CI&CEQ

Available on line at
Association of the Chemical Engineers AChE
www.ache.org.rs/CICEQ

Chemical Industry & Chemical Engineering Quarterly 16 (4) 329–343 (2010) CI&CEQ

SANDIP KUMAR LAHIRI
K.C. GHANTA

Department of Chemical Engineering, NIT, Durgapur, West Bengal, India

SCIENTIFIC PAPER

UDC 66.026.2

DOI 10.2298/CICEQ091030034L

REGIME IDENTIFICATION OF SLURRY TRANSPORT IN PIPELINES - A NOVEL MODELING APPROACH USING ANN AND DIFFERENTIAL EVOLUTION

Four distinct regimes were found existent (namely sliding bed, saltation, heterogeneous suspension and homogeneous suspension) in slurry flow in pipelines depending upon the average velocity of flow. In the literature, few correlations have been proposed for identification of these regimes in slurry pipelines. Regime identification is important for slurry pipeline design as they are the prerequisite for applying different pressure drop correlations in different regimes. However, the available correlations fail to predict the regime over a wide range of conditions. Based on a databank of around 800 measurements collected from the open literature, a method has been proposed to identify the regime using artificial neural network (ANN) modeling. The method incorporates hybrid artificial neural network and differential evolution technique (ANN-DE) for efficient tuning of ANN Meta parameters. Statistical analysis showed that the proposed method has an average misclassification error of 0.03%. A comparison with selected correlations in the literature showed that the developed ANN-DE method noticeably improved prediction of regime over a wide range of operating conditions, physical properties, and pipe diameters.

Key words: artificial neural network; differential evolution; slurry flow regime; slurry flow.

Cut throat competition, minimization of energy requirement and optimization of production cost is the buzzword of today's chemical process industry. Transportation of slurries through pipeline is common in solid handling, mineral and petrochemical industries and its huge power consumption is drawing attention in recent years. The need and benefit of accurately predicting the pressure drop of slurry pipelines during the design phase is enormous as it gives better selection of slurry pumps and pipelines, better optimization of power consumption and helps to maximize economic benefit.

Great attention is being paid to reduction of the hydraulic losses as power consumption constitutes a substantial portion of operational costs for the overall pipeline transport. Four distinct regimes were found existent (namely sliding bed, saltation, heterogeneous suspension and homogeneous suspension) in slurry flow in pipelines depending upon the average velocity of flow. The pressure drop correlations available in literature are applicable to a particular regime for which they were developed. The correlations show bad prediction of pressure drop when they apply for other regimes. Thus regime identification becomes important for slurry pipeline design as they are the prerequisite for applying different pressure drop correlations in different regimes. The understanding of different regime formation and accurate prediction of pressure drop of slurries in design phase makes it possible to optimize energy and water requirements.

Experimental observations have shown that different correlations should be used in each of the identifiable flow regimes. Although this is a logical approach it is not straightforward to apply. The main difficulty arises because it is not easy to define the boundaries between the flow regimes. These boundaries are poorly defined because they are based on visual observations of particle motions in small laboratory pipelines. Many researchers have attempted to establish correlations among the relevant experimental variables that can be used to define the boundaries of the flow regimes. These attempts have met with only limited success and an approach developed by Turian and Yuan [1] is most popular and promising.

Correspondening author: S.K. Lahiri, Department of Chemical Engineering, NIT, Durgapur, West Bengal, India.
E-mail: sk_lahiri@hotmail.com
Paper received: 29 November, 2009
Paper revised: 20 April, 2010
Paper accepted: 22 June, 2010

Turian's approach claims to provide a completely self-consistent definition of the flow regime boundaries that results directly from the head loss correlations and no additional correlations are required to define the boundaries. The method claims that it is based on a large database of reliable experimental data and consequently the method can be used with confidence for practical engineering work. However a databank of around 800 measurements collected from the open literature is exposed to Turian and Yuan's calculations and the result of prediction of flow regimes is found very poor. These poor results motivated this work and an attempt has been made to develop a method which can identify the flow regime at different velocity and particle size distributions.

To facilitate the design and scale up of pipelines and slurry pumps, there is a need for a correlation that can predict flow regime over a wide range of operating conditions, physical properties and particle size distributions. The industry needs quick and easily implementable solutions. The model derived from the first principle is no doubt the best solution. But in the scenario where the basic principles of regime identification model accounting all the interactions for slurry flow is absent, the numerical model may be promising to give some quick, easy solutions for slurry flow regime prediction.

In the last decade, artificial neural networks (ANNs) have emerged as attractive tools for nonlinear process modeling especially in situations where the development of phenomenological or conventional regression models becomes impractical or cumbersome. ANN is a computer modeling approach that learns from examples through iterations without requiring a prior knowledge of the relationships of process parameters and, is consequently, capable of adapting to a changing environment. It is also capable of dealing with uncertainties, noisy data, and non-linear relationships. ANN modeling has been known as "effortless computation" and readily used extensively due to their model-free approximation capabilities of complex decision-making processes. Owing to their several attractive characteristics, ANNs have been widely used in chemical engineering applications such as steady state and dynamic process modeling, process identification, yield maximization, nonlinear control, and fault detection and diagnosis [2-10].

The most widely utilized ANN paradigm is the multi-layered perceptron (MLP) that approximates nonlinear relationships existing between an input set of data (causal process variables) and the corresponding output (dependent variables) data set. A three-layered MLP with a single intermediate (hidden) layer housing a sufficiently large number of nodes (also termed neurons or processing elements) can approximate (map) any nonlinear computable function to an arbitrary degree of accuracy. It learns the approximation through a numerical procedure called "network training" wherein network parameters (weights) are adjusted iteratively such that the network, in response to the input patterns in an example set, accurately produces the corresponding outputs. There exists a number of algorithms—each possessing certain positive characteristics—to train an MLP network, for example, the most popular error-back-propagation (EBP), quick prop and resilient back-propagation (RPROP) [11]. Training of an ANN involves minimizing a nonlinear error function (*e.g.*, root-mean-squared-error, RMSE) that may possess several local minima. Thus, it becomes necessary to employ a heuristic procedure involving multiple training runs to obtain an optimal ANN model whose parameters (weights) correspond to the global or the deepest local minimum of the error function. The building of a back-propagation network involved the specification of the number of hidden layers and the number of neurons in each hidden layer. In addition, several parameters including the learning rule, the transfer function, the learning coefficient ratio, the random number seed, the error minimization algorithm, and the number of learning cycles had to be specified.

Existing software implementations of ANN regression usually treat these ANN meta-parameters as user-defined inputs. For non expert users it is a very difficult task to choose these parameters as they have no prior knowledge for these parameters for their data. In such a situation, users normally rely on a trial and error method. Such an approach apart from consuming enormous time may not really obtain the best possible performance. The present paper addresses this issue and develops a new hybrid procedure to find the optimum ANN architecture and tune the ANN parameters and thus relieve the "non-expert" users.

Basically, the setting of optimum ANN architecture and tuning of ANN meta-parameters can be viewed mathematically as an optimization problem where test set errors (generalization error) have to be minimized. In the recent years, differential evolutions (DEs) that are members of the stochastic optimization formalisms have been used with great success in solving problems involving very large search spaces. DEs were originally developed as genetic engineering models mimicking population evolution in natural systems. Specifically, DEs like genetic algorithms (GA) enforce the "survival-of-the-fittest" and "genetic propagation of characteristics" principles of biological evo-

lution for searching for the solution space of an optimization problem. DE has been used to design several complex digital filters [12] and to design fuzzy logic controllers [13]. DE can also be used for parameter estimations, *e.g.* Babu and Sastry [14], used DE for the estimation of effective heat transfer parameters in trickle-bed reactors using radial temperature profile measurements. DE is also used to tune the ANN parameters and their application in slurry flow as reported in literature [5-8]. It was concluded that DE takes less computational time to converge compared to the existing techniques without compromising with the accuracy of the parameter estimates.

