cogent engineering

COMPUTER SCIENCE | RESEARCH ARTICLE

Modeling friction factor in pipeline flow using a GMDH-type neural network

Saeb M. Besarati[1], Philip D. Myers[1*], David C. Covey[2] and Ali Jamali[3]

Received: 04 March 2015
Accepted: 16 May 2015
Published: 18 June 2015

*Corresponding author: Philip D. Myers, Department of Chemical and Biomedical Engineering, University of South Florida, 4202 E Fowler Ave, ENB118, Tampa, FL, USA
E-mail: PhilipMyers@mail.usf.edu

Reviewing editor:
Jenhui Chen, Chang Gung University, Taiwan

Additional information is available at the end of the article

Abstract: The standard methods of calculating the fluid friction factor, the Colebrook–White and Haaland equations, require iterative solution of an implicit, transcendental function which entails high computational costs for large-scale piping networks while introducing as much as 15% error. This study applies the group method of data handling to the development of an artificial neural network optimized by multi-objective genetic algorithms to find an explicit polynomial model for friction factor. We developed a relatively simple and explicit model for friction factor that performs well over the entire range of applicability of the Colebrook–White equation: Reynolds number from 4,000 to 10^8 with relative roughness ranging from 5×10^{-6} to 0.05. For a network with only two hidden layers and a total of five neurons, this model was found to have a mean relative error of only 3.4% in comparison with the Colebrook–White equation; a determination coefficient (R^2) over the range of input data was calculated to be 0.9954. The accuracy and simplicity of this model may make it preferable to traditional, transcendental representations of fluid friction factor. Further, this method of model development can be applied to any pertinent data-set—that is to say, the model can be tuned to the physical situation and input data range of interest.

Subjects: Fluid Mechanics; Mathematical Modeling; Neural Networks

Keywords: friction factor; Colebrook–White equation; artificial neural networks; genetic algorithms

ABOUT THE AUTHOR

Saeb M. Besarati earned his bachelor and master's degrees in mechanical engineering from University of Guilan, Iran. He earned his PhD in chemical engineering from the University of South Florida (USF) in 2014. The focus of Dr. Besarati's work has been the use of advanced numerical techniques to solve complex engineering problems in the field of clean energy research. He conducted his doctoral studies at the USF Clean Energy Research Center under the guidance of Dr. Yogi Goswami. His doctoral research concerned the analysis of supercritical carbon dioxide power cycles for concentrated solar power applications. During his PhD study, he contributed in writing two book chapters and ten technical papers. He has been a lecturer for courses in the Department of Mechanical Engineering at the Azad University of Takestan and the Department of Chemical Engineering at USF. He is currently an engineering analyst with the firm eSolar.

PUBLIC INTEREST STATEMENT

In this paper, we analyze a classical engineering problem—that of friction in pipeline flow—using the comparatively modern techniques of artificial neural networks and genetic algorithms. In the process, we obtain a set of simple (i.e. non-transcendental) equations that give excellent agreement with the Colebrook–White equation. The method can also be tailored to experimental data for more specific applications.

1. Introduction

The flow of real fluids (e.g. through pipes, ducts, etc.) involves the loss of mechanical energy through the friction that accompanies flow-induced shear stresses. The magnitude of this loss depends on the viscosity of the fluid(s) in question as well as the size of the fluid flow system. Often, this loss can be quite severe, and calculations to determine the degree of viscous friction become a necessary part of the design, implementation, and control of engineered applications of fluid flow. In process control especially, such calculations must be repeated continually, and for systems of substantial size (e.g. large-scale piping networks) they constitute a high computational cost.

1.1. Colebrook–White equation

A common and comparatively simple context for these fluid flow analyses is fully-developed Newtonian flow in pipes or ducts. Through dimensional analysis, it is possible to express the viscous losses as a function of a dimensionless friction factor—as a matter of convention, fluid mechanics texts typically choose the Darcy friction factor, *f*, which is a function of Reynolds number, *Re*, and the relative roughness of the pipe, ϵ/D. In such a way, the loss of mechanical energy, or head loss, h_f, as it is termed when expressed as an equivalent height of fluid, can be calculated with the Darcy–Weisbach equation (White, 1998).

$$h_f = f\frac{L}{D}\frac{V^2}{2g} \tag{1}$$

Here, *L* represents the length of the pipe or duct, *D* is the hydraulic diameter, *V* is the average flow velocity, and *g* is the acceleration due to gravity. It should be noted that this equation is valid for any flow regime or pipe/duct geometry, provided the hydraulic diameter and the Darcy friction factor are appropriately derived.

The straightforward nature of the calculation of head loss for a given friction factor (i.e. Equation 1) belies the complexity of the calculation of the friction factor itself. While, in the laminar case ($Re < 2{,}000$), the friction factor becomes a rather simple function of Reynolds number (White, 1998), turbulent flow requires more complicated, typically transcendental functions to capture the variation of the friction factor with respect Reynolds number and relative roughness. The most broadly applicable function for determining the friction factor in turbulent flow regimes is the Colebrook–White equation, developed by the eponymous researcher, Colebrook (1939).

$$\frac{1}{f^{1/2}} = -2.0\log\left(\frac{\frac{\epsilon}{D}}{3.7} + \frac{2.51}{Re\, f^{\frac{1}{2}}}\right) \tag{2}$$

This expression is known to apply over a range in Reynolds number of approximately 4,000–10^8 (White, 1998).

