

PERFORMANCE EVALUATION OF THE EXPLICIT APPROXIMATIONS OF THE IMPLICIT COLEBROOK EQUATION

U. H. Offor¹, S. B. Alabi²

¹Lecturer, Department of Chemical and Petroleum Engineering, University of Uyo, Akwa Ibom State, Nigeria

²Senior Lecturer, Department of Chemical and Petroleum Engineering, University of Uyo, Akwa Ibom State, Nigeria

Abstract

The implicit Colebrook equation has been the standard for estimating pipe friction factor in a fully developed turbulent regime. Several artificial intelligence (AI)-based and non AI-based explicit models have been developed as viable replacement for the implicit Colebrook equation. However, it is not obvious which of the models and/or approaches is the best. In this paper, the performances of the available non AI-based explicit models were compared with those of the AI-based models. The results show that genetic algorithm has been successfully utilized in optimizing the explicit model parameters with the best improvements being from 0.12% to 0.0026% based on maximum relative error index. Although genetic programming and gene expression programming techniques offer the advantage of producing explicit analytical formulas for determination of output parameters, they are found to be grossly inaccurate with errors up to 7% for most accurate model developed. Artificial neural network, a prominent AI-based method has been used to significantly improve friction factor predictions with a high accuracy of 0.004% equivalent to that obtainable with the non-AI based models. The most accurate models are among those developed using the non AI-based techniques with errors up to 1.04×10^{-10} %. There is still possibility of improving on the gains made using the artificial intelligence techniques.

Keywords: Colebrook Equation, Artificial Intelligence, Genetic Algorithm, Artificial Neural Network and Gene Expression Programming

-----***-----

1. INTRODUCTION

The Colebrook equation [1, 2] is traditionally used to obtain friction factor (f) in pipe flows calculations. The friction factor, a dimensionless quantity, is a function of the Reynolds number (Re) only in the laminar flow regime and a function of both Re and relative roughness (ϵ/D) of the pipe in the turbulent flow regime. Unfortunately, this equation given in (1) is implicit, that is, having the friction factor on both sides of the equation and so has to be solved iteratively.

$$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{\epsilon/D}{3.71} + \frac{2.51}{\text{Re}\sqrt{f}} \right) \quad (1)$$

For calculations involving small datasets, the iteratively solved Colebrook equation will generally suffice [3]. For the simulation of long pipes and networks of pipes, the Colebrook equation must be solved a huge number of times [4]. Therefore, an iterative solution to the implicit Colebrook equation is time consuming. The use of the Moody chart [5], as an alternative to the Colebrook equation, eliminates the requirement for iteration. However, it is a graphical tool and therefore not convenient for computer-based simulations. In order to facilitate computer-based simulations, the implicit Colebrook equation has been resolved using explicit approximations derived from numerous analytical approaches [6, 7]. However, artificial intelligence (AI) techniques such as genetic algorithm [8], genetic programming [9-11], artificial neural network [12-18] and neuro-fuzzy logic [19] in recent times have been

applied. However, it is not obvious which approach is the best among the proposed explicit models and the modelling techniques.

In this paper therefore, the accuracies of the available AI-based and non AI-based explicit models for estimating hydraulic friction factor in the turbulent regime in the range for which the Colebrook equation is valid were closely examined.

The remaining sections of this paper are organized as follows: Section 2 reviews the accuracy of the available friction factor approximations. In section 3, the artificial intelligence techniques are examined and the performances of these techniques are discussed. Section 4 compares the performances of the explicit artificial intelligence-based and non artificial intelligence-based friction factor models. In the final section, relevant conclusions are drawn and recommendations are made.

2. FRICTION FACTOR MODELS

Friction factor models are referred to as explicit models by some authors [6, 7] that is, those which were developed using techniques other than the AI techniques. However, in this paper, any model devoid of iterations is regarded to as being explicit. Therefore, for the purpose of the study, the explicit friction factor models are classified into two groups viz: Artificial intelligence (AI)-based and non AI-based models.

2.1 Non AI-based Explicit Friction Factor Models

Several non AI-based approximations to the Colebrook equation have been developed. These approximations however vary in their degrees of accuracy and complexity. The most recent reviews include the work of Genić et al. [20], Brkić [6], Winning and Coole [7] and, Asker, Turgut and Coban [21].

Genić et al. [20] in their review used the model selection criteria (MSC) and the Akaike information criterion (AIC) as the performance indices for 16 explicit equations. These indices have also been used by Romeo, Royo and Monzon [22] for the explicit model selection. However, there is an apparent discrepancy in the MSC values reported by Romeo, Royo and Monzon [22] and Genić et al. [20] for the same models. For example, the MSC values reported by Romeo, Royo and Monzon [22] and Genić et al. [20] for models developed by Moody [23] and Chen [24] showed a wide contrast. They recommended the use of Zigrang and Sylvester [25] approximation in lieu of the implicit Colebrook equation.

Brkić [6] conducted a review of 26 explicit approximations. Based on maximum relative error (MRE) criterion, he classified the existing explicit models as extremely accurate ($MRE \leq 0.14\%$), very accurate (MRE up to 0.5%), moderately accurate (MRE up to 1.5%), less accurate (MRE up to 5%), non advisable (MRE up to 25%) and extremely inaccurate ($MRE \geq 80\%$). The most accurate models in his work were those of Serghides [26], Romeo, Royo and Monzon [22], Buzzelli [27], Zigrang and Sylvester [25] and Vantankhah and Kouchakzadeh [28]. He concluded with the remarks that most available explicit models are very accurate with the exception of models by Round [29], Eck [30], Moody [23], Wood [31] and, Rao and Kumar [32].

The review by Winning and Coole [7] assessed the accuracy alongside the relative computational efficiency of 28 explicit

friction factor approximations. They presented a more practical approach of determining computational efficiency by comparing the time required by a model to perform a given task. Elsewhere, model efficiency or complexity has been determined [33] by making reference to either the number of algebraic notation calculation key stroke. The authors observed that models with greater number of internal iterations were more accurate and require a greater computational time. The approximations by Serghides [26] and Buzzelli [27] were the most accurate when ordered by absolute and relative errors, but when ordered by relative computational efficiencies, they ranked very low. The overall ranking reported was biased since it was not based on actual values of accuracy and relative computational efficiency. It was based on the number of available explicit models. If this number is altered, the values of the combined ranking may change [34]. Unfortunately, neither of the above comparative studies considered the approximation by Goudar and Sonnad [35].

Vantankhah and Kouchakzadeh [28] modified the parameters in the model earlier developed by Sonnad and Goudar [36]. This modification increased the accuracy from a maximum relative error of 0.8% to about 0.15% (see Fig-1) [6]. In 2013, another set of more accurate models were proposed [37]. These models given in equations (28) and (29) were reported to have accuracy up to 0.02% and 0.008%, respectively with the latter being more complex than the former. The performances of these two models were evaluated in this present study. Despite their varying complexities, it was observed that they both have approximately the same degree of accuracy with maximum errors up to 0.05% and 0.059%, respectively. This means that the increased complexity of equation (29) does not pay off. Model accuracy is dependent on the complexity of their functional form, with the more complex ones providing friction factor estimates of higher accuracy [19].