In this paper, we present a hybrid ANN-DE approach, which not only relieves the user from choosing these meta-parameters but also finds the optimum values of these parameters to minimize the generalization error. The strategy (henceforth referred to as "ANN-DE") uses an ANN as the nonlinear process modeling paradigm, and the DE for optimizing the meta-parameters of the ANN model such that an improved prediction performance is realized. In the present work, we illustrate the ANN-DE approach by applying it for identification of different regimes of solid liquid flow.

ARTIFICIAL NEURAL NETWORK (ANN) MODELING

Neural networks are computer algorithms inspired by the way information is processed in the nervous system.

Network architecture

The MLP network [5-8] usually consists of three layers of nodes. The layers described as input, hidden and output layers, comprise N, L and K number of processing nodes, respectively. Each node in the input (hidden) layer is linked to all the nodes in the hidden (output) layer using weighted connections. In addition to the N and L number of input and hidden nodes, the MLP architecture also houses a bias node (with fixed output of +1) in its input and hidden layers; the bias nodes are also connected to all the nodes in the subsequent layer and they provide additional adjustable parameters (weights) for the model fitting. The number of nodes (N) in the MLP network's input layer is equal to the number of inputs in the process whereas the number of output nodes (K) equals the number of process outputs. However, the number of hidden nodes (L) is an adjustable parameter whose magnitude is determined by issues, such as the desired approximation and generalization capabilities of the network model.

Back propagation algorithm (BPA)

The back propagation algorithm modifies network weights to minimize the mean squared error between the desired and the actual outputs of the network. Back propagation uses supervised learning in which the network is trained using data for which input as well as desired outputs are known. Once trained, the network weights are frozen and can be used to compute output values for new input samples.

A typical back propagation algorithm can be given as follows [5-8]. The MLP network is a nonlinear function-mapping device that determines the K dimensional nonlinear function vector f, where $f: X \rightarrow Y$. Here, X is a set of N-dimensional input vectors ($X = \{x_p\}$; $p = 1,2,...,P$ and $x = (x_1,x_2,...,x_n,...,x_N)^T$), and Y is the set of corresponding K-dimensional output vectors ($Y = \{y_p\}$; $p = 1,2,...,P$ and $y = (y_1,y_2,...,y_k,...,y_K)^T$). The precise form of f is determined by: *i*) network topology, *ii*) choice of the activation function used for computing outputs of the hidden and output nodes, and *iii*) network weight matrices $\mathbf{W}^H$ and $\mathbf{W}^O$ (they refer to the weights between input and hidden nodes, and hidden and output nodes, respectively). Thus, the nonlinear mapping can be expressed as:

$$y = y(x,\mathbf{W}) \tag{1}$$

where $\mathbf{W} = \{\mathbf{W}^H,\mathbf{W}^O\}$. This equation suggests that y is a function of x, which is parameterized by $\mathbf{W}$ (Bishop21). It is now possible to write the closed-form expression of the input-output relationship approximated by the three-layered MLP as:

$$y_k = \tilde{f}_1\left(\sum_{l=0}^{L} \mathbf{W}_{lk}^O \tilde{f}_2\left(\sum_{n=0}^{N} \mathbf{W}_{nl}^H x_n\right)\right),\ k = 1,2,...,K \tag{2}$$

where y_k refers to the k^{th} network output; $\tilde{f}_1$ and $\tilde{f}_2$ denote the nonlinear activation functions; $\mathbf{W}_{lk}^O$ refers to the weight between l^{th} hidden node and k^{th} output node; $\mathbf{W}_{nl}^H$ is the weight between n^{th} input and l^{th} hidden node, and xn represents the n^{th} network input.

Note that in Eq. (2) the bias node is indexed as the zero node in the respective layer. In order that an MLP network approximates the nonlinear relationship existing between the process inputs and the outputs, it needs to be trained in a manner such that a prespecified error function is minimized. Essentially, the MLP-training procedure aims at obtaining an optimal set ($\mathbf{W}$) of the network weight matrices $\mathbf{W}^H$ and $\mathbf{W}^O$, which minimize an error function. The commonly employed error function is the average absolute relative error (*AARE*) defined as:

$$AARE = \frac{1}{N}\sum_{1}^{N}\left|\frac{y_{\text{Predicted}} - y_{\text{Experimental}}}{y_{\text{Experimental}}}\right| \tag{3}$$

The most widely used formalism for the *AARE* minimization is the error-back-propagation (EBP) algorithm utilizing a gradient-descent technique known as the generalized delta rule (GDR). In the EBP methodology, the weight matrix set, W, is initially randomized. Thereafter, an input vector from the training set is applied to the network's input nodes and the outputs of the hidden and output nodes are computed. The outputs are computed as follows. First the weighted-sum of all the node-specific inputs is evaluated, which is then transformed using a nonlinear activation function, such as the logistic sigmoid. The outputs from the output nodes are then compared with their target values and the difference is used to compute the *AARE* defined in Eq. (3). Upon *AARE* computation, the weight matrices $\mathbf{W}^H$ and $\mathbf{W}^O$ are updated using the GDR framework. This procedure, when repeated with the remaining input patterns in the training set, completes one network training iteration. For the *AARE* minimization, several training iterations are usually necessary.

Tuning parameters of ANN

It is well known that ANN generalization performance (estimation accuracy) depends on a good setting of meta-parameters listed below.

1. Number of nodes in hidden layer: the number of nodes in hidden layer has a profound effect on ANN performance. Too few nodes could not learn the relationship in data properly and too large number of nodes increases the network complexity and execution time. From literature it is found that the optimal number of nodes in hidden layer normally calculated by trial and error method. Such an approach apart from consuming enormous time may not really obtain the best possible performance.

2. The activation functions in input layer: each hidden node and output node applies the activation function to its net input. The five types of activation functions reported in literature and used in this work are shown in Table 1.

There is no consensus in the literature on which type of activation function is to be used and it depends on the type of input training data and the case under investigation. For new users it is difficult to choose the activation function for their data as they have no guidelines to choose. Multilayer networks typically use sigmoid transfer functions in the hidden layers. Sigmoid functions are characterized by the fact that their slope must approach zero, as the input gets large. This causes a problem when using the steepest descent to train a multilayer network with sigmoid functions, since the gradient can have a very small magnitude; and therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

3. The activation function of output layer: the same remarks for input activation are applicable.

4. The learning rate: the performance of the back propagation algorithm can be improved if we estimate the optimal learning rate. For a new user choosing the optimal learning rate is very difficult. The learning rate is multiplied with the negative of the gradient to determine the changes to the weights and biases. The larger the learning rate, the bigger the step. If the learning rate is made too large, the algorithm becomes unstable. If the learning rate is set too small, the algorithm takes a long time to converge.

Apart from the above 4 parameters, the ANN performance also depends upon the training algorithm used for back propagation. Over the years different researchers have developed many ANN training algorithms to reduce execution time and computer storage requirements. There are several different back propagation training algorithms published in literature [7]. Figure 1 shows some of those algorithms used in the present study. They have a variety of different computation and storage requirements, and no one algorithm is best suited to all locations. The basic differences between these algorithms are how they handle the weight up-gradation in Eqs. (1) and (2) to reduce error and how they modify learning rate (η) to reduce convergence time.