It is clear that, owing to the implicit nature of the friction factor in Equation 2, iterative methods are required to calculate the friction factor for a given Reynolds number and relative roughness. Although approximations of the Colebrook–White equation that are explicit in Darcy friction factor *f* have been formulated, they tend to apply to specific flow scenarios and/or pipe geometries, and they introduce additional error. The Haaland equation (Equation 3), for example, assumes the same basic form of the Colebrook–White, but is explicit in terms of the friction factor; it is among the more accurate approximations, varying less than 2% from the behavior of Equation 2 (Haaland, 1983; White, 1998).

$$\frac{1}{f^{1/2}} = -1.8\log\left[\left(\frac{\frac{\epsilon}{D}}{3.7}\right)^{1.11} + \frac{6.9}{Re}\right] \tag{3}$$

It should be noted that the Colebrook–White equation itself is known to introduce a degree of error, perhaps as high as 15%, over its range of applicability (White, 1998). Of course, this error applies with respect to the real (i.e. experimentally quantified) behavior of fluids. The error in the Haaland equation, on the other hand, applies with respect to the performance of the Colebrook–White equation, thereby compounding the overall error in design calculation, potentially to a degree greater than 15% of the real value.

1.2. Calculation of friction factor via neural networks

For calculations concerning smaller data-sets, the iteratively solved Colebrook–White equation will generally suffice. However, a review of recent literature demonstrates the need for more efficient methods of computation, especially for larger, more complex piping networks, such as large-scale micro-irrigation systems (Shayya & Sablani, 1998), combined sewer systems (Haaland, 1983), etc. In particular, much work has been done to employ artificial neural networks (ANNs) and evolutionary algorithms (EAs) to calculate pipeflow friction factors. Indeed, such numerical techniques proved to be a powerful tool in solving non-linear, transcendental, or otherwise complex mathematical relations such as the Colebrook–White equation, as they allow for the modeling of perhaps mathematically intractable phenomena with little or no *a priori* knowledge of the mathematical characteristics of the physical processes that inhere therein.

An ANN is typically structured in three distinct parts: the input layer, an output layer, and any number of hidden layers between the input and output. Each of these layers is made up of neurons, and each neuron is interconnected through adjacent layers. An ANN is developed via "training" with a data-set, allowing the network to adjust the relative weights of the connections between each neuron such that an output of acceptable error is produced (Fausett, 1993). In the case of the Colebrook–White equation considered in this study, the network is made up of two input neurons, corresponding to each input (Reynolds number and relative roughness), an undetermined number of neurons in the hidden layers to operate on the input data, and one output neuron (the desired calculable quantity, the friction factor). The input or output data used to train the network may be transformed in any number of ways to improve the performance of the resultant model—for example, the base-ten logarithm of the Reynolds number may be chosen as an input neuron in lieu of the Reynolds number itself, as described in the methodology section. Of course, additional input neurons may be introduced by considering the individual parameters that constitute the Reynolds number (i.e. fluid velocity, effective diameter, and kinematic viscosity) (Mittal & Zhang, 2007), though such a complication was deemed unnecessary for the purposes of this study.

1.3. Literature review

A significant amount of effort has been expended on the application of ANNs to the problem of fluid flow friction factor determination in both Newtonian (Bilgil & Altun, 2008; Fadare & Ofidhe, 2009; Mittal & Zhang, 2007; Özger & Yildirim, 2009; Shayya & Sablani, 1998; Yazdi & Bardi, 2011; Yuhong & Wenxin, 2009) and non-Newtonian (Mittal & Zhang, 2007; Sablani & Shayya, 2003; Sablani, Shayya, & Kacimov, 2003; Shayya, Sablani, & Campo, 2005) flows. For reference, a summary of these studies in included here as Table 1. Inasmuch as the Colebrook–White equation assumes Newtonian flow, this study will focus primarily on the work dealing with such flow behavior. While in some cases, the data range for Reynolds number employed in ANN development was outside the acceptable range for the Colebrook White equation (Fadare & Ofidhe, 2009; Shayya & Sablani, 1998), very close agreement was obtained for sufficiently large networks and appropriate transformation of input and/or output parameters. In one particular study, an attempt was made to fit the friction factor behavior in both the laminar and turbulent flow regimes, though the inclusion of points in the highly indeterminate transition region introduces some uncertainty as to the reliability of the model and its ostensibly low error (Mittal & Zhang, 2007). Another study performed a model fit of the Haaland equation (Equation 3) with very high accuracy, although, here again, points were included in the training of the network that fall outside the realm of acceptability for such empirical relations (Yazdi & Bardi, 2011). The work performed in this field demonstrated that rather complex ANNs achieve a high degree of accuracy in modeling experimental data (Yuhong & Wenxin, 2009), the Colebrook–White equation (Fadare & Ofidhe, 2009; Mittal & Zhang, 2007; Shayya & Sablani, 1998), and other empiricisms of Newtonian flow (Bilgil & Altun, 2008; Mittal & Zhang, 2007; Yazdi & Bardi, 2011; Yuhong & Wenxin, 2009). No explicit relations for the friction factor were put forth in these works.

