# Artificial neural network (ANN) modeling of dynamic effects on two-phase flow in homogenous porous media

Navraj S. Hanspal, Babatunde A. Allison, Lipika Deka and Diganta B. Das

## ABSTRACT

The dynamic effect in two-phase flow in porous media indicated by a dynamic coefficient $\tau$ depends on a number of factors (e.g. medium and fluid properties). Varying these parameters parametrically in mathematical models to compute $\tau$ incurs significant time and computational costs. To circumvent this issue, we present an artificial neural network (ANN)-based technique for predicting $\tau$ over a range of physical parameters of porous media and fluid that affect the flow. The data employed for training the ANN algorithm have been acquired from previous modeling studies. It is observed that ANN modeling can appropriately characterize the relationship between the changes in the media and fluid properties, thereby ensuring a reliable prediction of the dynamic coefficient as a function of water saturation. Our results indicate that a double-hidden-layer ANN network performs better in comparison to the single-hidden-layer ANN models for the majority of the performance tests carried out. While single-hidden-layer ANN models can reliably predict complex dynamic coefficients (e.g. water saturation relationships) at high water saturation content, the double-hidden-layer neural network model outperforms at low water saturation content. In all the cases, the single- and double-hidden-layer ANN models are better predictors in comparison to the regression models attempted in this work.

**Key words** | artificial neural network (ANN), dynamic coefficient, porous media, regression models, two-phase flow

**Navraj S. Hanspal**
School of Mechanical,
Aerospace and Civil Engineering (MACE),
University of Manchester,
Greater Manchester,
UK

**Navraj S. Hanspal**
Computational Engineering Group,
Dow Corning Corporation (HSC),
North Americas, Hemlock,
Michigan,
USA

**Babatunde A. Allison**
**Diganta B. Das** (corresponding author)
Chemical Engineering Department,
Loughborough University,
Loughborough LE11 3TU,
Leicestershire,
UK
E-mail: *D.B.Das@lboro.ac.uk*

**Lipika Deka**
Department of Computer Science,
Indian Institute of Technology,
Guwahati 110032, Assam,
India

## INTRODUCTION

Determining flow and transport behavior of non-aqueous phase liquids (NAPLs) (e.g. tetrachloroethene or PCE, polychlorinated biphenyl or PCB, trichloroethene or TCE, creosote, soltrol) is of enormous importance in solving many subsurface contamination problems. Characterization of the flow processes involving these chemicals depends upon the flow hydrodynamics (dynamic/static), capillary/viscous forces, mobility ratios, temperature, grain size distribution, fluid properties and length scales of observation. In general, modeling the two-phase flow processes requires the solution of equations for conservation of mass and momentum in conjunction with constitutive equations for capillary pressure $P_c$, saturation $S$ and relative permeability $K_r$.

An extended version of Darcy's law is most commonly used as the governing equation of motion for the fluid phases. The conservation of mass in the two-phase system is given by an equation for conservation of phase saturation, that is, the ratio of the volume of the fluid phase to the total pore volume in the domain. As the constitutive $P_c$–$S$ relationship, models such as the Brooks–Corey (Brooks & Corey 1964) or van Genuchten model (van Genuchten 1980) are frequently used. Similarly, other formulations such as the Brooks–Corey–Burdine formula (Brooks & Corey 1964) exists to calculate the $K_r$–$S$ relationship.

The current work is limited to the study of the $P_c$–$S$ relationship. Physically, this relationship represents curves which are determined by taking a porous medium sample

doi: 10.2166/hydro.2012.119

initially saturated with a wetting fluid (e.g. water), and then letting it gradually drain off by increasing the capillary pressure at the domain boundary and displacing the wetting fluid by a non-wetting fluid (e.g. air or oil). The main theoretical definition currently used to quantify the capillary pressure is an empirical relationship obtained under equilibrium conditions between individual phase pressures, defined:

$$P_{nw} - P_W = P_C(S_W) = f(S_W) \quad (1)$$

where $P_{nw}$ and $P_w$ are the average pressures of non-wetting and wetting phases, respectively, and $S_w$ is the wetting phase saturation.