Table 1. Different activation function

Case	Name of activation function	Equation[a]
1	Log sigmoid function (logsig)	$Y_i = \frac{1}{1+e^{-net_i}}$
2	Tan hyperbolic function (tansig)	$Y_i = \tanh(net_i)$
3	Linear function (purelin)	$Y_i = net_i$
4	Radial basis function (radbas)	$Y_i = -net_i^2$
5	Triangular basis function (tribas)	$Y_i = \begin{cases} 1\text{-abs}(net_i), & -1 \le net_i \le 1 \\ 0, & \text{Otherwise} \end{cases}$

[a] Y_i is the output from node i, net_i is the input to the node i, $net_i = \sum w_i x_i$

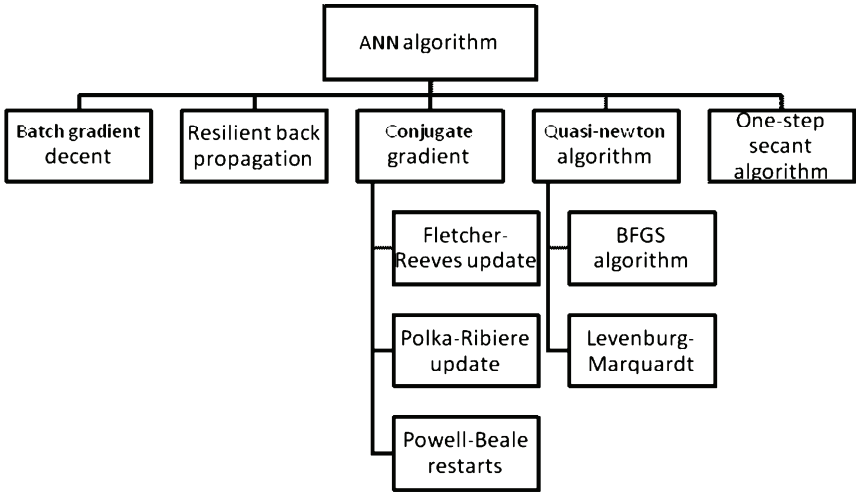

Figure 1. Different ANN algorithms published in various literatures.

Most of the available ANN software requires the above four parameters as user inputs or calculates the above parameters on trial and error basis. A long execution time is needed to explore all the combinations of the above parameters to really find the best possible solutions. In the present paper we use the differential evolution technique to find out the optimum solution.

Differential evolution (DE): at a glance

Having developed an ANN-based process model, a DE algorithm is used to optimize the *N*- dimensional input space of the ANN model. Conventionally, various deterministic gradient-based methods are used for performing optimization of the phenomenological models. Most of these methods require that the objective function should simultaneously satisfy the smoothness, continuity, and differentiability criteria. Although the nonlinear relationships approximated by an ANN model can be expressed in the form of generic closed-form expressions, the objective function(s) derived thereby cannot be guaranteed to satisfy the smoothness criteria. Thus, the gradient-based methods cannot be efficiently used for optimizing the input space of an ANN model and, therefore, it becomes necessary to explore alternative optimization formalisms, which are lenient towards the form of the objective function.

In the recent years, differential evolutions that are members of the stochastic optimization formalisms have been used with great success in solving problems involving very large search spaces [12,15–17]. The DEs were originally developed as genetic engineering models mimicking population evolution in natural systems. Specifically, DE like genetic algorithm (GA) enforces the "survival-of-the-fittest" and "genetic propagation of characteristics" principles of biological evolution for searching the solution space of an optimization problem. The principal features possessed by the DEs are: *i*) they require only scalar values and not the second- and/or first-order derivatives of the objective function, *ii*) the capability to handle nonlinear and noisy objective functions, *iii*) they perform global search and thus are more likely to arrive at or near the global optimum and *iv*) DEs do not impose pre-conditions, such as smoothness, differentiability and continuity, on the form of the objective function [12,14].

Differential evolution, an improved version of GA, is an exceptionally simple evolution strategy that is significantly faster and robust at numerical optimization and is more likely to find a function's true global optimum. Unlike simple GA that uses a binary coding for representing problem parameters, DE uses real coding of floating point numbers. The mutation operator here is addition instead of bit-wise flipping used in GA. And DE uses non-uniform crossover and tournament selection operators to create new solution strings. Among the DEs advantages are its simple structure, ease of use, speed and robustness. It can be used for optimizing functions with real variables and many local optima.

This paper demonstrates the successful application of Differential Evolution to the practical optimization problem. As already stated, DE in principle is similar to GA. So, as in GA, we use a population of points in our search for the optimum. The population size is denoted by NP. The dimension of each vector is denoted by *D.* The main operation is the NP number of competitions that are to be carried out to decide the next generation.

To start with, we have a population of NP vectors within the range of the objective function. We

select one of these NP vectors as our target vector. We then randomly select two vectors from the population and find the difference between them (vector subtraction). This difference is multiplied by a factor F (specified at the start) and added to third randomly selected vector. The result is called the noisy random vector. Subsequently, crossover is performed between the target vector and noisy random vector to produce the trial vector. Then, a competition between the trial vector and target vector is performed and the winner is replaced into the population. The same procedure is carried out NP times to decide the next generation of vectors. This sequence is continued until some convergence criterion is met. This summarizes the basic procedure carried out in differential evolution. The details of this procedure are described in Appendix 1.

DE-Based optimization of ANN models

There are different measures by which ANN performance is assessed, validation and leave-one-out error estimates being the most commonly used ones. Here we divide the total available data as training data (75% of data) and test data (25% data chosen randomly). While ANN algorithm was trained on training data but the ANN performance is estimated on test data.

The statistical analysis of ANN prediction is based on the following performance criteria:

1. The average absolute relative error (*AARE*, Eq. (3)) on test set should be minimum.

2. The standard deviation of error (σ) on test data should be minimum:

$$\sigma = \sqrt{\frac{1}{N-1}\left(\left|\frac{y_{\text{Predicted},i} - y_{\text{Experimental},i}}{y_{\text{Experimental},i}}\right| - AARE\right)^2}$$

3. The cross-correlation co-efficient (R) between input and output should be around unity:

$$R = \frac{\sum_{i=1}^{N}(y_{\text{Experimental},i} - \bar{y}_{\text{Experimental}})(y_{\text{Predicted},i} - \bar{y}_{\text{Predicted}})}{\sqrt{\sum_{i=1}^{N}(y_{\text{Experimental},i} - \bar{y}_{\text{Experimental}})^2}\sqrt{\sum_{i=1}^{N}(y_{\text{Predicted},i} - \bar{y}_{\text{Predicted}})^2}}$$

ANN learning is considered successful only if the system can perform well on test data on which the system has not been trained. The above five parameters of ANN are optimized by DE algorithm stated below.

The objective function and the optimal problem of ANN model of the present study are represented as:

Minimize *AARE*(**X**) on test set
X ∈ {x1, x2, x3, x4, x5}
where
x1 = Number of nodes in hidden layer {1,2,…,100}
x2 = Input layer activation function {1,2,3,4,5, corresponds to five activation function in Table 1}
x3 = Output layer activation function {1,2,3,4,5, corresponds to five activation function in Table 1}
x4 = Learning rate {0 to 5}
x5 = Training algorithm {1,2,...,8, corresponds to eight training algorithm as per Figure 1}

The objective function is minimization of average absolute relative error (*AARE*) on test set and **X** is a solution string representing a design configuration of ANN architecture. The design variable x1 takes any integer values for number of nodes in the range of 1 to 100, x2 represents the input layer activation function taking any values in the range of 1 to 5 corresponds to five activation function in Table 1. x3 Represents the output layer activation function taking any values in the range of 1 to 5 corresponds to five activation function in table 1. x4 Represents learning rates and can take any value between 0 and 5. The variable x5 takes eight values of the training algorithm which corresponds to eight ANN training algorithm in Figure 1.

The total number of design combinations with these variables is 100×5×5×5×8 = 100,000. This means that if an exhaustive search is to be performed it will take a maximum 1000,000 function evaluations before arriving at the global minimum *AARE* for the test set (assuming 5 trials for to arrive optimum learning rate). So the strategy which takes fewer function evaluations is the best one. Considering minimization of *AARE* as the objective function, differential evolution technique is applied to find the optimum design configuration of ANN model. The methodology adopted is shown in Figure 2.