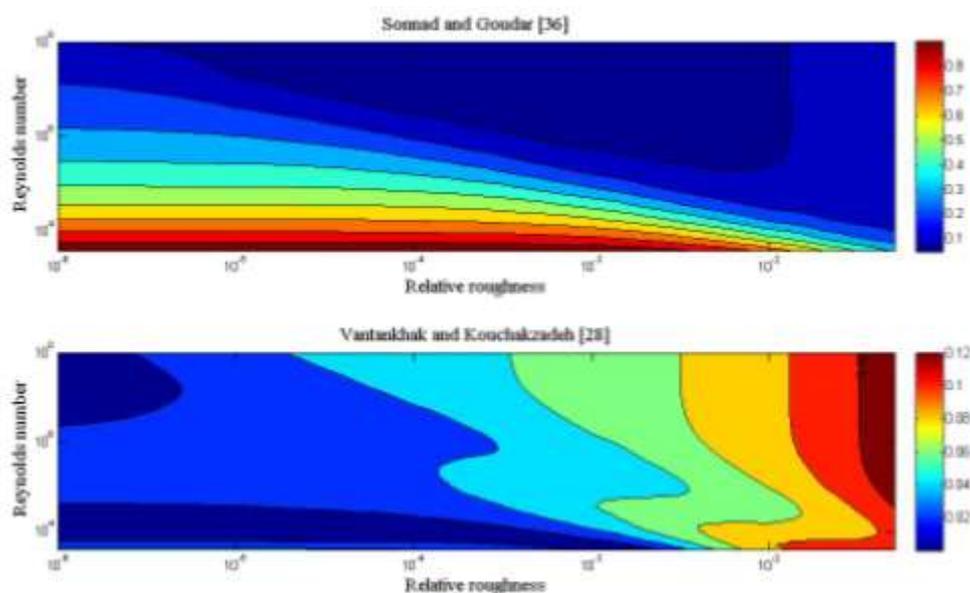


Fig – 1: Relative errors distribution of Sonnad and Goudar [36], Vantankhah and Kouchakzadeh [28] when compared with the implicit Colebrook equation

Asker, Turgut and Coban [21] conducted a review of several explicit approximations of the Colebrook equation. This review included the approximations developed by Goudar and Sonnad [35]. Based on relative error criterion, their models ranked best with very high accuracy when compared with other explicit approximations (see Fig-2). Their approximations, given in equation (20) and equation (21),

vary in terms of complexity with (21) being more complex than (20). However, this complexity is compensated for by its higher accuracy. This review concluded that the need to use the implicit Colebrook equation seems eliminated given the high accuracy of this model. A list of the available explicit approximations is presented in Table-1

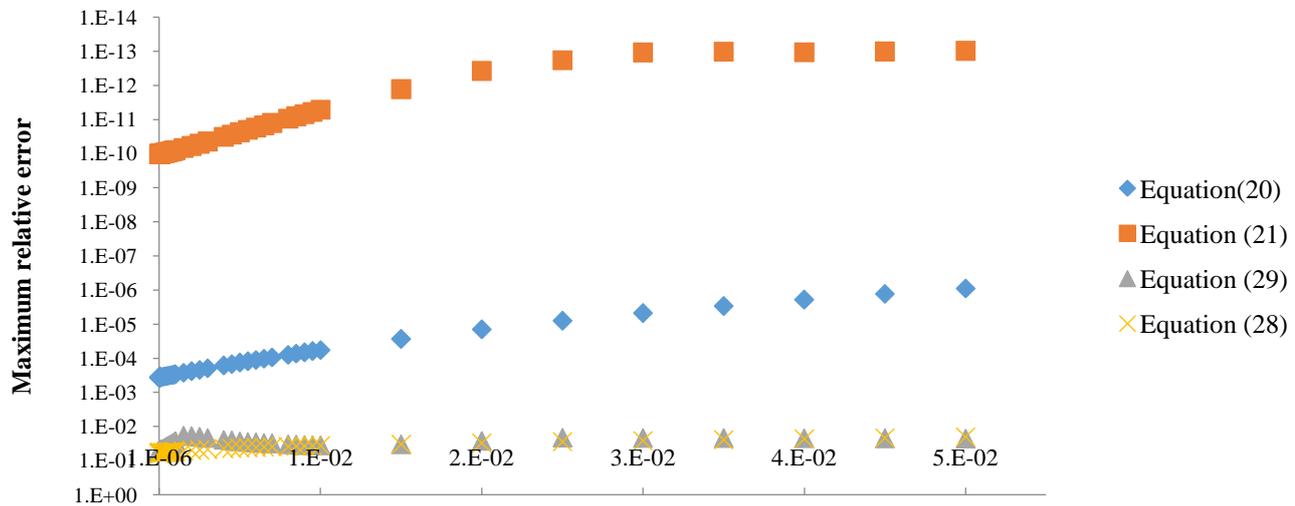


Fig – 2: Relative errors distribution of Sonnad and Goudar [35] (eqns. 20 and 21) and Vantankhah [37] (eqns. 28 and 29) when compared with the implicit Colebrook Equation

Table -1: Existing Non AI-based Friction Factor Models

Equation No.	Author [Reference]	Explicit Models	Applicable ranges of Re and ε/D
(2)	Moody [23]	$f = 0.0055 \left(1 + \left(2 \times 10^4 \frac{\epsilon}{D} + \frac{10^6}{Re} \right)^{1/3} \right)$	$4 \times 10^3 \leq Re \leq 10^8$ $0 \leq \frac{\epsilon}{D} \leq 10^{-2}$
(3)	Wood [31]	$f = 0.094 \left(\frac{\epsilon}{D} \right)^{0.225} + 0.53 \left(\frac{\epsilon}{D} \right) + 88 \left(\frac{\epsilon}{D} \right)^{0.4} Re^{-B}$ where $B = 1.62 \left(\frac{\epsilon}{D} \right)^{0.134}$	$4 \times 10^3 \leq Re \leq 5 \times 10^7$ $10^{-5} \leq \frac{\epsilon}{D} \leq 4 \times 10^{-2}$
(4)	Eck [30]	$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{\epsilon}{3.715D} + \frac{15}{Re} \right)$	$0 \leq \frac{\epsilon}{D} \leq 10^{-2}$
(5)	Swamee and Jain [39]	$\frac{1}{\sqrt{f}} = -2 \log \left(\left(\frac{\epsilon}{3.7D} \right)^{1.11} + \left(\frac{5.74}{Re^{0.9}} \right) \right)$	$5 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(6)	Churchill [40]	$f = 8 \left(\left(\frac{8}{Re} \right)^{12} + (A + B)^{-3/2} \right)^{1/12}$; where $A = \left[-2 \log \left(\left(\frac{\epsilon/D}{3.70} \right) + \left(\frac{7}{Re} \right)^{0.9} \right) \right]^{16}$, $B = \left(\frac{37530}{Re} \right)^{16}$.	$Re > 0$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(7)	Chen [24]	$\frac{1}{\sqrt{f}} = -2 \log_{10} \left(\frac{\epsilon}{3.7065 \cdot D} - \frac{5.0452}{Re} \cdot \log_{10} \left(\frac{1}{2.8257 \cdot \left(\frac{\epsilon}{D} \right)^{1.1098}} + \frac{5.8506}{Re^{0.8981}} \right) \right)$	$4 \times 10^3 \leq Re \leq 4 \times 10^8$ $10^{-7} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(8)	Shacham [41]	$\frac{1}{\sqrt{f}} = -4 \log \left[\frac{\epsilon}{3.7D} - \frac{5.02}{Re} \log \left(\frac{\epsilon}{3.7D} + \frac{14.5}{Re} \right) \right]$	$4 \times 10^3 \leq Re \leq 4 \times 10^8$