A number of recent studies (Tsakiroglou *et al.* 2003; Das *et al.* 2006, 2007; Mirzaei & Das 2007; O'Carroll *et al.* 2010; Bottero *et al.* 2011; Gray & Miller 2011; Joekar-Nisar & Hassanizadeh 2011; Das & Mirzaei 2012; Hanspal & Das 2012) that describe two-phase flow processes in porous domains under the assumption of dynamic flow conditions are based on the use of a dynamic coefficient $\tau$. These dynamic coefficients determine the speed or ease with which flow equilibrium is attained and the dependence of capillary pressure on the time derivative of saturation, $\partial S/\partial t$. t establishes the speed at which flow equilibrium ($\partial S/\partial t = 0$) is reached. If $\tau$ is small, the equivalence between $P_{c,dyn}$ and $_{c,equ}$ is established quickly. On the other hand, the necessary time period to reach the equilibrium is high for larger $\tau$ values. The dynamic coefficient $\tau$ therefore behaves as a capillary damping coefficient and indicates the dynamics of the two-phase flow system.

Most of the experimental and computational flow-physics-based techniques for determining $P_c$–$S$ relationships and the corresponding dynamic effects through dynamic coefficients $\tau$ calculations are very resource intensive and exceedingly time consuming for complex three-dimensional flows in homogeneous or heterogeneous porous domains (Das *et al.* 2007; Mirzaei & Das 2007; Das & Mirzaei 2012; Hanspal & Das 2012). In order to circumvent these difficulties, we present an artificial neural network (ANN) model that can be effectively used to determine the dynamic coefficients $\tau$ for two-phase flow in porous media; in this particular study, PCE and water are the fluid components. The motivation to develop and apply an ANN model for two-phase flow computations results from the ability of ANNs to impose fewer constraints on the functional form of the relationships between input and output variables when the complexity of the systems is difficult to anticipate (Johnson & Rogers 2000). In the following section we discuss the background in more detail.

## Artificial neural networks (ANNs)

An ANN is a computational tool composed of simple elements operating in parallel (Demuth *et al.* 2008), commonly known as neurons, that can simulate the working of the human brain and the nervous system in learning to perform functions (an input/output map). The neurons are grouped into subsets (input, output and hidden layers) connected to one another, having bias and transfer functions associated with them. Generally, networks with biases, a sigmoid layer and a linear output layer are capable of approximating any function with a finite number of discontinuities. The weight and bias are adjustable scalar parameters of a neuron that are modified in a sequential mode, for the network to exhibit the desired behavior. The assigned weights in conjunction with the presence of hidden layers within the network help to determine the complicated relationships between the input and output data.

A back-propagation algorithm is used in this work to reduce the observed error in the predicted output variables by modifying the connection weights. Standard back-propagation includes a gradient descent algorithm, such as the Widrow–Hoff learning rule (Widrow 1962), for the multiple-layer networks and non-linear differentiable transfer functions in which the network weights are moved along the negative of the gradient of the performance function (Demuth *et al.* 2008; Khataee & Kasiri 2010). When the error measure of the network is reduced below a user-defined minimum, the training is stopped and the connection weights are recorded and used to perform computations. There are different architectures for neural networks which consequently require different types of algorithms but, despite an apparently complex system, a neural network is relatively simple serving two important functions: (1) pattern classifiers, and (2) non-linear adaptive filters.

The most commonly used ANN in engineering applications is the feed-forward network (Haykin 1999). The presence of multiple layers of neurons with non-linear transfer functions allows the network to learn non-linear relationships between input and output vectors.