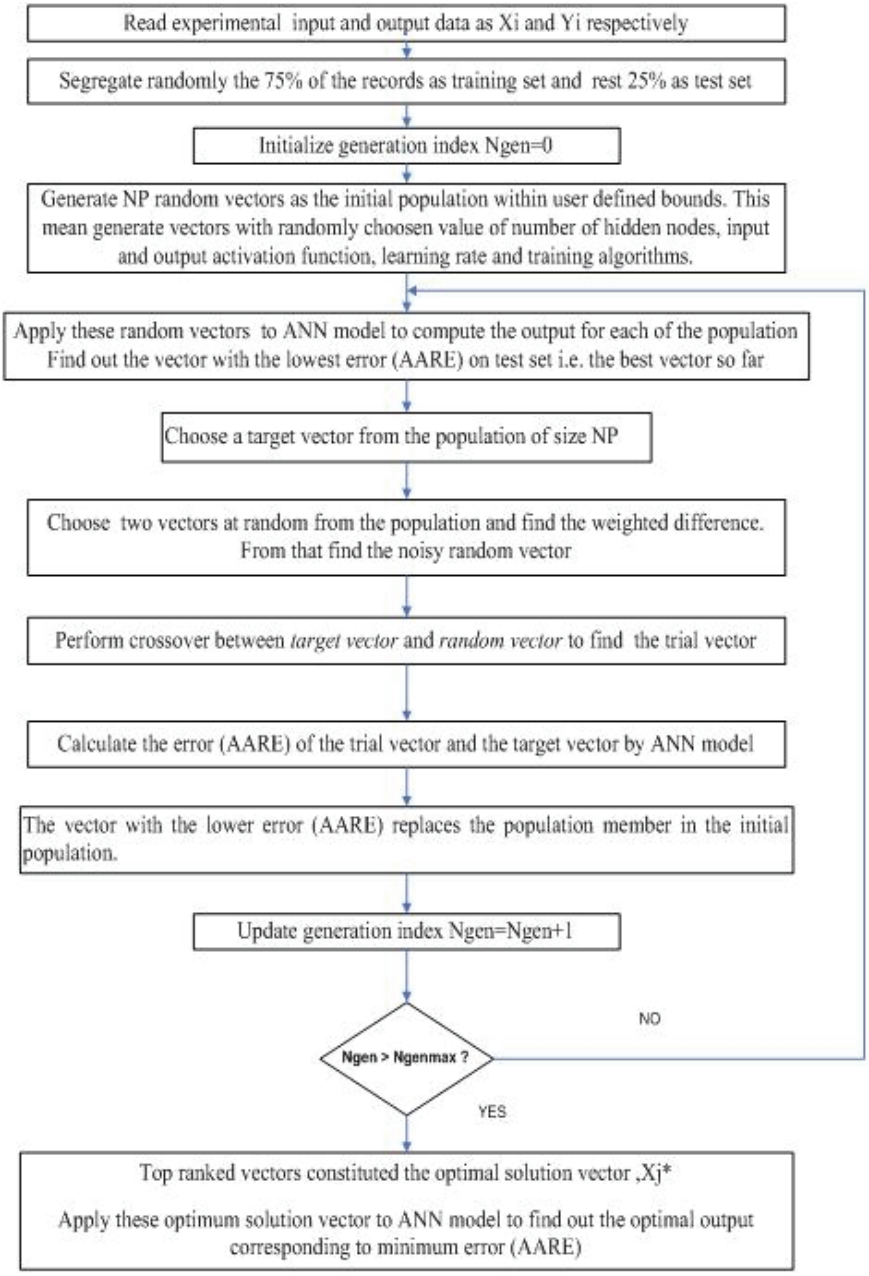

Figure 2. Schematic for hybrid ANN-DE algorithm implementation.

CASE STUDY: IDENTIFICATION OF DIFFERENT REGIMES IN SOLID-LIQUID SLURRY FLOW

The hybrid ANN-DE algorithm has been applied for prediction of different regimes of solid liquid slurry flow. The regime identification case study was chosen as phenomenological model is not available for it and it has a practical importance to reduce power consumption in slurry transport in pipeline.

Different regimes in slurry flow

To understand the phenomena, it is very important to know the different flow regimes of slurry flow. There are four main flow regimes in a horizontal pipeline flow (Figure 3) [18]. These are:

1. Flow with a stationary bed;
2. Flow with a moving bed and saltation (with or without suspension);
3. Heterogeneous mixture with all solids in suspension;
4. Pseudo homogeneous or homogeneous mixtures with all solids in suspension.

The tendency that the solid particles have to settle under the influence of gravity has a significant effect on the behavior of slurry that is transported in a horizontal pipeline. The settling tendency leads to a

significant gradation in the concentration of solids in the slurry. The concentration of solids is higher in the lower sections of the horizontal pipe. The extent of the accumulation of solids in the lower section depends strongly on the velocity of the slurry in the pipeline. The higher the velocity, the higher the turbulence level is and the greater the ability of the carrier fluid to keep the particles in suspension is. It is the upward motion of eddy currents transverse to the main direction of flow of the slurry that is responsible for maintaining the particles in suspension. At very high turbulence levels the suspension is almost homogeneous with very good dispersion of the solids while at low turbulence levels the particles settle towards the floor of the channel and can in fact remain in contact with the flow and are transported as a sliding bed under the influence of the pressure gradient in the fluid. Between these two extremes of behavior, two other more or less clearly defined flow regimes can be identified. When the turbulence level is not high to prevent any deposition of particles on the floor of the channel, the flow regime is described as being heterogeneous suspension. As the velocity of the slurry is reduced further a distinct mode of transport known as saltation develops. In the saltation regimes, there is a visible layer of particles on the floor of the channel and these are being continually picked up by turbulent eddies and dropped to the floor again further down the pipeline.

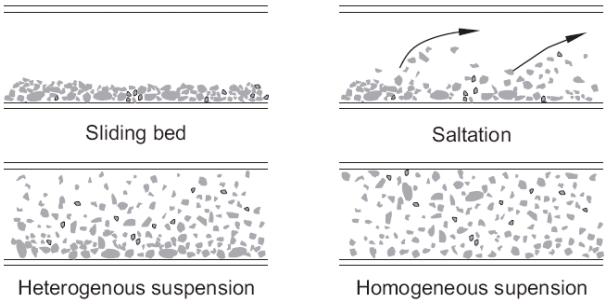

Figure 3. Four regimes of flow of settling slurries in horizontal pipeline.

The solids therefore spend some of their time on the floor and the rest in suspension in the flowing fluid. Under saltation conditions the concentration of solids is strongly non uniform. The four regimes of flow are illustrated in Figure 3.

The four regimes of the flow described above can be represented by a plot of the pressure gradient versus the average speed of the mixture (Figure 4).