Equation No.	Author [Reference]	Explicit Models	Applicable ranges of Re and ϵ/D
(9)	Round [29]	$\frac{1}{\sqrt{f}} = 1.8 \log \left(\frac{Re}{0.135 Re \left(\frac{\epsilon}{D} \right) + 6.5} \right)$	$4 \times 10^3 \leq Re \leq 10^8$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(10)	Barr [42]	$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{\epsilon}{3.7D} + \frac{4.518 \log \left(\frac{1}{7} Re \right)}{Re \left(1 + \frac{1}{29} Re^{0.52} \left(\frac{\epsilon}{D} \right)^{0.7} \right)} \right)$	Not specified
(11)	Zigrang and Sylvester [25]	$\frac{1}{\sqrt{f}} = -2 \log_{10} \left(\frac{\epsilon}{3.7D} - \frac{5.02}{Re} \cdot \log_{10} \left(\frac{\epsilon}{3.7D} - \frac{5.02}{Re} \cdot \log_{10} \left(\frac{\epsilon}{3.7D} + \frac{13}{Re} \right) \right) \right)$	$4 \times 10^3 \leq Re \leq 10^8$ $4 \times 10^{-5} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(12)	Serghides [26]	$f = \left(s_1 - \frac{(S_2 - S_1)^2}{S_3 - 2 \cdot S_2 + S_1} \right)^{-2}$ $s_1 = -2 \log_{10} \left(\frac{\epsilon}{3.71D} + \frac{12}{Re} \right), s_2 = -2 \log_{10} \left(\frac{\epsilon}{3.71D} + \frac{2.51 \cdot S_1}{Re} \right),$ $s_3 = -2 \log_{10} \left(\frac{\epsilon}{3.7D} + \frac{2.51 \cdot S_2}{Re} \right)$	Not specified
(13)	Cojbasic and Brkić [8] ^a	$f = \left(s_1 - \frac{(S_2 - S_1)^2}{S_3 - 2 \cdot S_2 + S_1} \right)^{-2}$ $s_1 = -2 \log_{10} \left(\frac{\epsilon}{3.71D} + \frac{12.585}{Re} \right), s_2 = -2 \log_{10} \left(\frac{\epsilon}{3.71D} + \frac{2.51 \cdot S_1}{Re} \right), s_3 = -2 \log_{10} \left(\frac{\epsilon}{3.71D} + \frac{2.51 \cdot S_2}{Re} \right)$	Not specified
(14)	Romeo, Royo and Monzon [22]	$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{\epsilon/D}{3.7065} - \frac{5.0272}{Re} \log \left(\frac{\epsilon/D}{3.827} - \frac{4.567}{Re} \log \left(\left(\frac{\epsilon/D}{7.7918} \right)^{0.9924} + \left(\frac{5.3326}{208.815 + Re} \right)^{0.9345} \right) \right) \right)$	$3 \times 10^3 \leq Re \leq 1.5 \times 10^8$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(15)	Cojbasic and Brkić [8] ^a	$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{\epsilon/D}{3.7065} - \frac{5.0272}{Re} \log \left(\frac{\epsilon/D}{3.827} - \frac{4.567}{Re} \log \left(\left(\frac{\epsilon/D}{7.7918} \right)^{0.9924} + \left(\frac{5.3326}{208.815 + Re} \right)^{0.9345} \right) \right) \right)$	$3 \times 10^3 \leq Re \leq 1.5 \times 10^8$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(16)	Haaland [43]	$\frac{1}{\sqrt{f}} = -1.8 \log \left(\left(\frac{\epsilon}{3.7D} \right)^{1.11} + \frac{6.9}{Re} \right)$	$4 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(17)	Manadilli [44]	$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{\epsilon}{3.70D} + \frac{95}{Re^{0.983}} - \frac{96.82}{Re} \right)$	$5.235 \times 10^3 \leq Re \leq 10^8$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(18)	Sonnad and Goudar [36]	$\frac{1}{\sqrt{f}} = 0.8686 \ln \left(\frac{0.4587 Re}{S^{(S/S+1)}} \right); \text{ where } S = 0.124 Re(\epsilon/D) + \ln(0.4587 Re)$	$4 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$

Equation No.	Author [Reference]	Explicit Models	Applicable ranges of Re and ϵ/D
(19)	Vantankhah and Kouchakzadeh [28]	$\frac{1}{\sqrt{f}} = 0.8686 \ln \left(\frac{0.4587 Re}{(S-0.31)^{(S/S+0.9633)}} \right)$ where $S = 0.124 Re(\epsilon/D) + \ln(0.4587 Re)$	$4 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(20)	Goudar and Sonnad [35]	$\frac{1}{\sqrt{f}} = a \left[\ln \left(\frac{d}{q} \right) + \delta_{LA} \right]$ <p>Where $a = \frac{2}{\ln(10)}$; $d = \left(\frac{\ln(10)}{5.02} \right) \cdot Re$; $q = s^{\left(\frac{s}{s+1} \right)}$; $s = bd + \ln(d)$; $\delta_{LA} = \left(\frac{g}{g+1} \right) z$;</p> $g = bd \ln \left(\frac{d}{q} \right) \text{ and } z = \ln \left(\frac{q}{g} \right)$	$4 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(21)	Goudar and Sonnad [35]	$\frac{1}{\sqrt{f}} = a \left[\ln \left(\frac{d}{q} \right) + \delta_{CLA} \right]$ <p>Where $\delta_{CLA} = \delta_{LA} \left(1 + \frac{z/2}{(g+1)^2 + (z/3)(2g-1)} \right)$</p>	$4 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(22)	Buzzelli [27]	$\frac{1}{\sqrt{f}} = A - \left[\frac{A + 2 \log_{10} \left(\frac{B}{Re} \right)}{1 + \left(\frac{2.18}{B} \right)} \right]$; where $A = \frac{(0.774 \ln(Re))^{-1.41}}{(1 + 1.32 \sqrt{\frac{\epsilon}{D}})}$, $B = \left(\frac{\epsilon Re}{3.7 D} \right) + 2.51 A$	$3 \times 10^3 \leq Re \leq 1.5 \times 10^8$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(23)	Fang, Xu and Zhou [45]	$f = 1.613 \left[\ln \left(0.234 \left(\frac{\epsilon}{D} \right)^{1.1007} - \frac{60.525}{Re^{1.1105}} + \frac{56.291}{Re^{1.0712}} \right) \right]^{-2}$	$3 \times 10^3 \leq Re \leq 1.5 \times 10^8$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(24)	Rao and Kumar [32]	$\frac{1}{\sqrt{f}} = 2 \log_{10} \left(\frac{\left(\frac{\epsilon}{D} \right)^{-1}}{\left(\frac{0.444 + 0.135 Re}{Re} \right)^\beta} \right)$ where $\beta = 1 - 0.55 e^{-0.33 \left[\ln \left(\frac{Re}{6.5} \right) \right]^2}$	Not specified
(25)	Brkić [38]	$\frac{1}{\sqrt{f}} = -2 \log_{10} \left(10^{-0.4343 \beta} + \frac{\epsilon}{3.71 D} \right)$ where $\beta = \ln \frac{Re}{1.816 \ln \left(\frac{1.1 Re}{\ln(1 + 1.1 Re)} \right)}$	Not specified
(26)	Offor and Alabi [34]	$\frac{1}{\sqrt{f}} = -2 \log_{10} \left(\frac{\epsilon}{3.71 D} - \frac{1.975}{Re} \left(\ln \left(\left(\frac{\epsilon}{3.93 D} \right)^{1.092} + \left(\frac{7.627}{Re + 395.9} \right) \right) \right) \right)$	$4 \times 10^3 \leq Re \leq 10^8$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(27)	Ghanbari, Farshad and Rieke [46]	$f = \left[-1.52 \log \left(\left(\frac{\epsilon}{7.21} \right)^{1.042} + \left(\frac{2.731}{Re} \right)^{0.9152} \right) \right]^{-2.169}$	$2.1 \times 10^3 \leq Re \leq 10^8$ $0 \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(28)	Vantankhah [37]	$\frac{1}{\sqrt{f}} = 0.8686 \ln \left(\frac{0.4599 Re}{(G - 0.2753)^{G/g + 0.9741}} \right)$ <p>Where $G = 0.124 Re(\epsilon/D) + \ln(0.4599 Re)$</p>	$5 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$
(29)	Vantankhah [37]	$f = \left(\frac{2.51/Re + 1.1513 \delta}{\delta - \frac{\epsilon/D}{3.71} - 2.3026 \delta \log(\delta)} \right)^2$ <p>Where $\delta = \frac{6.0173}{Re(0.07(\epsilon/D) + Re^{-0.885})^{0.109}} + \frac{\epsilon/D}{3.71}$</p>	$5 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \frac{\epsilon}{D} \leq 5 \times 10^{-2}$