Table 1. Literature review summary

References	Description	Method	Data-set	Source for data-set	Range—ϵ/*D*	Range—*Re*
Shayya et al. (2005)	Authors developed ANN to find Fanning friction factor for Herschel–Bulkley fluids in closed pipes. Input parameters were ***Re*** and ***He***.	Feed-forward ANN	1,991 (training), 29,507 (validation)	Regula-Falsi iterative method to solve an implicit functions based on mechanical energy balance, laminar flow function, pressure gradient and friction factor. Solutions apply for laminar, transition and turbulent flows.	n/a	2–1E8
Sablani et al. (2003)	Authors develop an ANN to find Fanning friction factor for flow of Bingham plastic fluids in closed pipes. Input parameters were ***Re*** and ***He***.	Feed-forward ANN	1,177 (training), 32,141 (validation)	Regula-Falsi iterative method was used to solve for *f* given ***Re*** and ***He*** in laminar and turbulent conditions.	n/a	2–1E8
Sablani and Shayya (2003)	Authors developed an ANN to find the Fanning friction factor for power law (shear thinning) fluids in closed pipes.	Feed-forward ANN	7,420 (training), 72,400 (validation)	Regula-Falsi iterative method was used to solve an equation the authors developed based on the von Karman equation, which is a function of ***Re*** and ***n***, where ***n*** is the flow behavior index.	n/a	2E3–1E8
Shayya and Sablani (1998)	Authors developed an ANN to find the Darcy friction factor for flow of Newtonian fluids in closed pipes.	Feed-forward ANN	1,720 (training), 82,100 (validation)	Regula-Falsi iterative method was used to solve the Colebrook–White equation for specified ranges of relative roughness and Reynolds number.	1E-6 to 5E-2	2E3–1E8
Tufail and Ormsbee (2006)	Authors develop a FFSGA based on genetic algorithms and least squares optimization to find Darcy friction factor. The method combines numerical coefficients, functions of decision variables, and mathematical operators.	Fixed functional set genetic algorithm (FFSGA)	100	Authors used the Swamee-Jain equation (explicit version of Colebrook–White equation, limited to full flow circular pipe) to generate data-sets, with specified ranges of relative roughness and ***Re***.	0.001–0.01	1E5–1E6
Mittal and Zhang (2007)	Authors created 4 ANNs to find friction factor for Newtonian, Bingham plastic, Power law, and Herschel–Bulkely fluids. Inputs were mean velocity, viscosity, density, and pipe diameter.	Back-propogation ANN	45,747 (training), 6,862 (validation)	Authors used an iterative solution to yield friction factor from the von Karman correlation.	n/a	n/a
Bilgil and Altun (2008)	Authors develop an ANN to predict friction factor in open channel flow for a Newtonian fluid in comparison with prediction of the Manning equation, given experimental data.	Feed-forward ANN	n/a	Authors created an experimental setup with an adjustable-slope open channel and varied the W/h ratio of the channel and water discharge. They compared their experimental results to their ANN as well as the Manning equation.	n/a	8,291–217,791

(*Continued*)

Table 1. (*Continued*)

References	Description	Method	Data-set	Source for data-set	Range—ϵ/D	Range—*Re*
Özger and Yildirim (2009)	Authors used a set of 9 “fuzzy” rules (IF THEN statements) relating ***Re***, relative roughness and ***f***, whereby the rules are structured similar to an ANN to predict ***f***.	Adaptive neuro-fuzzy inference system (ANFIS)	1,042 (training), 700 (validation)	Authors used the graph reading software ***Techdig 2.0*** to sensitively read data from the Moody diagram.	1E-6–5E-2	4E3–1E8
Yazdi and Bardi (2011)	Authors developed an ANN to predict Darcy friction factor, given ***Re*** and relative roughness as inputs. They also compared predictions of *f* depending on the training algorithm used.	Feed-forward ANN	1,600 (training), 400 (validation)	Authors generated a data-set from the Haaland equation, which is explicit in ***f***, given specified ranges of relative roughness and ***Re***.	n/a	3E3–1E8
Fadare and Ofidhe (2009)	Authors developed an ANN to predict Darcy friction factor, given ***Re*** and relative roughness as inputs. They also compared predictions of *f* depending on the training algorithm used.	Feed-forward ANN	1,920 (training), 640 (validation)	Authors generated a data-set from the Haaland equation, which is explicit in ***f***, given specified ranges of relative roughness and ***Re***.	5E-6–7E-2	2.5E3–1E8

Acronyms: MRE = Mean relative error; MSE = Mean square error; MAPE = Mean absolute percentage error; SSE = Sum of square error; *R* = Correlation coefficient; R^2 = Coefficient of determination; STD = Standard deviation.

MAE, STD (of MAE), MRE, STD (of MRE), R^2.

MAE, STD (of MAE), MRE, STD (of MRE), R^2.

MAE, STD (of MAE), MRE, STD (of MRE), R^2.

MAE, STD (of MAE), MRE, STD (of MRE), R^2.

MSE, Max absolute error.

Avg. Absolute error, *R*.

R^2.

MSE, *R*, Max % Error, Time, Iterations.

MSE, MAPE (%), SSE, *R*.

There is a tendency in ANN-based modeling to develop the network model as a “black-box” (Tufail & Ormsbee, 2006)—a complex, abstruse system of functions that convert input data to desired results, oftentimes quite accurately, yet with little innate understanding of the mathematical machinery of the model. For the problem considered in this paper, it is clear that such a complex model defeats the purpose of approximating the Colebrook–White equation in the first place. Equation 3 is acceptable in terms of approximating the Colebrook–White equation, although its transcendental nature makes it less desirable, in terms of computational effort, than a simple polynomial (Davidson, Savic, & Walters, 1999). However, certainly a simply stated transcendental function is more tractable than a complex system of intermarried polynomial functions, such as those often output by proprietary neural network software packages. The maxim of model-parsimony may suggest the choice of the simpler and slightly less accurate over the more accurate and more unwieldy, a balancing act described in the non-Newtonian flow ANN modeling of Sablani, Shayya, et al. (Sablani & Shayya, 2003; Sablani et al., 2003; Shayya et al., 2005). For more information on the black box method employing software packages for ANN development, the reader is encouraged to peruse the pertinent references (Fadare & Ofidhe, 2009; Mittal & Zhang, 2007; Shayya & Sablani, 1998; Yazdi & Bardi, 2011).

Also of unique interest in this field is the combination of fuzzy logic with ANNs in model development. An adaptive neuro-fuzzy inference modeling system involves the use of stochastic membership functions and conditional statements with associated model functions to fit real values of interest from pseudo-qualitative assessment of input data (e.g. low, medium, and high). Such a system was shown to have near perfect agreement with the Colebrook–White equation over the Reynolds number range of interest, 4,000–10^8 (Özger & Yildirim, 2009). Unfortunately, the multitude of fuzzy rules and attendant modeling functions render this method a black box, as well.