The transitional velocities are defined as:

- V_1: velocity at or above which the bed in the lower half of the pipe is stationary. In the upper half of the pipe, some solids may move by saltation or suspension;
- V_2: velocity at or above which the mixture flows as an asymmetric mixture with the coarser particles forming a moving bed;

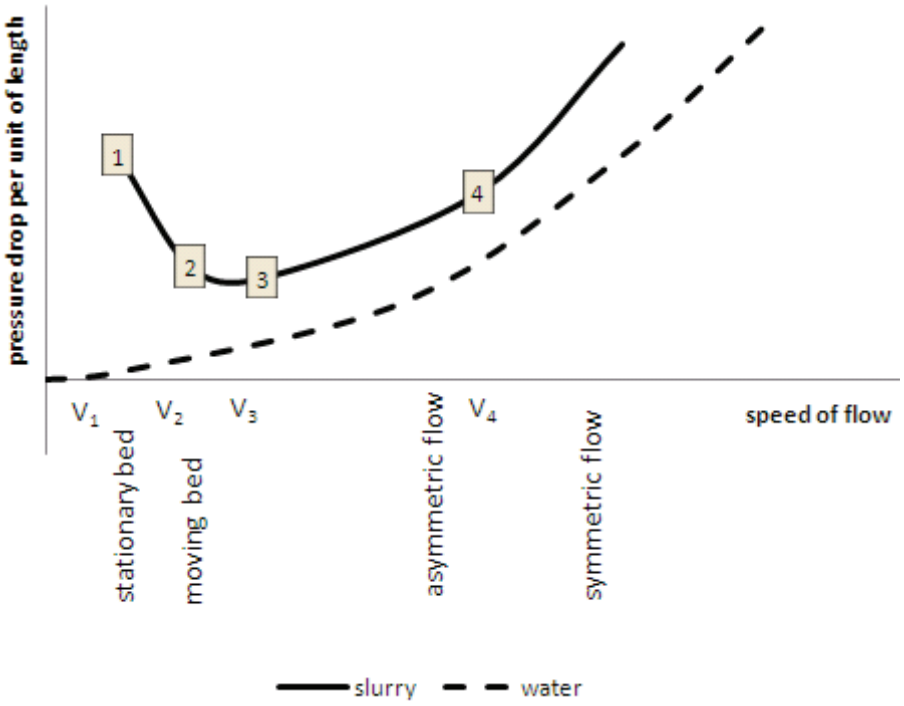

Figure 4. Plot of transitional mixture velocity with pressure drop.

- V_3 or V_c: velocity at or above which all particles move as an asymmetric suspension and below which the solids start to settle and form a moving bed;

- V_4: velocity at or above which all solids move as a symmetric suspension.

Head loss correlations for separate flow regimes

The saltation and heterogeneous suspension regimes have been studied most widely and the best known correlation for the excess pressure gradient due to the presence of solid particles in the slurry is due to Durand, Condolios and Worster [19] which is given by:

$$\phi = \frac{\Delta p_{\mathrm{f,sl}} - \Delta p_{\mathrm{fw}}}{\Delta p_{\mathrm{fw}}} = \Omega C(\sqrt{C_{\mathrm{D}}}Fr)^{-1.5} \tag{4}$$

where Ω is a constant, C is the volumetric fraction of solids in the suspension, C_{D} is the drag coefficient at terminal settling velocity and Fr is the Froude number. The value to be used for the constant Ω is uncertain and values between 65 and 150 are reported in the literature. Because this correlation does not apply to all regimes of flow, the experimental data cannot be used to fix the value more precisely. Errors of 100 percent and more in the calculated value of ϕ can result [18].

While the Durand-Condolios-Worster correlation is useful in the heterogeneous suspension flow regime, it deviates more and more from actual conditions in the other regimes flow. Experimental observations have shown that different correlations should be used in each of the identifiable flow regimes. Using the experimental data, Turian and Yuan established that the excess pressure gradient in each flow regime can be correlated using an equation of the form:

$$f_{sl} - f_w = KC^{\alpha} f_{\mathrm{w}}^{\beta} C_{\mathrm{D}}^{*\gamma} Fr^{\delta} \tag{5}$$

where the coefficients K, α, β, γ and δ have values that are specific to each flow regime and C_{D}^{*} is the drag coefficient at the terminal settling velocity of slurry. Using experimental data gathered from experiments in each flow regime, the best available values of these parameters in each flow regime are given for:

- sliding bed (regime 0)

$$f_{sl} - f_w = 12.13C^{0.7389} f_{\mathrm{w}}^{0.7717} C_{\mathrm{D}}^{*-0.4213} Fr^{-1.096} \tag{6}$$

- saltation (regime 1)

$$f_{sl} - f_w = 107.1C^{1.018} f_{\mathrm{w}}^{1.046} C_{\mathrm{D}}^{*-0.4213} Fr^{-1.354} \tag{7}$$

- heterogeneous suspension (regime 2)

$$f_{sl} - f_w = 30.11C^{0.868} f_{\mathrm{w}}^{1.2} C_{\mathrm{D}}^{*-0.1677} Fr^{-0.6938} \tag{8}$$

- homogeneous suspension (regime 3)

$$f_{sl} - f_w = 8.538C^{0.5024} f_{\mathrm{w}}^{1.428} C_{\mathrm{D}}^{*-0.1516} Fr^{-0.3531} \tag{9}$$

A fairly consistent trend in the variation of the correlating parameters is seen in the four correlations. Here, C_{D}^{*} is drag co-efficient at the terminal settling velocity of slurry.

Flow regime boundaries (Turian and Yuan's approach)

The boundaries of the flow regimes are defined in a self-consistent manner by noting that any two regimes are contiguous at their common boundary and therefore each of the two correlation equations must be satisfied simultaneously [18]. For example, the boundary between the sliding bed regime (Regime 0) and the saltation regime (regime 1) must lie along the solution locus of the equation:

$$\begin{aligned} &12.13C^{0.7389} f_{\mathrm{w}}^{0.7717} C_{\mathrm{D}}^{*-0.4213} Fr^{-1.096} = \\ &= 107.1C^{1.018} f_{\mathrm{w}}^{1.046} C_{\mathrm{D}}^{*-0.4213} Fr^{-1.354} \end{aligned} \tag{10}$$

This is simplified to:

$$Fr = \frac{\bar{V}^2}{Dg(s-1)} = 4679C^{1.083} f_{\mathrm{w}}^{1.064} C_{\mathrm{D}}^{*-0.0616} \tag{11}$$

The regime transition number for transitions between regime 0 and regime 1 is defined by:

$$R01 = \frac{Fr}{4679C^{1.083} f_{\mathrm{w}}^{1.064} C_{\mathrm{D}}^{*-0.0616}} \tag{12}$$

and this number must be unity on the boundary between these two regimes.

The transition numbers for the other possible transitions are found in the same way and are given by:

$$R02 = \frac{Fr}{0.1044C^{-0.3255} f_{\mathrm{w}}^{-1.065} C_{\mathrm{D}}^{*-0.5906}} \tag{13}$$

$$R12 = \frac{Fr}{6.8359C^{0.2263} f_{\mathrm{w}}^{-0.2334} C_{\mathrm{D}}^{*-0.3840}} \tag{14}$$

$$R13 = \frac{Fr}{12.522C^{0.5153} f_{\mathrm{w}}^{-0.3820} C_{\mathrm{D}}^{*-0.5724}} \tag{15}$$

$$R23 = \frac{Fr}{40.38C^{1.075} f_{\mathrm{w}}^{-0.6700} C_{\mathrm{D}}^{*-0.9375}} \tag{16}$$

$$R03 = \frac{Fr}{1.6038C^{0.3138} f_{\mathrm{w}}^{-0.8837} C_{\mathrm{D}}^{*-0.7496}} \tag{17}$$

These numbers define the boundaries between any two flow regimes a and b by the condition $Rab = 1$. It is possible to identify the regime that applies to a

particular set of physical conditions quite simply from knowledge of the transition numbers *Rab*. If $a < b$ the value of *Rab* increase monotonically as the velocity increases. At low velocities, $Rab < 1$ and with increasing velocity, the value of *Rab* will eventually pass through the value 1.0. This must signal a transition out of regime *a*. The following simple rules will fix the flow regime:

If $Rab < 1$ the regime is not b.

If $Rab > 1$ the regime is not a.

This inequalities must be tested for the combinations of *a* and *b*. No more than three of the transition numbers need to be calculated to fix the flow regime uniquely. Notice that these rules test the flow regimes negatively and a single test will never suffice to define the flow regime. It is always necessary to test at least three different combinations of *a* and *b* to get a definitive identification of the flow regime (Figure 5). The applicable flow regime can be identified quickly and easily using the figure and the appropriate equation can be selected from equations above to calculate the slurry friction factor.