^a. Model Parameters have been modified using AI technique.

3. ARTIFICIAL INTELLIGENCE BASED

Artificial intelligence (AI) broadly refers to the ability to mimic or replicate the human behaviour / reasoning into machines and software using cutting edge techniques. AI technologies have matured to the point of offering real

practical benefits in many of their applications [47]. In particular, much work has been done to employ AI such as artificial neural networks (ANNs), genetic algorithm/programming, gene expression programming, data mining and evolutionary algorithms (EAs) to calculate pipe flow friction factors. Indeed, such technique proves to

be a powerful tool in solving non-linear, transcendental, or otherwise complex mathematical relations such as the Colebrook–White equation [2].

3.1 Artificial Neural Network (ANN) Model

Artificial neural network (ANN) is an authoritative modelling technique that exhibits analogies to the way arrays of neurons function in the biological and learning memory. ANN offers numerous benefits over conventional modelling techniques because they can model based on no assumptions or prior knowledge concerning the phenomenological mechanisms and understanding the mathematical background of the problem underlying the process and the ability to learn linear and non-linear relationships between variables directly from a set of examples [48].

A typical artificial neural network has three distinct layers namely the input, hidden and output layers respectively. It can be trained to perform a particular function by adjusting the values of the connections (weights) between elements so that a particular input leads to a specific target output. Several training techniques exist, but the most commonly used is the back propagation network [12-18]. **Table -2** gives a list of ANN models and their respective accuracies for friction factor estimation. Besides friction factor modelling, the use of ANN has been reported [48-51]. However, Salmasi, Khatibi and Ghorbani [9], based on their research, are of the opinion that the application of ANN in predicting friction factor has not been successful.

The earliest work on the use of ANN for predicting friction factor in pipe flow for Newtonian fluids dates back to the work of Shayya and Sablani [12]. They used a combination of 40 Re and 43 e/D parameters resulting in a total of 1720 input data points. The best network architecture for their work (with 3 hidden layers) was obtained using the transformed input parameters (Re and e/D) to a logarithmic scale and has a maximum relative error of about 1.22% when compared to the implicit Colebrook equation.

Fadare and Ofidhe [15] also developed an ANN model for predicting pipe friction factor estimation in the fully developed turbulent flow regime. A total of 1920 input/output normalized datasets was used for the training. Their results showed that the network with 2-20-31-1 configuration trained with the Levenberg-Marquardt function had the best performance with R^2 -value (0.999), maximum absolute error (0.68%), mean square error (MSE) of 5.335×10^{-7} and sum of squared error (SSE) of 3.414×10^{-4} . These error indices show their work has a significant improvement over the work of Shaya and Sablani [12].

Yazdi and Bardi [16] used 2000 normalized input/output datasets. The best network architecture were 2-10-20-1 and 2-20-25-1 with MSE values of 3.14×10^{-8} and $3.39 \times$

10^{-9} , respectively. However, in terms of relative error, it has error up to 3.5%. In addition to the evaluation of the network's accuracy, this work also considered the training time (the time required to process an input/data set to get a desired result based on predefined network architecture) for developing the network. The computational burden of training the data set is of no significance since it does not reflect the efficiency (computational speed / complexity) of the ANN model when put to use for friction factor estimation.

It is observed from the above studies, that the range of the Reynolds number chosen for the inputs all fall outside the applicable range of the Colebrook equation [17]. Although the works are similar, the differences lie in the normalization techniques employed (See **Table-2**). Normalization helps to put the datasets within the same order of magnitude. The quality of an ANN is largely dependent on the quantity of available data for training, the structure or architecture of the network.

ANN is considered a black box, because while it can approximate any function, studying its structure would not give any insight on the structure of the function being approximated. Most researchers have adopted this approach in the development of their ANN models. Besarati et al. [3] views this approach as defeating the aim of explicitly resolving the Colebrook equation given the intricate nature of these models. Sequel to this, they applied the group method of data handling (GMHD) to the development of an artificial neural network optimized by multi-objective genetic algorithms to find an explicit polynomial model for friction factor. A total of 250 input/output datasets were used in their work. The polynomial model developed therein has error up to 3.4%.

In the recent study conducted by Brkić and Čojbašić [17], an ANN model was developed to predict friction factor. Their research utilized a total of 90,000 data sets of input and output, respectively. The best network architecture for their work is 2-50-1 with a maximum relative error of 0.07% compared with the implicit Colebrook equation.

Offor and Alabi [18] also developed an ANN model for friction factor prediction using the multilayer perceptron network. The best network architecture obtained in their study was a 2-30-30-1 trained with 60,000 datasets with a maximum relative error of 0.004% compared with the implicit Colebrook equation.

These models [17,18] is found to outperform most of the available non AI-based models with the exception of those developed by Brkić and Čojbašić [8], Gouder and Sonnand [35], Vantankhah and Kouchakzadeh [28] and Vantankhah [37].