In the context of flows within porous media, ANNs have been used for a variety of applications that include for example: prediction of gas diffusion layer properties within polymer electrolyte membrane (PEM) fuel cells (Kumbur *et al.* 2008; Lobato *et al.* 2010); prediction of dialysis performance in ultrafiltration (Godini *et al.* 2011); hygrothermal property characterization in porous soils (Coelho *et al.* 2009); oil saturation and petrophysical property predictions in oilfield sands (Boadu 2001); groundwater contamination and pollutant infiltration forecasting (Tabach *et al.* 2007); simulating cross-flow filtration processes (Silva & Flauzino 2008); optimization of groundwater remediation problems (Rogers & Dowla 1994; Johnson & Rogers 2000); large-scale water resource management (Yan & Minsker 2006); permeability modeling in petroleum reservoir management (Karimpouli *et al.* 2010); water/wastewater treatment using various homogeneous and heterogeneous nano-catalytic processes (Khataee & Kasiri 2010); determination of stress-strain characteristics in composites (Lefik *et al.* 2009); and characterization of outflow parameters influencing fractured aquifers outflows (Lallahem & Mania 2003).

For example, Rogers & Dowla (1994) proposed an ANN-based groundwater management model for optimizing aquifer remediation. The flow and transport model generated a set of sample data upon which the network could be trained. The study indicated that the ANN-based management solutions were consistent with those resulting from a more conventional optimization technique, which combined solute transport modeling and non-linear programming.

It is apparent from the studies described in the literature that, although most of them signify the importance and reliability of ANN techniques for implementing porous flows for a variety of engineering problems, none of them are directly concerned with the need for quantifying the dynamic effects and their influence on the flow of multiple phases. We therefore present for the first time an ANN-based framework for handling complex three-dimensional two-phase flow computations of dense non-aqueous phase liquid (DNAPL) displacements in the presence of dynamic effects in a robust, computationally economical and reliable fashion (in comparison to sophisticated numerical-methods-based computational fluid dynamics (CFD) simulators, which can be enormously time consuming for large-scale recurring calculations). We also discuss the development and training strategies employed for a variety of single- and double-hidden-layer network models. Finally, the best ANN and regression model architectures are identified for reliable dynamic coefficient predictions by evaluating the simulation results and the model performance on the basis of statistical performance parameters.

## ANN MODELING AND IMPLEMENTATION

The input or the reference data used for the ANN model development and training result from the modeling studies conducted within three-dimensional cylindrical coarse and fine sand domains (Das *et al.* 2007; Mirzaei & Das 2007; Hanspal & Das 2009, 2012) using immiscible DNAPL displacement experiments and quasi-static/dynamic 'water-oil' mode simulations handled via the Subsurface Transport Over Multiple Phases (STOMP) model (http://stomp.pnl.gov/; Nichols *et al.* 1997; White & Oostrom 2006).

The input data were collected from a number of previous studies (Das *et al.* 2007; Mirzaei & Das 2007; Hanspal & Das 2012), which indicated that these variables are important in determining the value of $\tau$. Furthermore, expert judgments were used to choose these variables. As explained above, these data were obtained using numerical (finite volume method) simulations with the main purpose to report the significance of the dynamic effect. The dataset included approximately 150 data points. The important statistics of the datasets are listed in Table 1.

In this work, a multilayer feed-forward network trained using a back-propagation training algorithm was implemented with MatLab's ANN toolbox and used to model the complex non-linear relationship persistent among the dynamic coefficient and physical properties characterizing the DNAPL displacement in multiphase transport. The back-propagation algorithm used for training the feed-forward neural network problem (Demuth *et al.* 2008) was implemented in four sequential steps discussed in detail in the following sections.