EAs present a potential improvement when wedded with conventional ANN techniques (Onwubolu, 2009). The idea behind the use of such genetic algorithms in this context is that models or functional interactions that produce the best fit to the training data have the best chance of reproducing and therefore shaping the final model system. Already, genetic algorithms, without the use of neural networks, have been employed to fit the Colebrook–White equation. In one study, standalone polynomial functions of varying complexity were developed with such a technique, and most were shown to yield a good fit (Davidson et al., 1999). This work was extended in a subsequent study that utilized a fixed function set genetic algorithm that considered a wider variety of functional forms in its model development (Tufail & Ormsbee, 2006). Explicit functions for the Colebrook–White equation were presented in both cases; however, the models developed applied only over a relatively narrow range of Reynolds number and relative roughness ($10^5 \le Re \le 10^6$; $0.001 \le \epsilon/D \le 0.01$). A wider range of Reynolds number and relative roughness ($4 \times 10^4 \le Re \le 10^8$; $10^{-6} \le \epsilon/D \le 0.05$) was considered in the gene expression programming (GEP) analysis of Samadianfard (Samadianfard, 2012). The explicit expression obtained in that study compared well with the various approximations described therein; however, a separate study highlighted the potentially high errors obtained with its use and also suggested similar expressions with improved accuracy (Vatankhah, 2014). It should be noted that the GEP techniques referenced here, while generally related to genetic algorithms, make no use of ANNs.

1.4. Investigation objectives

The main goal of this study is to apply the group method of data handling (GMDH) ANN, optimized by multi-objective genetic algorithms, to find an explicit model for friction factor. This methodology has been proven effective in the modeling and prediction of complex and non-linear processes such as explosive cutting, a variable valve-timing spark-ignition engine, and others (Atashkari, Nariman-Zadeh, Gölcü, Khalkhali, & Jamali, 2007; Onwubolu, 2009). The quasi-Monte Carlo method of Hammersley (Press, Teukolsky, Vetterling, & Flannery, 2007) will be used to generate an input dataset of random yet uniformly spaced points over the entire range of applicability of the Colebrook–White equation: Reynolds number from 4,000 to 10^8 and relative roughness from 5×10^{-6} to 0.05. The Colebrook–White equation shall be solved with non-linear solution methods to obtain the friction factor values for use in training and testing the neural network. Ultimately, the development of this GMDH-type ANN shall yield an explicit polynomial model for friction factor to be presented herein.

2. Methodology

2.1. Construction of the neural network

By means of the GMDH algorithm, a model can be represented as a set of neurons in which different neuron pairs in each layer are connected through a quadratic polynomial and thus produce new neurons in subsequent layers. Such representation can be used in modeling to map inputs to outputs. The formal definition of the identification problem is to develop a function, $\hat{f}$, that approximates the actual function, f, to a sufficient accuracy to predict the actual output y for a given input vector $X = (x_1, x_2, \ldots, x_n)$ as $\hat{y}$. Therefore, given M observations of multi-input, single output data pairs, we define the system in the following manner.

$$y_i = f(x_{i1}, x_{i2}, \ldots, x_{in}), \text{ for } i = 1, 2, \ldots M \tag{4}$$

It is possible to train a GMDH-type neural network to predict the output values for any given input vector, as illustrated in Equation 5.

$$\hat{y}_i = \hat{f}(x_{i1}, x_{i2}, \ldots, x_{in}), \text{ for } i = 1, 2, \ldots M \tag{5}$$

The problem is now to determine a GMDH-type network so that the square of the differences between the actual output and the predicted one is minimized, that is:

$$\sum_{i=1}^{M} [\hat{y}_i - y_i]^2 \rightarrow \min \tag{6}$$

The general connection between the inputs and the output variables can be expressed by a discrete form of the Volterra functional series, as follows.

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j + \ldots \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk} x_i x_j x_k + \cdots$$

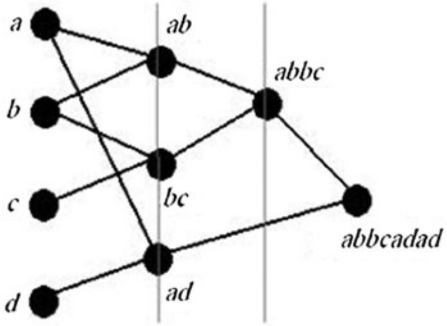
This form of the series is known as the Kolmogorov–Gabor polynomial (Farlow, 1984). This full form of mathematical description can be represented by a system of partial quadratic polynomials consisting of only two input variables (neurons), in the following form.

$$\hat{y} = G(x_i, x_j) = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \tag{7}$$

Such partial quadratic description is recursively used in a network of connected neurons to build the general mathematical relation of the inputs and output variables. The coefficients a_i in Equation 7 are calculated using regression techniques (Farlow, 1984; Iba, DeGaris, & Sato, 1996; Ivakhnenko, 1971), so that the difference between the actual output, y, and the calculated one, $\hat{y}$ for each pair of x_i, x_j as input variables is minimized. It can be seen that a tree of polynomials is constructed using the quadratic form, Equation 7, whose coefficients are regressed in a least-squares sense. In this way, the coefficients of each quadratic function G_i are obtained to fit optimally the output in the whole set of input/output data pairs.

In the basic form of the GMDH algorithm, all the possibilities of two independent variables out of the total n input variables are taken in order to construct the regression polynomial (Equation 7) that best fits the dependent observations (Jamali, Nariman-zadeh, Darvizeh, Masoumi, & Hamrang, 2009; Nariman-Zadeh, Darvizeh, & Ahmad-Zadeh, 2003; Nariman-Zadeh, Darvizeh, Jamali, & Moeini, 2005). In this paper, the general structure of GMDH-type neural network (GS-GMDH) (Jamali et al., 2009; Nariman-Zadeh et al., 2005) is used for modeling and prediction of the friction factor. In the GS-GMDH ANN, neuron connections can occur between different layers that are not necessarily immediately adjacent, unlike the conventional structure GMDH (CS-GMDH) neural networks in which such connections only occur between adjacent layers. Consequently, this generalization of the network's structure can extend the performance of GS-GMDH neural networks in modeling of real-world complex processes.

Figure 1. Example GMDH ANN.



