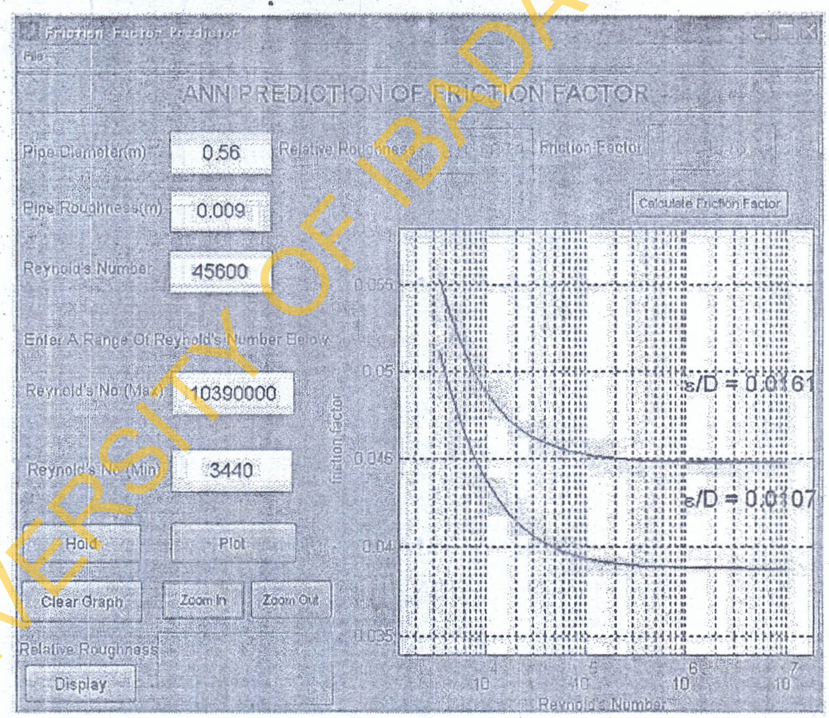

Fig. 6: Graphical user interface (GUI) for prediction of friction factor in pipe flow

REFERENCES

1. Çengel, Y.A. and R.H. Turner, 2005. Fundamentals of Thermal-Fluid Sciences, 2nd Ed., McGraw Hill, Inc., 1221 Avenue of the Americas, New NewYork, NY 10020, pp: 625.
2. Rouse, H. and S. Ince, 1957. History of Hydraulics, Iowa Institute of Hydraulic Research, Univ. of Iowa, Iowa City, pp: 269.
3. Darcy, H., 1857. Recherches Expérimentales Relatives au Mouvement de l'eau dans les Uyaux, Mallet-Bachelier, Paris, pp: 268 (In French).

4. Haaland, S.E., 1983. Simple and Explicit Formulas for the friction factor in Turbulent Pipe Flow. Journal of Fluids Engineering, 30: 89-90.
5. Colebrook, C.F., 1939. Turbulent Flow in Pipes with Particular Reference to Thetransition Region Between the Smooth and Rough pipe Laws." Proc. Institution Civil Engrs., 12: 393-422.
6. Colebrook, C.F. and C.M. White, 1937. Experiments with Fluid- Friction in Roughened Pipes. Proc. Royal Soc. London, 161: 367-381.
7. Moody, L.F., 1944. Friction Factors for Pipe Flow. Trans. ASME, 66: 671-678.
8. Mellit, A., M. Benghanen and S.A. Kalogirou, 2006. An Adaptive Wavelet Network Model for Forecasting Daily Total Solar Radiation. Applied Energy, 83: 705-722.
9. Hernández, S., S. Nesic, G. Weckman and V. Ghai, 2005. Use of Artificial Neural Networks for Predicting Crude Oil Effect on CO_2 Corrosion of Carbon Steels". Corrosion 2005. Paper No 05554.NACE.
10. Fadare, D.A., 2009. Modelling of Solar Energy Potential in Nigeria Using an Artificial Neural Network Model. Applied Energy (in press) doi :10.1016/j.apenergy. 2008.12.005.
11. Krasnopolsky, V.M., D.V. Chalikov, H.L. Tolman, 2002. A Neural Network Technique to Improve Computational Efficiency of Numerical Oceanic Models. Ocean Modelling, 4: 363-383.
12. Haykin, S., 1999. "Neural Networks- A Comprehensive Foundation, 2nd Ed.", Prentice Hall, Inc., New Jersey.
13. Bishop, C.M., 1995. Neural Networks for Pattern Recognition, Oxford University Press Inc., Oxford.
14. Depold, H.R. and F.D. Gass, 1999. The Application of Expert Systems and Neural Networks to Gas Turbine Prognostics and Diagnostics. Journal of Engineering for Gas Tiurbines and Power, ASME, 121: 607-612.
15. Grauppe, D., 1995. Principles of Artificial Neural Networks, 2nd Ed.Advanced Series on Circuits and Systems – Vol. 6, World Scientific Publishing Co. Pte. Ltd.
16. Negm, A.M., A.A. Ibrahim, A.A. El-Saiad1 and A.M. Al-Brahim, 2004. Flow resistance due to cylindrical piles. Egyptian Journal of Engineering Science and Technology (EJEST), 7: 123-234.
17. Azimian, A.R., 2005. Application of Artificial Neural Networks in Pipe Flow Calculations. Proceedings of 4th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics Cairo, HEFAT 2005, Egypt, (On CD) Paper AA1.
18. Demuth, H. and M. Baale, 2000. Neural Network Toolbox Manual. Mathworks Inc. USA., pp: 850.