**Table 1** | Important statistics of the variables used in this study

| | Independent variable 1: water saturation $S_w$ | Independent variable 2: viscosity ratio $V_r = \mu_{nw}/\mu_w$ | Independent variable 3: density ratio $D_r = \rho_{nw}/\rho_w$ | Independent variable 4: permeability $K$ (m$^2$) | Independent variable 5: temperature $T$ (°C) | Dependent variable: dynamic coefficient $\tau$ (Pa s) |
|---|---|---|---|---|---|---|
| Range | 0.105–0.464 | 0.5–2 | 0.5–2 | $5.00\times10^{-11}$–$5.00\times10^{-9}$ | 20–80 | $2.82\times10^{5}$–$1.05\times10^{11}$ |
| Arithmetic average value | 0.257 | 0.946 | 1.39 | $3.43\times10^{-9}$ | 24.55 | $1.26\times10^{10}$ |
| Standard deviation | 0.091 | 0.32 | 0.47 | $2.31\times10^{-9}$ | 13.89 | $2.69\times10^{10}$ |

## Data assimilation

The data described above were imported to MatLab by using calling functions to ensure that while one independent variable was changed others remained constant, resulting in variations in the dynamic coefficient values. This procedure was repeated to train the network on each of the independent variables in order to produce an output close enough to the target (dependent variable). The input (independent) variables are denoted $p$ while the output (dependent) variables are represented by the target $t$.

## Network object creation

MatLab's ANN toolbox was utilized to create a feed-forward network requiring three arguments before returning the network object. The network object was created after providing the input and output parameters which then initialized the weight and bias values to determine the size of the output layer. In addition, the input data were segregated into three different sets, namely the training, validation and test data in a split of 60, 20 and 20%, respectively.

Two-layer (single-hidden) and three-layer (two-hidden) feed-forward networks were developed and investigated in this study. The two-layer network has the typical format of the input variables, the target and the number of hidden neurons {$p$, $t$, 3} while the three-layer network is characterized by two sets of hidden layer neurons {$p$, $t$, [3 5]}.

## Network training

After the network weights and biases were initialized, the network was trained for function approximation (non-linear regression), pattern association and pattern classification. During training, the weights and biases of the network were adjusted iteratively to minimize the network performance function. The default performance function for the feed-forward networks is mean square error (mse), which is the average difference of the squared errors (Demuth *et al.* 2008) between the network ($a$) and the target outputs ($t$). The training was carried out in MatLab by segregating the available data into three datasets: 60% for training, 20% for validation and 20% for testing. Training was conducted multiple times in conjunction with using five-fold cross-validation to ensure each of the data points in the 150-point dataset was a part of 60% test dataset. The network training data were then utilized to recognize the behavioral patterns in the data, validated in order to assess the network generalization and tested to provide an independent evaluation of network generalization for new data that the network had not experienced previously.

The process parameters, goal and epoch were used to determine the stopping criteria for the network training. The training was stopped when the number of iterations exceeded the epochs or if the performance function dropped below the set tolerance value. The training was carried out until there was a continued reduction in the network error on the validation vectors. After the network memorized the training set, training was terminated. Re-initialization of the network was pursued depending upon its accuracy. The initial weights of nodes are assigned randomly in the MatLab toolbox, so repeated training may result in different ANN performances. In this work, each ANN was trained multiple times. The number of hidden neurons was varied gradually, since large neuron numbers within the hidden layer gave the

network more flexibility due to multiple parameter optimization. Under-characterization was tackled by instructing the network to optimize more parameters than the number of data vectors.

Although, the training was carried out in MatLab by segregating the available data randomly into three datasets (i.e. 60% for training, 20% for validation and 20% for testing), the well-known measure of stratified sampling was applied to ensure that the statistics of the testing and training data are in close vicinity.

### Network response simulation

After the network was trained, it was re-applied to the original vectors. Network outputs were produced by incorporating the network input and the network object, and finally applied to simulate dynamic coefficient values for a range of input parameters.

## PRE- AND POST-PROCESSING PROCEDURES: ANN MODEL TRAINING

Specific pre- and post-processing steps discussed in the subsequent section were required to train the ANN model effectively, as described in the following sections.