Table -2: Available ANN Architectures for approximating the Colebrook equation

Author [Reference]	Structure	R ²	MRE ^a (%)	MAE	MSE	SSE	Max. RE (%)
Shayya and Sablani [12]	2-14-14-14-1	0.9996	1.22	2.28e-6	-	-	
Sablani, Shayya and Kacimov [13]	2-8-8-1	0.9870	2.01	0.000079	-	-	
Sablani, Shayya and Kacimov [13]	2- 4-1	0.972	3.16	0.00105	-	-	
Shayya, Sablani and Capo [14]	2-12-12-12-1	0.9899	1.73	0.00013	-	-	
Ozger and Yildirim [19]	2-6-9-9-1-1	-	0.68	-	-	-	
Fadare and Ofidhe [15]	2-20-31-1	-	0.68	-	5.335E-7	3.414E-04	
Yadzi and Bardi [16]	2-20-25-1	-	0.35	-	3.339E-9		
Besarati, Myers, Covey and Jamali [3]	2-2-1	0.9954	3.4	-	-	-	
Brkić and Čojbašić [17]	2-50-1	-	0.0018	3.657E-07	4.067E-13	2.440E-08	0.07
Offor and Alabi [18]	2-30-30-1	1	1.99E-04	3.89E-08	2.46E-15	1.48E-10	0.004

^a MAPE (Mean absolute percentage error) is the same as MRE (mean relative error)

3.2 Evolutionary Algorithms

This technique uses the mechanisms inspired by biological evolution such as reproduction, mutation, recombination and selection. These include genetic algorithm, genetic programming and gene expression programming.

3.2.1 Genetic Algorithms

Genetic algorithms are one of the evolutionary computational intelligence techniques inspired by Darwin's theory of biological evolution which is based on the survival of the fittest. Given a population of individuals which represents possible solutions to a problem, the population is modified (evolved) by repeatedly selecting the fittest solution and producing new ones from them [52]. This new solutions replaces the existing ones until the best is obtained. Genetic algorithms are a global search method which works well in any search space.

3.2.2 Gene Expression Programming

Gene expression programming (GEP) is a technique that allows the solution of problems by automatically generating algorithms and expressions. These expressions are coded or represented as a tree structure with its terminals (leaves) and nodes (functions). GP applies GAs to a "population" of programs, i.e., typically encoded as tree-structure. Trial programs are evaluated against a "fitness function" and the best solutions selected for modification and re-evaluation. This modification-evaluation cycle is repeated until a "correct" program is produced [9].

3.2.3. Genetic Programming

It is an application of genetic algorithm (GA) to problems where each individual in a population represents a computer program. The programs are then encoded as a set of genes that are modified (evolved) using genetic algorithm (GA). GA evolves solutions to a given problem while GP aims at evolving computer programs that can solve a given problem [48].

The fundamental difference between GA, GP, and GEP is due to the nature of the individuals: in the GA, the

individuals are linear strings of fixed length (chromosomes); in GP the individuals are nonlinear entities of different sizes and shapes (parse trees); and in GEP the individuals are encoded as linear strings of fixed length (the genome or chromosomes), which are afterwards expressed as nonlinear entities of different sizes and shapes [43]. While genetic algorithm is solely a search algorithm, genetic programming and gene expression programming in addition offers the advantage of producing explicit models for determination of output parameters [9, 52].

Salmasi [9] applied the GP technique in developing a friction factor model. A total of 2072 data points which comprised 74 Re and 28 e/D values ranging from 2000 to 10^8 and 10^{-6} to 0.09, respectively, were used. The reported accuracy based on the selected data points was a maximum relative error of 2.64×10^{-12} . This model however, fails to reproduce the same level of accuracy over a dataset different from that used to develop the model. Thus, the model has a poor generalization capability. Till date, friction factor models developed using the GP and GEP techniques have been marked by a high degree of inaccuracy.

Samadianfard [10] proposed a model for estimating flow friction factor using GEP based technique. However, it has been observed [37] that the model is not very accurate and has errors up to 7%, which has been attributed to inaccurate solution of the implicit Colebrook's equation which was used in gene expression programming (GEP) by the author. A list of the available GP/GEP models is presented in Table -3.

Evolutionary algorithms can be utilized as search algorithms to enhance the performances of friction factor models. Čojbašić and Brkić [8] have used genetic algorithm to optimize the model parameters of the models developed by Serghides [26] (12) and Romeo, Royo and Monzon [22] (14) as shown in **Fig-3**. This technique reduced the relative errors from 0.1385% and 0.1345% to 0.0026% and 0.0083% for (12) and (14), respectively [8]. The modification of (12) is not as accurate as reported; however, the accuracy of (12) was significantly improved.

Table -3: Available GP/GEP Architectures for approximating the Colebrook Equation

Equation no.	Author [Reference]	GP/GEP based explicit models	Applicable ranges of Re and ϵ/D
30	Salmasi, Khatibi and Ghorbani [9]	$f = -0.0575 + \frac{\epsilon}{D} + e^{-11.764(\epsilon/D) - \log(2R_n)}$ $+ e^{-2.567 + 9.065/R_n - \epsilon/D}$	$2 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \epsilon/D \leq 0.05$
31	Samadianfard [10]	$f = \left(\frac{Re^{\epsilon/D} - 0.6315093}{Re^{1/3} + Re * \epsilon/D} \right)$ $+ 0.0275308 \left(\frac{6.929841}{Re} + \epsilon/D \right)^{1/9}$ $+ \left(\frac{10^{\epsilon/D}}{\epsilon + 4.781616Re} \right) \left(\sqrt{\frac{\epsilon}{D}} + \frac{9.99701}{Re} \right)$	$4 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \epsilon/D \leq 0.05$
32	Samadianfard, Taghi, Kisi and Kazemi [11]	$f = 152.137 \times (\epsilon/D)^3 + 1223 \times (\epsilon/D)^4 + \frac{(9.96213 + Re)}{Re^2}$ $+ \arctan \left[\frac{-8.79056 \times (\epsilon/D)^2 \times (Re - 9.72464)}{9.72464 + Re} \right]$ $+ \frac{0.0834528 \left((\epsilon/D) \times \sqrt{Re} \right)^{2/3}}{\ln \left[(\epsilon/D)^3 \right]^2}$ $+ \frac{\arctan \left[8.3663 \times (\epsilon/D) \times \sqrt{Re} \right]^2}{\ln \left[(\epsilon/D)^3 \right]^2}$ $+ \arctan \left[\sqrt{Re} \sin \left[(\epsilon/D) \right] \right]^2 + \left[\frac{\ln \left[\frac{Re}{(\epsilon/D)} \right]}{Re - 9.96213} \right]$ $+ \arctan \left[(\epsilon/D) \sin \left[\frac{\arctan \left(\frac{0.447021}{(\epsilon/D)} \right)}{(\epsilon/D) \times Re} \right] \right]$ $+ \sin \left[(\epsilon/D) \times \arctan \left[\frac{\arctan \left(\frac{2.70969}{(\epsilon/D)} \right)}{(\epsilon/D) \times Re} \right] \right]$ $- 2 \sin \left[(\epsilon/D) - \frac{\arctan \left[\arctan \left(\frac{0.488525}{(\epsilon/D)} \right) \right]}{\arctan \left[\arctan (Re) \right]} \right] + \sin \left[\frac{\arctan \left(\frac{0.488525}{(\epsilon/D)} \right)}{9.98978 + \ln(Re)} \right]^2$	$4 \times 10^3 \leq Re \leq 10^8$ $10^{-6} \leq \epsilon/D \leq 0.05$