Genetic algorithms can be used to derive the optimal connectivity configurations of such GMDH networks. The genome or chromosome representation, which depicts the topology of a GMDH-type neural network, consists of a symbolic string composed of alphabetic representation of input variables. In this coding scheme, each input variable is assigned an alphabetic name and the chromosome is a string consisting of the concatenated sub-strings of these names. In other words, for a given input vector $X = (x_1, x_2, \dots, x_n)$, a chromosome can be represented as a string of the concatenated symbols of $\alpha_i \in \{a, b, c, d, \dots\}$ in the form of $choromosome = (\alpha_1, \alpha_2, \dots, \alpha_i, \dots)$, where "*a*," "*b*," "*c*," etc., stand for alphabetical name of inputs x_1, x_2, x_3, etc., respectively.

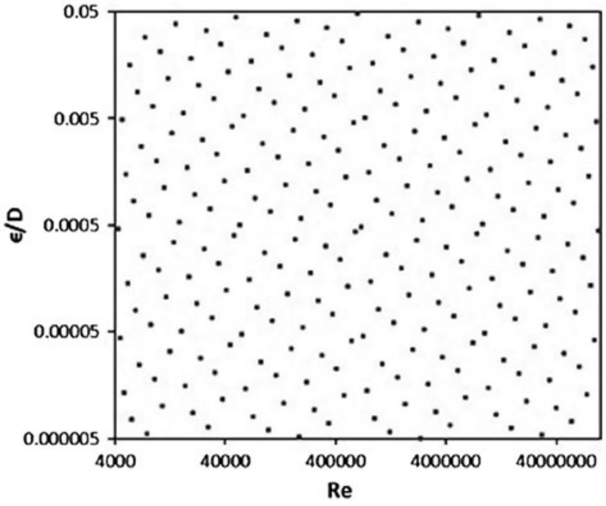
The same procedure for defining chromosomes described by Nariman-Zadeh et al. (2005) can now be readily modified to apply to GS-GMDH networks. This modification is accomplished by repeating the name of the neuron which directly passes the next layers. For example, a network structure, as depicted in Figure 1, exhibits such a connection between neuron "*ad*" and the output layer. It is clear that neuron "*ad*" in the first hidden layer is connected to the output layer by directly bypassing the second hidden layer. Therefore, the name of the output neuron includes "*ad*" twice as "*abbcadad*." Essentially, a virtual neuron named "*adad*" has been constructed in the second hidden layer and used with "*abbc*" in the same layer to make the output neuron "*abbcadad*." It should be noted that in this coding method such repetition occurs whenever a neuron passes adjacent hidden layers and connects to a neuron in subsequent layers. The number of repetitions of that neuron's signifier depends on the number of passed hidden layers, *ñ*, and is calculated as $2^{\tilde{n}}$. It is evident that a chromosome such as "*abab bcbc*," unlike chromosome "*abab acbc,*" is not valid in GS-GMDH networks, and therefore it should be rewritten as "*abbc*."

2.2. Application of the network to the Colebrook–White equation

For the purposes of finding an explicit form of the Colebrook–White equation, it is sufficient to consider two inputs—the Reynolds number and relative roughness—and one output—the friction factor. As suggested by multiple prior studies (Sablani & Shayya, 2003; Sablani et al., 2003; Shayya et al., 2005) and independently verified by this study, transforming the inputs with a base-ten logarithm greatly improves the accuracy of the resultant ANN model. As such, the two inputs used in developing this ANN model were the base-ten logarithms of Reynolds number and relative roughness—i.e. $\log_{10}Re$ and $\log_{10}(\epsilon/D)$. Also, a variety of transforms for the friction factor output is essayed, including a base-ten logarithm and several negative rational exponents between zero and one. It was determined through analysis of the resultant models' coefficients of determination (compared to the Colebrook–White equation) that the most accurate networks were obtained when the friction factor output was transformed by raising it to the power of negative one-half. As such, the output used to train and test the network was the inverse of the square root of the friction factor, $1/\sqrt{f}$. These inputs and outputs are not unexpected, as they mimic the form of the Colebrook–White and Haaland equations (Equations 2 and 3).

In addition to strategic transformation of the input and output data, a sufficiently large and appropriately spaced sampling set of input variables is necessary to ensure that the training and testing of

Figure 2. Distribution of samples over the range of Reynolds number and relative roughness.



the neural network results in an accurate model. To achieve adequate distribution over the range of input variables, Hammersley sequence sampling (Press et al., 2007) was employed to generate 250 sample points between 0 and 1 in two dimensions, and these fractional values were applied over the individual variable ranges to create the input data sampling set. The range of inputs considered, prior to logarithmic transform, was as follows: Reynolds number from 4,000 to 10^8 and relative roughness from 5×10^{-6} to 0.05. It was determined that using such a sampling method will allow for greater accuracy with fewer training/testing points. The sampling set is shown in Figure 2.

The output data for training and testing the network were obtained by non-linear solution of the Colebrook–White equation for each of the 250 data points generated by the Hammersley sampling algorithm. A variant of the Powell dogleg method was used to solve the non-linear, transcendental equations at each data point; thereafter, the friction factor data was transformed into the appropriate ANN output, as described previously. To train the network, a total of 125 points of the sampling set were used—that is, an equal number of points were used to train and then test the network. To prevent either the training or testing sets from becoming biased to any particular region of the input data range, the 250-point Hammersley sampling set was randomized prior to equal separation into training and testing sampling sets. In the next step, genetic algorithms were used to find an optimal structure of the GMDH-type neural network to yield an explicit equation for friction factor. Two hidden layers were ultimately settled upon for the general structure of the network: while three hidden layers provided some improved accuracy, the reduction in error in that case was deemed too slight to justify the additional complexity of the model.

2.3. Error analysis

In comparison with the Colebrook–White equation and other efforts made in this field of study, the following error analysis metrics were used.