Performance check of Turian Yuan's approach

Data Collection. As mentioned earlier, over the years researchers have amply quantified the flow regime of slurry flow in pipeline. In this work, about 800 experimental points have been collected from 20 sources from open literature spanning the years 1950--2002 (most of them compiled in PhD thesis report of Hsu (1987) [20]). The data were screened for incompleteness, redundancies and evident inaccuracies. This wide range of database includes experimental information from mainly two regimes namely saltation regime (regime 1) and heterogeneous flow regime (regime 2). These two regimes were selected as they have great practical significance and all practical hydrotransport carried out in heterogeneous suspension region because of lowest pressure drop and subsequent less power requirement. Table 2 indicates the wide range of the collected databank for regime identification. Table 4 shows some of these data.

Table 2. System and parameter studied

Slurry system	Coal-water, coal-brine, ash-water, copper ore--water, sand-water, gypsum-water, glass-water, gravel-water, iron--water, iron-kerosene, high density material--water, iron tailings-water, limestone-water, limonite-water, plastic--water, potash-brine, sand-ethylene glycol, nickel shot-water, iron powder-water, ore-water
Pipe diameter, m	0.0127 – 0.80
Particle diameter, m	(0.0017 – 0.868) x 10^{-2}
Liquid density, kg/m^3	770 – 1350
Solids density, kg/m^3	1150 – 8900
Liquid viscosity, Pa s	0.008 -0.019
Solids concentration (volume fraction)	0.005 – 0.561
Velocity, m/s	0.18 - 4.56

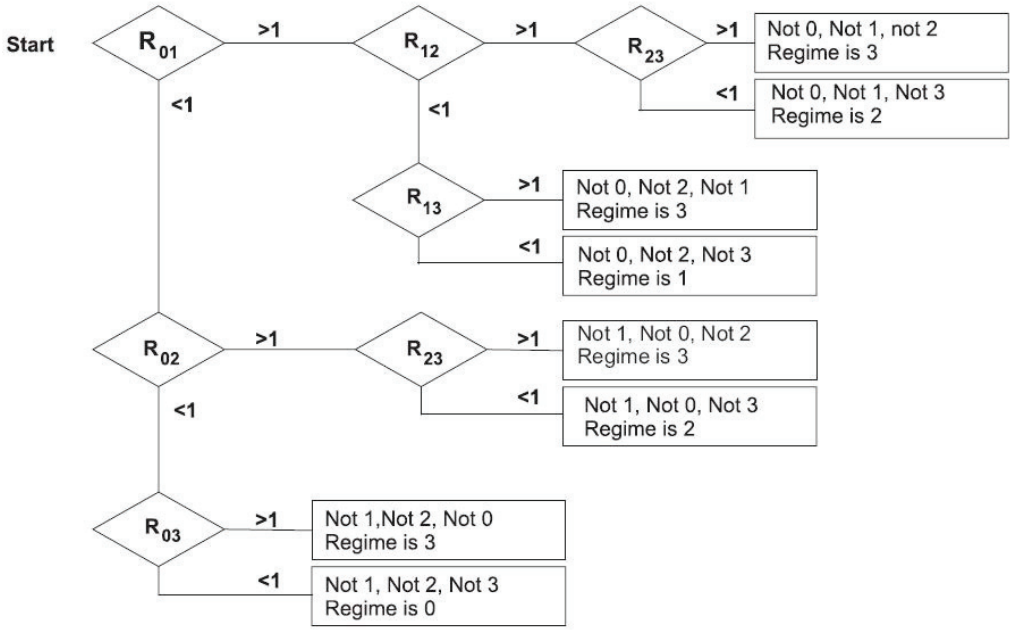

Figure 5. Decision tree for identification of regimes.

Regime identification. Now Table 2 data was exposed to Turian and Yuan's calculation and regimes were identified as shown in Table 3. It is evident from the last two columns of the Table 3 that the Turian Yuan's approach is not producing promising results and fails to correctly identify the regimes. These poor results motivated this work and an attempt has been made to explore the new ANN classification methodology to identify the flow regime correctly.

This paper presents a systematic approach using robust hybrid ANN-DE techniques to build a regime identification correlation from available experimental data. This correlation has been derived from a broad experimental data bank collected from the open literature (800 measurements covering a wide range of pipe dimensions, operating conditions and physical properties).

Development of the artificial neural network (ANN) based correlation

The development of the ANN-based correlation was started with the collection of a large databank. The next step was to perform an artificial neural network, and to validate it statistically.

After extensive literature survey all physical parameters that influence regimes identification are put in a so-called "wish-list". Based on the extensive literature survey, the input variables such as pipe diameter, particle diameter, solids concentration, solid and liquid density and viscosity and velocity of flowing medium have been finalized to predict different regimes in slurry pipeline. Some portion of the input and output data used for ANN classification is shown in Table 4.

RESULTS AND DISCUSSION

Prediction performance of hybrid ANN-DE model

The purpose of present study is to develop some simple methodology for flow regime classification so that users do not have to go through the rigorous calculation of Turian and Yuan's approach to detect the regime. The present study focuses only on classification and identification of two major regimes namely heterogeneous suspension and saltation regime. These two regimes are chosen purposefully as pressure drop is minimal in the heterogeneous regime and thus the most important from a power consumption point of view. All commercial slurry transport is in the heterogeneous region as it contributes lowest power cost per ton of slurry transported. Another reason to choose these two regimes for this study is the availability of a

Table 3. Performance of Turian Yuan's correlation to identify different flow regime

Sl. No.	*R*01	*R*02	*R*03	*R*12	*R*13	*R*23	Regime identified by Turian approach	Regime identified by experiments
1	11.89	0.03	0.06	0.34	0.24	0.12	1	2
2	0.66	1.75	0.55	1.19	0.58	0.14	2	2
3	0.13	3.16	2.25	0.91	1.08	1.51	3	2
4	0.75	9.82	7.38	3.60	4.09	5.25	3	2
5	4.30	0.04	0.02	0.25	0.08	0.01	1	1
6	1.30	0.01	0.01	0.09	0.04	0.01	1	1
7	0.32	0.01	0.00	0.03	0.01	0.00	0	1
8	8.05	0.01	0.01	0.11	0.07	0.03	1	1
9	3.10	0.03	0.02	0.18	0.08	0.02	1	2
10	4.07	0.01	0.02	0.07	0.07	0.06	1	2

Table 4. Typical input and output data for ANN training

Sl. No.	Input to ANN							Output regime
	Particle diameter cm	Solid concentration, vol. fraction	Solid density g/cm^3	Fluid density g/cm^3	Fluid viscosity Pa s	Pipe diameter cm	Fluid velocity cm/s	
1	0.013	0.028	1.834	0.998	0.00098	7.62	147.22	1
2	0.010	0.300	2.820	0.982	0.00130	5.00	332.50	-1
3	0.220	0.304	1.670	1.325	0.15200	7.62	85.98	-1
4	0.220	0.304	1.670	1.325	0.15200	10.16	204.14	-1
5	0.868	0.165	1.530	0.998	0.00098	20.80	254.85	1
6	0.183	0.056	1.530	0.998	0.00098	4.00	52.55	-1
7	0.030	0.170	3.360	0.998	0.00098	10.30	126.73	-1
8	0.014	0.022	2.690	0.998	0.00098	20.70	210.62	-1
9	0.009	0.125	3.000	0.998	0.00098	20.70	337.16	1
10	0.006	0.014	3.100	0.998	0.00098	14.90	120.59	1

large number of commercial and experimental data for these two regimes compared to other regimes. The method developed here can also be extended to classification of other regimes. Initially all the data related to these two regimes were collected from open literature. Six parameters were identified as input (Table 4) to ANN and the +1 or -1 is put as target. The output was designated as +1 for heterogeneous suspension flow regimes data and -1 for saltation regimes data.

As the magnitudes of inputs greatly differ from each other, they are normalized in -1 to +1 scale using following formula:

$$x_{\text{Normal}} = \frac{2(x - x_{\min})}{(x_{\max} - x_{\min})} - 1$$

75% Of total dataset was chosen randomly for training and the other 25% was selected for validation. These 25% validation set data were used to stop the iterations at appropriate time to avoid "over training" phenomena.