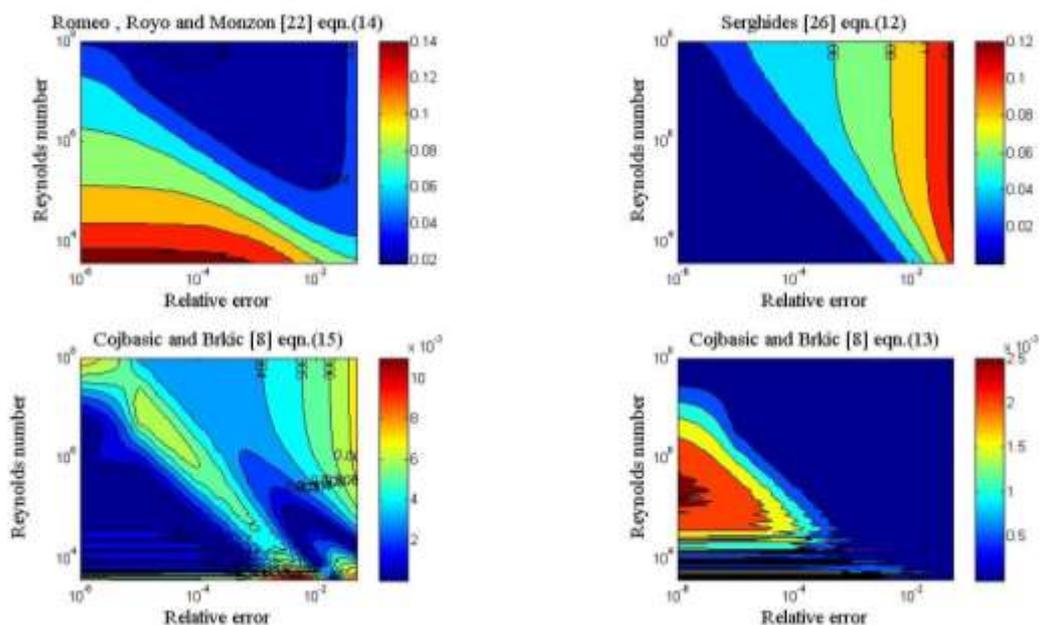


Fig – 3: Relative errors distribution of the models by Romeo, Royo and Monzon [22], Serghides [26] and Cojbasic and Brkić[8] when compared with the implicit Colebrook equation

Davidson [52] developed a polynomial friction factor model using genetic programming to fit the implicit Colebrook equation and the model parameters therein were determined using numerical method. A total of 100 transformed data sets consisting of 10 data points each of Re and (ϵ/D) values in the ranges 100000-1000000 and 0.001-0.01, respectively were used. The 14-term polynomial was reported to have maximum absolute error value of 0.000194 compared with the implicit Colebrook equation. This value compares well with those obtained from most of the moderately accurate non AI- based explicit approximations.

3.2.4 Artificial Neuro Fuzzy Inference System (ANFIS)

This is a soft computing method in which a given input-output data set is expressed in a fuzzy inference system. It is a kind of artificial intelligence which integrates both neural network and fuzzy logic principles i.e. it uses a hybrid learning algorithm. It has the advantage of allowing users to

insert prior knowledge into a neural network as a rule, which is not obtainable with the conventional ANN. The approach was used by Ozger and Yildirim [19] to develop an ANN model to approximate the Colebrook equation. The best network topology from their work gave an MRE value of 0.68%. The results obtained from their work compares well only with the less accurate non AI-based explicit approximations.

4.0 Comparative Evaluation of the Explicit non AI-based and AI-based Friction Factor Models

Using the maximum relative error criterion, a comparison of all the available non AI-based models was made. It is shown that the most accurate non AI-based models are those by Goudar and Sonnad [35] and, Serghides [26] as modified by Cojbasić and Brkić [8] (13) using genetic algorithm (see Table-4).

Table -4: Non AI-based explicit models ordered by maximum relative error

Equation no.	Author [Reference]	Maximum absolute error	MSE	SSE	Max. RE (%)
(29)	Goudar and Sonnad [36]	4.14E-14	3.90E-29	1.17E-24	1.04E-10
(28)	Goudar and Sonnad [36]	1.45E-07	5.99E-16	1.80E-11	3.64E-04
(13)	Cojbasić and Brkić [8]	4.57E-07	1.35E-14	4.06E-10	0.0026
(15)	Cojbasić, and Brkić [8]	7.21E-06	2.82E-12	8.46E-08	0.0115
(28)	Vantankhah [37]	1.99E-05	1.93E-11	5.78E-07	0.0499
(29)	Vantankhah [37]	2.38E-05	1.02E-11	3.07E-07	0.0596
(26)	Offor and Alabi [34]	2.36 E-05	2 .86E-11	1.71 3E-06	0.0664
(12)	Serghides [26]	8.97E-05	7.90E-10	4.74	0.1225
(22)	Buzzelli [27]	8.98E-05	9.08E-10	2.73E-05	0.1255
(11)	Zigrang and Sylvester [25]	8.97E-05	3.47E-09	2.82E-05	0.1255
(19)	Vantankah and Kouchakzadeh [28]	9 .52E-05	8.80E-10	5.28E-05	0.1332
(14)	Romeo, Royo amd Monzon [22]	6.38E-05	4.82E-10	2.89E-05	0.1462
(7)	Chen [23]	1.26E-04	1.54E-09	4.78E-05	0.3559
(10)	Barr [42]	3.28E-04	2.11E-09	6.33E-05	0.5247
(28)	Fang, Xu and Zhou [45]	4.61E-04	4.35E-09	1.31E-04	0.5997
(8)	Shacham [41]	3.46E-04	4.41E-09	1.32E-04	0.8678
(18)	Sonnad and Goudar [36]	3.96E-04	1.35E-08	4.04E-04	0.9926
(16)	Haaland [43]	7.31E-04	2.32E-08	6.95E-04	1.4073
(27)	Ghanbari, Farshad and Rieke [46]	2.00E-03	2.11E-07	0.0066	2.7744
(17)	Manadilli [44]	2.10E-03	8.74E-08	0.0026	2.8174
(25)	Brkić [38]	2.18E-03	2.73E-07	0.0094	2.9427
(6)	Churchill [40]	2.03E-03	2.86E-07	0.0085	3.2178
(4)	Eck [30]	0.0034	4.28E-07	0.0257	8.1968
(9)	Round [29]	6.00E-03	2.24E-06	0.1341	10.2217
(2)	Moody [23]	0.0123	6.73E-09	0.4036	18.991
(3)	Wood [31]	0.0133	1.32E-06	0.0794	28.2335
(24)	Rao and Kumar [32]	3.84E-02	5.20E-05	1.5587	85.479

Table -5: AI-based GP/GEP explicit models ordered by maximum relative error

Equation no.	Author [Reference]	Maximum absolute error	MSE	SSE	Max. RE (%)
(30)	Salmasi, Khatibi and Ghorbani [9]	0.0057	4.295E-05	2.578	203.26
(31)	Samadianfard [10]	0.0024	2.163E-07	0.0125	7.439
(32)	Samadianfard, Taghi, Kisi and Kazemi [11]	199	3.734E+03	2.241E+08	2.785E+05

The AI-based models developed so far using GP and GEP have a high degree of inaccuracy when compared with the implicit Colebrook equation as shown in Table - 5. However, genetic algorithm has shown to be successful in optimization of model parameters. Although, the accuracy is largely dependent on the quality of the models whose parameters are to be optimized.