Coefficient of Determination

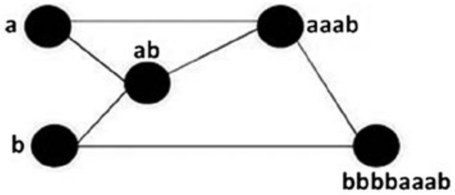
$$R^2 = 1 - \frac{\sum \left(f_p - f_d\right)^2}{\sum \left(f_d\right)^2} \tag{8}$$

Mean Relative Error

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{f_p - f_d}{f_d} \right| \tag{9}$$

Mean Absolute Error

Figure 3. Optimal structure of GS-GMDH network for modeling of friction factor.



$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|f_p - f_d\right| \tag{10}$$

Here, f_p is the predicted value from the ANN, f_d is the desired value from the iterative solution of the Colebrook–White equation, and n is the total number of data points—in this case, 250. R^2 is especially important here, as it measures the accuracy of the overall fit of the predicted data to the desired outputs. A value of unity indicates perfect correlation. For the mean-relative and mean-absolute errors (MRE, MAE), the greater the quantity, the greater the disagreement between the predicted and desired values. The MRE, incidentally, provides a more objective quantification of error than the MAE, as it normalizes the measured error to the mean value of the dataset in question.

3. Results and discussion

The input–output data-set was used to train a GMDH-type neural network to find a polynomial model of friction factor in terms of the input variables. In order to genetically refine this GMDH-type ANN, a population of 40 individuals in 120 generations with a crossover probability of 0.95 and a mutation probability of 0.01 was used; no additional improvement was observed for larger population sizes. The objective functions to be minimized simultaneously were the prediction and test error functions; the final trade-off structure of GMDH-type neural network was optimally selected by a genetic algorithm that minimized these two objective functions. The resultant ANN structure is shown in Figure 3, where "*a*" and "*b*" represent $\log_{10}(Re)$ and $\log_{10}(\epsilon/D)$, respectively.

The corresponding polynomial representation of the model is as follows.

$$\begin{aligned} y_1 = {} & -0.434566150456166 + 1.198691961744151a - 0.260419382502325b \\ & - 0.176284322797722a^2 - 0.246506318905692b^2 - 0.452913784505939ab \end{aligned}$$

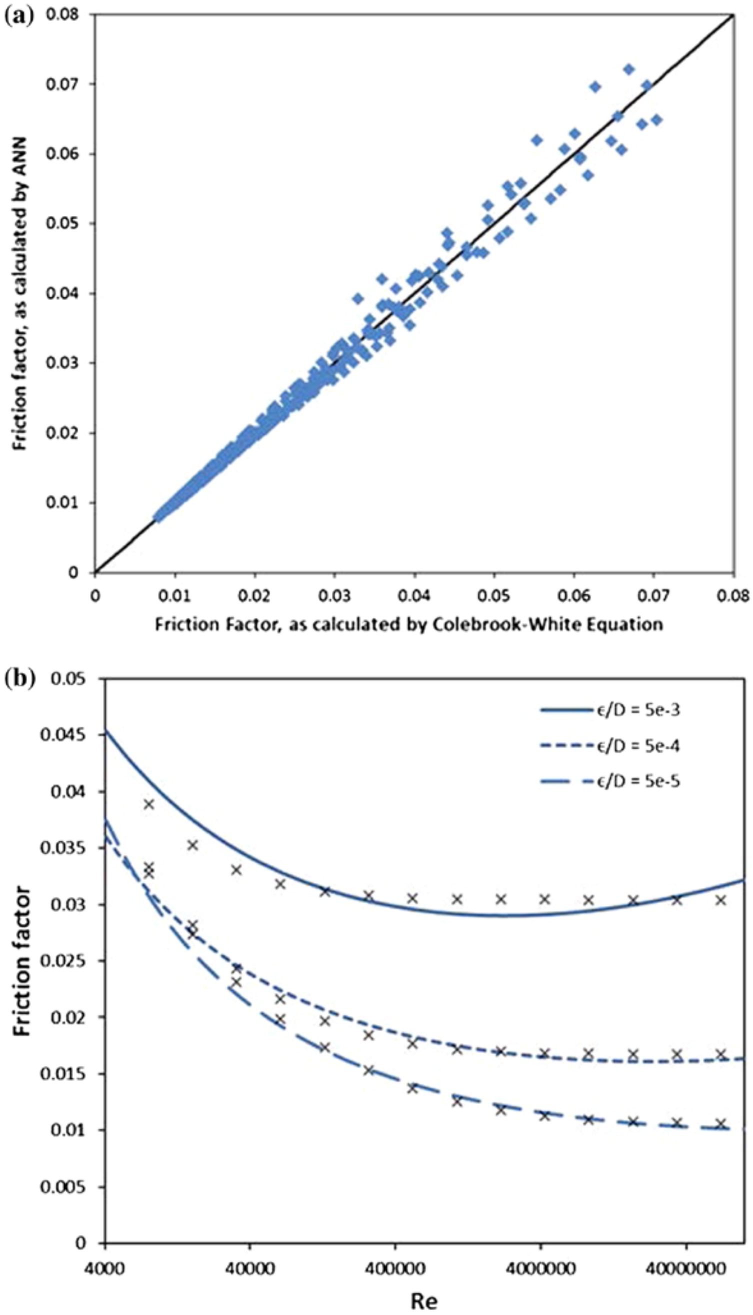
$$\begin{aligned} y_2 = {} & -0.264590878809085 + 0.104548473076918a + 0.991211881271789y_1 \\ & + 0.027044863177819a^2 + 0.036354915327913y_1^2 - 0.075133478936023ay_1 \end{aligned}$$

$$\begin{aligned} y_3 = {} & 0.049539595715222 - 0.574835644104601b + 0.741634220598117y_2 \\ & - 0.096991484123168b^2 + 0.016506184006448y_2^2 - 0.006996298079579by_2 \end{aligned}$$

$$f = \frac{1}{y_3^2}$$

Figure 4 shows the ability of the polynomial function to predict friction factor with a high level of accuracy. In Figure 4(a), we see the model function's behavior relative to that of the Colebrook–White equation over the entire 250-point data-set, with most deviation observed for higher values of the friction factor. In Figure 4(b), selected constant values for relative roughness were used to reproduce a simplified version of the classical Moody diagram: the performance of the ANN model is shown in solid and dashed lines with data points from the Colebrook–White equation overlain. Again, we see greater deviation at

Figure 4. (a) Correlation plot of the model versus the Colebrook–White equation for the entire dataset (b) A simplified reproduction of the classical Moody diagram using the ANN model (solid and dashed lines) with Colebrook–White data points (X) overlain.



higher friction factor values. Also, it seems the model is more accurate (compared to the Colebrook–White equation) for lower relative roughness values and higher Reynolds number values.