These data were then exposed to the hybrid ANN-DE model described above. The main advantage of the hybrid model is that no other user inputs are required to set the optimum training architecture. The ANN-DE model optimizes the five parameters of ANN architecture (namely the number of nodes in hidden layer, the activation functions in input and output layer, the learning rate and ANN training algorithm) and selects on its own the best architecture and training algorithm for the present data. The model output is shown in Figure 6 and prediction error is summarized in Table 5. Figure 6 shows the hybrid system performance to classify the regime on 200 test data. Other 600 results of training data are not shown as they were used in training and expected to give accurate results. How accurately the hybrid system classified the test data is actually the real performance test.

Table 5. Prediction error by hybrid ANN-DE based model

Parameter	Training	Testing
AARE	0.000315	0.000320
Sigma	0.000833	0.000853
R	0.9999	0.9999
Optimum number of nodes	7	-
Input activation function	Log sigmoid function	-
Output activation function	Linear	-
Optimum learning rate	0.042	-
Best training algorithm	BFGS	-

Out of all the possibilities, the BFGS algorithm with seven number of nodes in hidden layer and log sigmoid and linear function in input and output layer has emerged out as the best solution (with lowest AME) for the present case. The other performance parameters namely execution time and storage requirement are neglected, as they are not important for this type of study. The low prediction error (*AARE* 0.03%) may be considered as an excellent prediction performance considering the poor understanding of slurry flow phenomena and large databank for training comprising various systems.

Recall that all the 800 experimental data collected from open literature was exposed earlier to Tu-

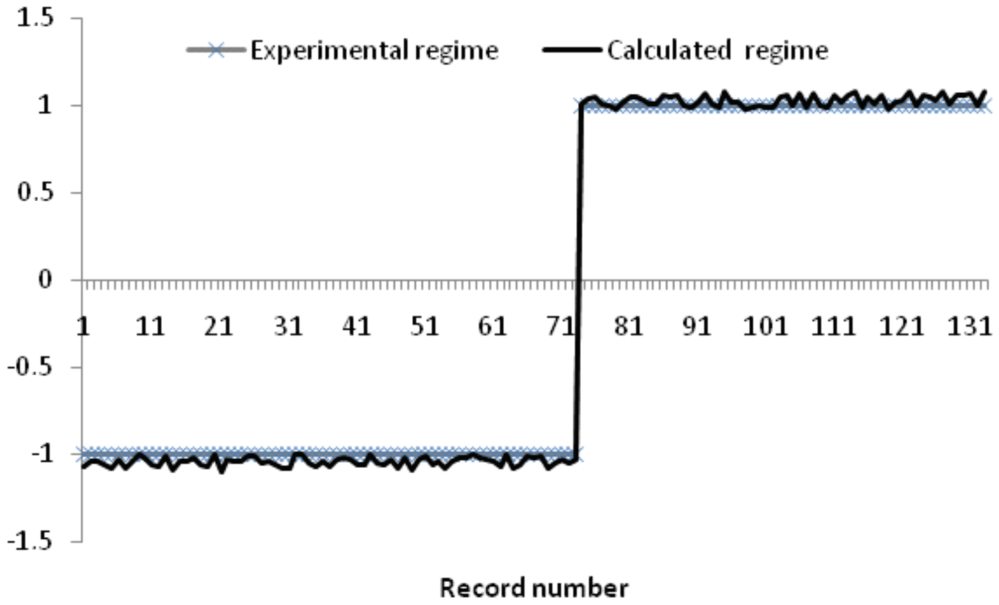

Figure 6. Experimental vs. predicted flow regime.

Table 6. Set of equations and fitting parameters for neural network correlations (i = 7, j = 7, k = 1); $H_j = 1/(1 + e^{-\Sigma w_{i,j} U_i})$, regime = $\Sigma w_{j,k} H_j$, where U_i is input variable (Table 4), H_j is hidden layer input and $U_8 = 1$, $H_8 = 1$ (bias)

w	1	2	3	4	5	6	7	8
$w_{i,j}$	43.47	-16.05	-31.47	-40.86	5.44	27.58	36.88	-14.62
	14.55	-12.56	2.49	-2.19	12.83	2.87	-9.78	-15.26
	1.29	2.33	5.83	0.69	3.74	1.37	-2.18	-9.24
	51.46	10.79	-61.61	-53.89	-19.25	47.63	-9.63	-23.57
	13.15	0.84	4.21	-18.66	14.74	3.10	-8.05	-14.23
	6.02	4.06	4.66	-2.98	-1.97	8.30	-1.20	-13.60
	-3.77	5.31	-5.08	0.63	-4.99	-2.04	2.13	10.77
$w_{j,k}$	-0.0006	18.75	10.94	1.99	-11.17	15.72	10.19	-11.19

rian Yuan's formulas for regime identification and *AARE* calculated was 25%. The present work has reduced the misclassification error from 25% to 0.03%. The advantage of the present method is that the user doesn't have to calculate the C_D, settling velocity, Froude number, *R*01, *R*02, *R*03, *R*12, *R*13, *R*23, etc. as in the case of Turian Yuan's approach to evaluate the regime. In the present approach, the regime will be evaluated effortlessly and more accurately from the basic slurry flow data (pipe diameter, solid concentration, fluid velocity, etc.) and the user will be relieved to calculate others parameters stated above. Once the regime has been evaluated correctly, appropriate pressure drop correlations (Eqs. (2)-(5)) can be used for the each regime. This will help to choose the appropriate correlations for pressure drop and to accurately predict the pressure drop in the design phase.

The weight, bias and final equations for calculating the regime for any solid-liquid slurry flow in pipelines are summarized in Table 6. With the help of the equation at the bottom of Table 6, any user can easily identify whether their slurry flow regime is in heterogeneous or moving bed regime. If the user found their flow to be in a moving bed regime they can increase the velocity of slurry flow to make it in a heterogeneous regime. This will ensure the lowest pressure drop and power consumption.

Comparison of hybrid ANN-DE model with ANN model

In a separate study, we exposed the same dataset to ANN algorithm only (without the DE algorithm) and tried to optimize the different parameters based on exhaustive search. We found that it was not possible to reach the best solutions starting from arbitrary initial conditions. The optimum choice of learning rate is especially very difficult to reach after starting with some discrete value. Many times the solutions got stuck up in sub-optimal local minima. These experiments justified the use of a hybrid technique for ANN parameter tuning. The best prediction after the exhaustive search along with ANN parameters was summarized in Table 7. From the Table 7, it is clear that even after 100,000 runs the ANN algorithm is unable to locate the global minima and the time of execution is 4 h with a Pentium 4 processor. On the other hand, the hybrid ANN-DE technique is able to locate the global minima with 2000 runs within 0.5 h. The prediction accuracy is also much better. Moreover it relieves the non expert users to choose the different parameters and find an optimum ANN meta-parameters with a good accuracy.

Table 7. Comparison of performance of ANN-DE hybrid model vs. ANN model

Parameter	Prediction performance by hybrid ANN-DE model	Prediction performance by ANN model only
AARE	0.000320	0.00451
Sigma	0.000853	0.0022
R	0.9999	0.98
Execution time, h	0.5	4.3

CONCLUSION

ANN regression methodology with a robust parameter tuning procedure has been described in this work which can be used effortlessly where phenolmenological model is difficult to develop. The method employs a hybrid ANN-DE approach for minimizing the generalization error. Superior prediction performances were obtained for the case study of regime identification and a comparison with selected correlations in the literature showed that the developed ANN correlation noticeably improved prediction of slurry flow regime over a wide range of operating conditions, physical properties, and pipe diameters. The proposed hybrid technique (ANN-DE) also relieves the non-expert users to choose the meta-parameters of ANN algorithm for his case study and find out optimum

value of these meta-parameters on its own. The results indicate that the ANN based technique with the DE based parameters tuning approach described in this work can yield excellent generalization and can be advantageously employed for a large class of regression problems encountered in process engineering.