The most accurate among the ANN model developed so far for predicting friction factor in the turbulent regime are those by Offor and Alabi [18], followed by Brkić and Čojbašić [17]. These models are in close agreement with those of the extremely accurate non-AI based models while the rest of ANN models reported (see Table -3) in the literature have accuracies equivalent to those of the less accurate non AI-based models with the exception of those by Round [28], Eck [30], Moody [23], Wood [31] and Rao and Kumar [32], which are extremely inaccurate. Neuro-fuzzy logic when combined with ANN, despite its advantage of offering prior knowledge to the system, has accuracy within the range of the less accurate non-AI based models. This combination did not yield any significant improvement.

The accuracy margin between the non AI-based and AI-based models still remains wide, a value of $1.04 \times 10^{-10}\%$ against 0.004% based on maximum relative error criterion, when compared with the implicit Colebrook equation. However, the use of genetic algorithm (GA) to optimize neural networks parameters has been reported [54]. Based on the available literature, this can be used to reduce the margin. This can be achieved by adjusting the weights, network architecture or the learning rates [55]. The GA-trained ANN has been shown to give improved accuracy over the conventional ANN [55].

Amongst the artificial intelligence techniques so far employed in determining friction factor, ANN produced the best results while genetic algorithm has shown to be a viable tool for model parameters optimization (see (13) and (15) in Table-4). This is no surprise since GA has the capability of obtaining a global solution while the GEP models developed so far are marked by a high degree of inaccuracy as shown in Table-5.

In the last 50 years, attempts have been made to resolve the implicit Colebrook equation. This was without much success initially but at the turn of the 21st century, the evolved approximations closed the margin between the implicit model and the explicit approximations. Now there are models with very high accuracy which eliminates the need for the use of the implicit Colebrook equation.

5. CONCLUSION

The performances of artificial intelligence (AI)-based and non- AI based methods for resolving the implicit Colebrook equation were considered. The available AI-based techniques are genetic algorithm, gene expression programming and artificial neural networks. The most accurate friction factor models fall among the non AI-based models with a few exceptions which are highly inaccurate. Amongst the artificial intelligence approaches, ANN network gives the best accuracy followed by GA-trained non-AI based model while GEP and GP techniques are marked by a high degree of error, although they offer the advantage of producing explicit analytical formulas for determination of output parameters.

REFERENCES

- [1]. Colebrook, C. F. and White, C. M., "Experiments with fluid friction factor in roughened pipes." Proc. Roy. Soc. Ser. A Math. Phy. Sci. 161 (906); pp. 367-381. 1937.
- [2]. Colebrook, C. F. "Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws." Journal of the Institution of Civil Engineers, 11, pp. 133-156, 1939.
- [3]. Besarati, S. M., Myers, P. D., Covey, D. C. and Jamali, A., "Modelling friction factor in pipeline flow using a GMDH-type neural network." Cogent Engineering, pp. 1-14, 2015.
- [4]. Clamond, D., "Efficient resolution of the Colebrook equation." Industrial & Engineering Chemistry Research, 48, pp. 3665-3671, 2009.
- [5]. Moody, L. F., "Friction factors for pipe flow." Transactions of the American Society of Mechanical Engineers, 66, pp. 671-681. 1944.
- [6]. Brkić, D., "Review of Explicit Approximations to the Colebrook Relation for Flow Friction." Journal of Petroleum Science and Engineering, 77, pp. 34-48, 2011.
- [7]. Winning, H. K. and Coole, T., "Explicit friction factor accuracy and computational efficiency for turbulent flows in pipes." Flow, Turbulence and Combustion, 90, pp. 1-27, 2013.
- [8]. Čojbašić, Ž. and Brkić, D., "Very accurate explicit approximations for calculation of the Colebrook friction factor." International Journal of Mechanical Sciences, 67, pp. 10-13, 2013.
- [9]. Salmasi, F., Khatibi, R., and Ghorbani, M. A., "A study

of friction factor formulation in pipes using artificial intelligence techniques and explicit equations.” *Turkish Journal of Engineering and Environmental Science* , 36, pp. 121-138, 2012.

[10]. Samadianfard, S., “Gene expression programming analysis of implicit Colebrook–White equation in turbulent flow friction factor calculation.” *Journal of Petroleum Science and Engineering*, pp. 48 – 55. , 2012.

[11]. Samadianfard, S., Taghi, S. M., Kisi, O. and Kazemi, H., “Determining Flow Friction Factor in Irrigation Pipes using Data Mining and Artificial Intelligence Approaches.” *Applied artificial intelligence*, 28, pp. 793- 813, 2014.

[12]. Shayya, W. H., and Sablani, S., “An artificial neural network for non-iterative calculation of the friction factor in pipeline flow. *Computer and Electronics in Agriculture* ,21, pp. 219-228, 1998.

[13]. Sablani, S. S., Shayya, W. H. and Kacimov, A. (2003). Explicit calculation of friction factor in pipeline flow of Bingham plastic fluids: a neural network approach. *Chemical Engineering Science* , pp. 99-106, 2003

[14]. Shayya, W. H., Sablani, S. S. and Capo, A. (2005) Explicit Calculation of the friction factor for Non-Newtonian Fluids using Artificial Neural networks. *Developments in Chemical Engineering and Material Processing*, 13, pp. 5-20.

[15]. Fadare, D. and Ofidhe, U., “ Artificial neural network model for prediction of friction factor in pipe flow”. *Journal of Applied Science Research* ,5, pp. 662-670, 2009.

[16]. Yazdi, M. and Bardi, A., “ Estimation of Friction Factor in Pipe Flow using Artificial Neural Network.” *Canadian Journal on Automation, Control and Intelligent Systems* ,2, pp. 52 – 56, 2011.

[17]. Brkić, D. and Cojbašić, Ž., “ Intelligent Flow Friction Estimation. *Computer Intelligence Neuroscience*.” 2016.

[18]. Ofor, U. H. and Alabi, S. B., “Artificial neural network model for friction factor prediction.” *Journal of Materials Science and Chemical Engineering*, 4, pp. 77- 83, 2016. <http://dx.doi.org/10.4236/msce.2016.47011>.

[19]. Özger, M., & Yildirim, G., “Determining turbulent flow friction coefficient using adaptive neuro fuzzy computing techniques.” *Advances in Engineering Software*, 40, pp.281-287, 2009.

[20]. Genić, S., Arandjelović, I., Kolelendić, P., Jarić, M., Budimir, N., & Genić, V. “A Review of explicit approximations of Colebrook's equation.” *FME Transactions* , 39, pp. 67-71, 2011.

[21]. Asker, M., Turgut, O. E., & Coban, T. M., ‘A review of non iterative friction factor correlations for the calculation of pressure drops in pipes”. *Bitlis Eren University Journal of Science and Technology* , 4 , 1-8, 2014.