In the review of previous work on this topic, the authors noted several statistical measurements that were common to many of the publications. Three of these statistical tools were selected to compare the ANN model developed herein to previously developed ANNs. Table 2 shows the error measurements and complexity of the various neural networks. It is clear from this table that the model developed in this study has the simplest structure among those considered, yet it maintains a high degree of accuracy.

Table 2. Comparison of error and network complexity between this paper and other investigations

References	Structure	No. of neurons	R^2	MRE	MAE
This study	2-1-1-1	5	0.9954	0.0344	0.0011
Shayya et al. (2005)	2-12-12-12-1	39	0.9899	0.0173	0.00013
Sablani et al. (2003)	2-8-8-1	19	0.9870	0.0201	0.000079
Sablani et al. (2003)	2-4-1	7	0.972	0.0316	0.00105
Sablani and Shayya (2003)	2-12-12-1	27	0.9999	0.0019	2.0E-7
Shayya and Sablani (1998)	2-14-14-14-1	45	0.9996	0.0122	3.0E-6
Bilgil and Altun (2008)	6-7-15-1	29	0.9926	–	–
Özger and Yildirim (2009)	2-6-9-9-1-1	28	–	0.0068	–

Only one other study produced a polynomial equation as an approximation for friction factor. Using a related method of genetic programming, Davidson et al. produced a 14-term polynomial in order to explicitly determine f (Davidson et al., 1999). Further, they found that this polynomial performed 33% better than the explicit Haaland equation; in terms of computational cost, it was simpler for computers as such a model requires no transcendental functions, unlike Haaland or any other explicit approximation of the Colebrook–White equation (Davidson et al., 1999). However, Davidson's polynomial applies to a much narrower range of both Reynolds number and relative roughness, 10^5–10^6 and 0.001–0.01, respectively.

This study considered the entire range of applicability of the Colebrook–White equation: 4,000–10^8 for Reynolds number; 5×10^{-6}–0.05 for relative roughness. The GS-GMDH offers a compromise on both fronts, whereby a polynomial can be given that applies to a realistic, applicable range for input variables, while still avoiding transcendental, large ANNs, and complex computations. Further, as is evident from Table 2, a good fit is obtained with acceptable error for a very simple network and system of model equations.

4. Conclusions

A GMDH-type neural network model was designed to predict friction factor from Reynolds number and relative roughness. The minimum error was achieved by using base-ten logarithms of the inputs, Reynolds number and relative roughness, and the inverse square root of the output, friction factor, as the input and output, respectively, used to train and test the neural network. Using Hammersley sequence sampling method decreased the quantity of input data to only 250 points, from which 125 were used for training and 125 were used for testing the neural network. A genetic algorithm was used to refine the model based on the minimization of two objective functions, the training and testing error. The regressed polynomial representation of friction factor showed a low percentage of error despite its simplicity. The value of the numerical methods used in the development of this model lay in their ability to derive a relatively simple system of equations with relatively low computational cost—lower volume of training/testing data, smaller population sizes and fewer generations in the EA, etc.

The simplicity of the ANN model developed here is one of its primary strengths. Transcendental functions, such as those seen in the Colebrook–White equation and many of its approximations, carry a high computational cost. For applications such as large-scale piping networks, which involve high volume of computation by their very nature, or for continuous computation applications, such as process system controls, the use of a model such as that presented here would improve the speed of the computations involved in calculation of the friction factor (e.g. for real-time determination or estimation of pressure drop in pipe flow).

Another strength of this method is its inherent tunability. Note that while this model showed a 3.4% relative error to the data with which it was trained and tested, the Colebrook–White equation itself can vary by as much as 15% from real fluid flow behavior. This model, unlike the Colebrook–White equation, can be tuned to represent a real system through use of experimental data. It is believed that the accuracy and simplicity of this method, as well as its simplicity in development, may make it preferable to traditional, transcendental representations of fluid friction factor.

Funding
The authors received no direct funding for this research.

Author details
Saeb M. Besarati[1]
E-mail: sbesarati@mail.usf.edu
Philip D. Myers[1]
E-mail: PhilipMyers@mail.usf.edu
ORCID ID: http://orcid.org/0000-0002-4591-3319
David C. Covey[2]
E-mail: coveydc@gmail.com
Ali Jamali[3]
E-mail: Ali.jamali@guilan.ac.ir
[1] Department of Chemical and Biomedical Engineering, University of South Florida, 4202 E Fowler Ave, ENB118, Tampa, FL, USA.
[2] Department of Mechanical Engineering, University of South Florida, 4202 E Fowler Ave, ENB118, Tampa, FL, USA.
[3] Department of Mechanical Engineering, Engineering Faculty, The University of Guilan, P.O. Box 3756, Rasht, Iran.

Citation information
Cite this article as: Modeling friction factor in pipeline flow using a GMDH-type neural network, Saeb M. Besarati, Philip D. Myers, David C. Covey & Ali Jamali, *Cogent Engineering* (2015), 2: 1056929.