Nomenclature

C Volume fraction of solids in the suspension (-)
C_D* Drag coefficient at terminal settling velocity (-)
D Pipe diameter (m)
d_p Particle diameter (m)
Fr Froude number (-)
f_{sl} Friction factor for slurry (-)
f_w Friction factor for carrier fluid (-)
K Constant (-)
R_{ep} Particle Reynolds number (-)
V_T Terminal settling velocity (m/s)

Greek symbols

Ω Constant used in Durand Condolios equation (-)
α Constant (-)
β Constant (-)
γ Constant (-)
δ Constant (-)
μ_s Viscosity of slurry (Pa s)
ρ_f Density of fluid (kg/m^3)
ρ_s Density of solid (kg/m^3)

Abbreviations

ANN Artificial neural network
DE Differential evolution

Appendix 1

Steps performed in DE

Assume that the objective function is of D dimensions and that it has to be optimized. The weighting constants F and the crossover constant C_R is specified.

Step 1. Generate N_P random vectors as the initial population: generate ($N_P \times D$) random numbers and liberalize the range between 0 and 1 to cover the entire range of the function. From these ($N_P \times D$) numbers, generate N_P random vectors, each of dimension D, by mapping the random numbers over the range of the function.

Step 2. Choose a target vector from the population of size N_P: first generate a random number between 0 and 1. The value of the random number decides which population member is to be selected as the target vector (X_i) (a linear mapping rule can be used).

Step 3. Choose two vectors at random from the population and find the weighted difference: generate two random numbers. Decide which two population members are to be selected (X_a,X_b). Find the vector difference between the two vectors (X_a - X_b). Multiply this difference by F to obtain the weighted difference.

Weighted difference = $F(X_a - X_b)$

Step 4. Find the noisy random vector: generate a random number. Choose a third random vector from the population (X_c). Add this vector to the weighted difference to obtain the noisy random vector (X'_c).

Step 5. Perform crossover between X_i and X'_c to find X_t, the trial vector: generate D random numbers. For each of the D dimensions, if the random number is greater than C_R, copy the value from X_i into the trial vector; if the random number is less than C_R, copy the value from X'_c into the trial vector.

Step 6. Calculate the cost of the trial vector and the target vector: for a minimization problem, calculate the function value directly and this is the cost. For a maximization problem, transform the objective function $f(x)$ using the rule $F(x) = 1/(1 + f(x))$ and calculate the value of the cost. Alternatively, directly calculate the value of f(x) and this yields the profit. In case cost is calculated, the vector that yields the lesser cost replaces the population member in the initial population. In case profit is calculated, the vector with the greater profit replaces the population member in the initial population.

Steps 1-6 are continued until some stopping criterion is met. This criterion may be one of two kinds. One may be some convergence criterion that states that the error in the minimum or maximum between two previous generations should be less than some specified value. The other may be an upper bound on the number of generations. The stopping criterion may be a combination of the two. Either way, once the stopping criterion is met, the computations are terminated.

Choosing DE key parameters N_P, F, and C_R is seldom difficult and some general guidelines are available. Normally, N_P ought to be about 5 to 10 times the number of parameters in a vector. As for F, it lies in the range 0.4 to 1.0. Initially, F = 0.5 can be tried then F and/or N_P is increased if the population converges prematurely. A good first choice for C_R is 0.1, but in general C_R should be as large a possible (Price and Storn, 1997).

REFERENCES

[1] R.M. Turian, T.F. Yuan, AIChE J. **23** (1977) 232-243
[2] A.B. Bulsari, J. Syst. Eng. **4** (1994) 131-170
[3] K. Huang, X.L. Zhan, F.Q. Chen, D.W. Lu, Chem. Eng. Sci. **58** (2003) 81-87

[4] L.B Jack, A.K. Nandi, Mech. Sys. Signal Proc. **16** (2002) 373-390

[5] S.K Lahiri, K.C. Ghanta, Chin. J. Chem. Eng. **16**(6) (2008) 50-58

[6] S.K Lahiri, K.C. Ghanta, Chem. Ind. Chem. Eng. Quar. **15**(2) (2009) 103-117

[7] S.K Lahiri, K.C. Ghanta, Chem. Eng. Sci. **63** (2008) 1497--1509

[8] S.K Lahiri, K.C. Ghanta, Chem. Prod. Proc. Model. **3**(1) (2008), Article 54

[9] G. Stephanopoulos, C. Han, Comp. Chem. Eng. **20** (1996) 743-791

[10] S.S. Tambe, B.D. Kulkarni, P.B. Deshpande, Simulations & Advanced Controls, Louisville, KY, 1996

[11] M. Riedmiller, H. Braun, Proc. of the IEEE Int. Conf. On Neural Networks, San Fransisco, CA, 1993

[12] K. Price, R. Storn, Dr. Dobb's J. (264) (1997) 18-24, 78

[13] K.K.N. Sastry, L. Behra, I. Nagrath, Fun. Inform. **37**(1-2) (1999) 121-136

[14] B.V. Babu, K.K.N Sastry, Comp. Chem. Eng. **23** (1999) 327-339

[15] S.Y. Shin, I.H. Lee, D. Kim, B.T. Zhang, IEEE **9**(2) (2005) 143-158

[16] J. Yaochu, J. Branke, IEEE **9**(3) (2005) 303-317

[17] Q. Zhang, J. Sun, E. Tsang, IEEE **9**(2) (2005) 192-200

[18] R.P. King, Introduction to practical fluid flow, Butterworth-Heinemann, Burlington, MA, 2002, pp. 59-67

[19] R. Durand , E. Condolios, Soc. Hydrotech. de France **1** (1952) 29-55

[20] F.L Hsu, PhD thesis, University of Illinois, Chicago, IL, 1987.

SANDIP KUMAR LAHIRI
K.C. GHANTA

Department of Chemical Engi¬neering, NIT, Durgapur, West Bengal, India

NAUČNI RAD

IDENTIFIKACIJA REŽIMA PROTICANJA SUSPENZIJA U CEVOVODIMA – NOVI PRISTUP MODELOVANJU POMOĆU VEŠTAČKIH NEURONSKIH MREŽA I DIFERENCIJALNOG PRIRASTA

U zavisnosti od prosečne brzine strujanja suspenzije utvrđeno je da se u cevovodima mogu ostvariti četiri različita režima proticanja (klizajući sloj, saltacija, heterogena suspenzija i homogena suspenzija). U literaturi je preporučeno nekoliko korelacija koje mogu poslužiti za identifikaciju ovih režima proticanja. Poznavanje režima proticanja je značajno za projektovanje cevovoda za suspenzije, jer oni određuju koja će korelacija za pad pritiska u cevovodu biti primenjena. Međutim, postojeće korelacije ne omogućavaju predviđanje režima proticanja u širokom opsegu operativnih uslova. U ovom radu je, na bazi oko 800 merenja iz literature, predložen metod identifikacije režima proticanja, korišćenjem modelovanja veštačkim neuronskim mrežama (ANN). Metod uključuje hibridnu veštačku neuronsku mrežu i tehniku diferncijalnog prirasta (ANN-DE), za efikasno podešavanje ANN meta parametara. Statistička analiza je pokazala da predloženi metod ima prosečnu grešku identifikacije od 0,03%. Poređenje sa izabranim korelacijama iz literature pokazalo je da predloženi ANN-DE metod značajno unapređuje predviđanje režima proticanja u širokom opsegu operativnih uslova, fizičkih osobina i prečnika cevovoda.

Ključne reči: neuronska mreža; diferencijalni prirast; režim toka guste suspenzije; tok guste suspenzije.