[22]. Romeo, E., Royo, C., and Monzon, A., “Improved explicit equation for estimation of the friction factor in rough and smooth pipes.” *Chemical Engineering Journal*, 86, pp. 369-374, 2002.

[23]. Moody, L. F., “ An approximate formula for pipe

friction factors”. *Transactions of the American Society of Mechanical Engineers* ,66, pp.1005-61011, 1947.

[24]. Chen, N. H., “An explicit equation for friction factors in pipes.” *Industrial and Engineering Chemistry fundamentals* , 18, pp. 296-297, 1979.

[25]. Zigrang, D. J. and Sylvester, N. D., “Explicit approximations to the solution of Colebrook’s friction factor equation.” *AIChE Journal* , 514-515, 1982.

[26]. Seghides, T. K., “Estimate friction factor accurately.” *Chemical Engineering Journal* , 91 , pp. 63- 64, 1984.

[27]. Buzzelli, D., “Calculating friction in one step.” *Machine Design*, 80, pp. 54-55, 2008.

[28]. Vantankhah, A. R. and Kouchakzadeh, S. (2008) Discussion of “Turbulent flow friction factor calculation using a mathematically exact alternative to Colebrook-White equation” by Jagadeesh R. Sonnad and Chetan T. Goudar. *Journal of Hydraulic Engineering*,134, 1187, 2008.

[29]. Round, G. F., “An explicit approximation for the friction factor-Reynolds number relation for rough and smooth pipes.” *Canadian Journal of Chemical Engineering*, 58, 122, 1980.

[30]. Eck, “*Technische Stromungslehre*.” Springer, New York, 1973.

[31]. Wood, D. J., “An Explicit Friction Factor Relationship.” ,60, *Civil Engineering*, 36, 60-61, 1966.

[32]. Rao, A. R. and Kumar, B., “Friction Factor for Turbulent Pipe Flow.” *Division of Mechanical sciences, Civil Engineering Indian Institute of Science, Bangalore, India*, 2007.

[33]. Zigrang, D. J. and Sylvester, N. D. “A Review of Explicit Friction Factor Equations.” *Journal of Energy Resources Technology*, 107, pp. 280-283, 1985.

[34]. Ofor, U. H. and Alabi, S. B., “An accurate and computationally efficient friction factor model.” *Advances in Chemical Engineering and Science* ,6, pp. 237- 245, 2016. <http://dx.doi.org/10.4236/aces.2016.63024>

[35]. Goudar, C. T. and Sonnad, J. R., “ Comparison of the iterative approximations of the Colebrook-White equation.” *Hydrocarbon Processing: Fluid and Rotating Equipment* , pp. 79-83, 2008.

[36]. Sonnad, J. R. and Goudar, C. T., “Turbulent Flow friction factor calculation using a mathematically exact alternative to the Colebrook-White equation.” *Journal of Hydraulic Engineering* , 132, pp. 863-867, 2006.

[37]. Vantankhah, A., Comment on "Gene Expression Programming Analysis of Implicit Colebrook-White equation in turbulent flow friction factor calculation". *Journal of Petroleum Science and Engineering* , pp. 402-405, 2013.

[38]. Brkić, D., “An Explicit Approximations to the Colebrook Relation for Flow Friction Factor.” *Journal of Petroleum Science and Technology*, 29, pp. 1596-1602, 2011.

[39]. Swamee, D. K. and Jain, A. K., “Explicit Equations for pipe flow problems. *Journal of the Hydraulics Division* , 102, pp. 657-664, 1976.

- [40]. Churchill, S. W., "Friction factor equations spans all fluid-flow regimes." *Chemical Engineering*, 84, 91-92, 1977.
- [41]. Schorle, B. J., Churchill S. W. and Shacham, M. (1980) Comments on: "An Explicit Equation for Friction Factor in Pipe," by Chen, N.H. *Ind. Eng. Chem. Fundamentals*, 19, pp. 228-229, 1908.
- [42]. Barr, D. I. H., "Solutions of the Colebrook-white function for resistance to uniform turbulent flow." *proceedings of the Institute of Civil Engineers, Part 2*, 71, pp. 529-535, 1981.
- [43]. Haaland, S. E., "Simple and explicit formulas for the friction factor in turbulent pipe flow." *Journal of fluids Engineering*, 105, pp. 89-90, 1983.
- [44]. Manadilli, G., "Replace implicit equations with signomial functions." *Chemical Engineering Journal*, 104, 129-130, 1997
- [45]. Fang, X., Xu, Y. and Zhou. Z., "New correlations of single-phase friction factor for turbulent pipe flow and evaluation of existing single-phase friction factor correlations." *Nuclear Engineering and Design*, 241, pp. 897-902, 2011.
- [46]. Ghanbari, A., Farshad, F. and Rieke, H., "Newly developed friction factor correlation for pipe flow and flow assurance." *Journal of Chemical Engineering and Material Science*, 2, pp. 83-86. 2011.
- [47]. Pannu, A., " Artificial intelligence and its application in different areas." *International Journal of Engineering and Innovative Technology* , 4 , pp. 79- 84, 2015.
- [48]. Goyal, S. and Goyal, G. K., "Artificial neural networks in food stuffs: a critical review." *Scientific Journal of Review* , pp. 147-155, 2012.
- [49]. Kaveh, M. and Chayjan, R. A, "Mathematical and neural network modelling of terebinth fruit under fluidized bed drying." *Research in Agricultural Engineering* ,61, pp. 55-65, 2015.
- [50]. Alabi, S. B. and Williamson C.J., "Neural Network-based Model for Joint Prediction of the Newtonian and Non-Newtonian Viscosities of Black Liquor." *International Journal of Chemical Engineering and Applications*, 6, pp.195-2001, 2015.
- [51]. Zuowei, W. and Xiaohui, W., "Mathematical and artificial neural network models to predict the membrane fouling behavior of an intermittently-aerated membrane bioreactor under sub-critical flux." *Clean- Soil, Air, Water*, 44, pp.1002-1009, 2015
- [52]. Davidson, J. W, Savic, D. and Walter, G. A. S., "Method for the identification of explicit polynomial formula for the friction in turbulent pipe flow." *ACM Journal of Hydro Informatics*, 1999.
- [53]. Ferreira, C., "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems." *Complex Systems*, Vol. 13, issue 2 , pp. 87-129., 2001.
- [54]. Sivapathasekaran, C., Mukherjee, S., Ray, A, Gupta, G., Sen, R. "Artificial neural network modelling and genetic algorithm based medium optimization for improved

production of marine biosurfactant." *Journal of Bioresource Technology* , 101, pp. 2884-2887, 2009.

[55]. Shifei, D., Li, X., Chunyang, S., & Zhu, H. "Using Genetic Algorithms to Optimize Artificial Neural Networks." *Journal of Convergence Information Technology* , 5, pp. 54-62, 2010.

Nomenclature

f	Darcy friction factor [dimensionless]
D	Internal pipe diameter [m]
ε	Pipe absolute roughness [m]
ε/D	Relative roughness (dimensionless)
Re	Reynolds number (dimensionless)
AI	Artificial intelligence
ANN	Artificial neural network
GA	Genetic algorithm
GEP	Gene expression programming
GP	Genetic programming