References

Atashkari, K., Nariman-Zadeh, N., Gölcü, M., Khalkhali, A., & Jamali, A. (2007). Modelling and multi-objective optimization of a variable valve-timing spark-ignition engine using polynomial neural networks and evolutionary algorithms. *Energy Conversion and Management, 48*, 1029–1041. http://dx.doi.org/10.1016/j.enconman.2006.07.007

Bilgil, A., & Altun, H. (2008). Investigation of flow resistance in smooth open channels using artificial neural networks. *Flow Measurement and Instrumentation, 19*, 404–408. http://dx.doi.org/10.1016/j.flowmeasinst.2008.07.001

Colebrook, C. F. (1939). Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws. *Journal of the ICE, 11*, 133–156. http://dx.doi.org/10.1680/ijoti.1939.13150

Davidson, J. W., Savic, D., & Walters, G. A. (1999). Method for the identification of explicit polynomial formulae for the friction in turbulent pipe flow. *Journal of Hydroinformatics, 1*, 115–126.

Fadare, D. A., & Ofidhe, U. I. (2009). Artificial neural network model for prediction of friction factor in pipe flow. *Journal of Applied Sciences, 5*, 662–670.

Farlow, S. J. (1984). *Self-organizing method in modeling: GMDH type algorithm*. New York, NY: Marcel Dekker.

Fausett, L. (1993). *Fundamentals of neural networks*. Upper Saddle River, NJ: Prentice Hall.

Haaland, S. E. (1983). Simple and explicit formulas for the friction factor in turbulent pipe flow. *Journal of Fluids Engineering, 105*, 89–90. http://dx.doi.org/10.1115/1.3240948

Iba, H., DeGaris, H., & Sato, T. (1996). A numerical approach to genetic programming for system identification. *Evolutionary Computation, 3*, 417–452.

Ivakhnenko, A. G. (1971). Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics, 1*, 364–378. http://dx.doi.org/10.1109/TSMC.1971.4308320

Jamali, A., Nariman-zadeh, N., Darvizeh, A., Masoumi, A., & Hamrang, S. (2009). Multi-objective evolutionary optimization of polynomial neural networks for modelling and prediction of explosive cutting process. *Engineering Applications of Artificial Intelligence, 22*, 676–687. http://dx.doi.org/10.1016/j.engappai.2008.11.005

Mittal, G. S., & Zhang, J. (2007). Friction factor prediction for Newtonian and non-Newtonian fluids in pipe flows using neural networks. *International Journal of Food Engineering, 3*(1), 1–18.

Nariman-Zadeh, N., Darvizeh, A., & Ahmad-Zadeh, G. R. (2003). Hybrid genetic design of GMDH-type neural networks using singular value decomposition for modelling and prediction of the explosive cutting process. *Proceedings of the Institution of Mechanical Engineers Part B: Journal of Engineering Manufacture, 217*, 779–790. http://dx.doi.org/10.1243/09544050360673161

Nariman-Zadeh, N., Darvizeh, A., Jamali, A., & Moeini, A. (2005). Evolutionary design of generalized polynomial neural networks for modelling and prediction of explosive forming process. *Journal of Materials Processing Technology, 164–165*, 1561–1571. http://dx.doi.org/10.1016/j.jmatprotec.2005.02.020

Onwubolu, G. C. (2009). *Hybrid self-organizing modeling systems*. Berlin-Heidelberg: Springer-Verlag. http://dx.doi.org/10.1007/978-3-642-01530-4

Özger, M., & Yildirim, G. (2009). Determining turbulent flow friction coefficient using adaptive neuro-fuzzy computing techniques. *Advances in Engineering Software, 40*, 281–287.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3rd ed.). Cambridge: Cambridge University Press.

Sablani, S. S., & Shayya, W. H. (2003). Neural network based non-iterative calculation of the friction factor for power law fluids. *Journal of Food Engineering, 57*, 327–335. http://dx.doi.org/10.1016/S0260-8774(02)00347-3

Sablani, S. S., Shayya, W. H., & Kacimov, A. (2003). Explicit calculation of the friction factor in pipeline flow of Bingham plastic fluids: A neural network approach. *Chemical Engineering Science, 58*, 99–106. http://dx.doi.org/10.1016/S0009-2509(02)00440-2

Samadianfard, S. (2012). Gene expression programming analysis of implicit Colebrook–White equation in turbulent flow friction factor calculation. *Journal of Petroleum Science and Engineering, 92–93*, 48–55. http://dx.doi.org/10.1016/j.petrol.2012.06.005

Shayya, W. H., & Sablani, S. S. (1998). An artificial neural network for non-iterative calculation of the friction factor in pipeline flow. *Computers and Electronics in Agriculture, 21*, 219–228. http://dx.doi.org/10.1016/S0168-1699(98)00032-5

Shayya, W. H., Sablani, S. S., & Campo, A. (2005). Explicit calculation of the friction factor for non-Newtonian fluids using artificial neural networks. *Developments in Chemical Engineering and Mineral Processing, 13*, 5–20.

Tufail, M., & Ormsbee, L. E. (2006). A fixed function set genetic algorithm (FFSGA) approach for function approximation. *Journal of Hydroinformatics, 8*, 193–206.

Vatankhah, A. R. (2014). Comment on “Gene expression programming analysis of implicit Colebrook–White equation in turbulent flow friction factor calculation”. *Journal of Petroleum Science and Engineering, 124*, 402–405. http://dx.doi.org/10.1016/j.petrol.2013.12.001

White, F. M. (1998). *Fluid mechanics* (4th ed.). Boston, MA: McGraw-Hill.

Yazdi, M., & Bardi, A. (2011). Estimation of friction factor in pipe flow using artificial neural networks. *Canadian Journal on Automation, Control and Intelligent Systems, 2*, 52–56.

Yuhong, Z., & Wenxin, H. (2009). Application of artificial neural network to predict the friction factor of open channel flow. *Communications in Nonlinear Science and Numerical Simulation, 14*, 2373–2378. http://dx.doi.org/10.1016/j.cnsns.2008.06.020

***Cogent Engineering* (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.**