### Pre-processing procedure

When the network is created using MatLab's ANN toolbox, default processing functions are automatically assigned to the network inputs and outputs. These functions were overridden by adjusting the network parameters. User-defined functions were used to scale up the network where the epoch limit was set to be 200 iterations. The {*mapstd*} function was utilized in the scale-up operation by normalizing the mean and standard deviation of the training set to be 0 and 1, respectively. No other scaling functions or correction factors for the inputs and outputs were utilized in this work.

It was also important that we did not use too much over-fitting of the data. This was ensured as follows. First of all, we used a relatively large dataset as compared with the number of points needed to plot the $\tau$–$S$ curve. In our case, the dataset was about 30–35 times greater than the typical number of points (five to six points) needed to plot a $\tau$–$S$ curve. Secondly, we used a relatively simple ANN structure. This ensures that there is no artificial over-fitting of the data, as may be observed in complex ANN structures.

### Post-processing analysis

As an additional measure, regression analysis was carried out using the network outputs and the corresponding targets to validate the network performance.

## ANN MODEL PERFORMANCE TESTING AND CALIBRATION

The performance of various ANN models developed in this work were analyzed against standard performance parameters and criteria (Jain *et al.* 2001), described in the following sections. The performances were calculated using the entire dataset.

### Sum squared error (SSE)

The summed square of residuals (SSE) represents the total deviation of the simulated values in comparison to the observed values. This is defined:

$$\mathrm{SSE} = \sum_{i=1}^{N} (S_{\mathrm{obs}} - S_{\mathrm{cal}})^2 \tag{2}$$

where $N$ is total number of data points predicted; $S_{\mathrm{obs}}$ is the observed value of dynamic coefficient $\tau$; and $S_{\mathrm{cal}}$ is the calculated value of dynamic coefficient $\tau$.

### Average absolute relative error (AARE)

The average of the relative errors (AARE) commonly expressed as a percentage were calculated via:

$$\mathrm{AARE} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{S_{\mathrm{cal}} - S_{\mathrm{obs}}}{S_{\mathrm{obs}}} \right| \times 100 \tag{3}$$

Lower values of AARE indicate better model performance.

### Nash–Sutcliffe efficiency coefficient *E*

The Nash-Sutcliffe efficiency coefficient $E$ is defined:

$$E = 1 - \frac{\sum (S_{\text{cal}} - S_{\text{obs}})^2}{\sum (S_{\text{obs}} - \bar{S}_{\text{obs}})^2} \tag{4}$$

where $\bar{S}_{\text{obs}}$ is the average observed dynamic coefficient $\tau$. Values of $E$ nearing 1.0 indicate a perfect match between the observed data and outputs, signifying high model accuracy.

### Pearson product moment coefficient of correlation *R*

The Pearson product moment coefficient $R$ is defined:

$$R = \frac{\sum (S_{\text{obs}} - \bar{S}_{\text{obs}}) \times (S_{\text{cal}} - \bar{S}_{\text{cal}})}{\sqrt{\sum (S_{\text{obs}} - S_{\text{obs}})^2 \sum (S_{\text{cal}} - \bar{S}_{\text{cal}})^2}} \tag{5}$$

where $\bar{S}_{\text{cal}}$ is the average calculated dynamic coefficient $\tau$. Equation (5) was used to characterize the strength of linear dependency in the relationship between simulated and observed data. Values of $R$ nearing unity indicate a good model.

### Threshold statistics (TS)

The threshold statistics for a level of absolute relative error (ARE) of $x$% are defined:

$$\text{TS} = \frac{N_x}{N} \tag{6}$$

where $N_x$ is the number of data points predicted for which the average relative error (ARE) is less than $x$%. Equation (6) quantifies the consistency in the prediction errors (Jain & Ormsbee 2002). Large values of threshold statistics indicate better model performance.

## REGRESSION MODELING OF DYNAMIC COEFFICIENT

Linear and non-linear regression models were also developed as a part of this study to make comparisons with the predictions obtained using the ANN model. MatLab was utilized for all the regression modeling work carried out, described in the following sections. The regressions were calculated using the entire dataset.

### Linear multiple regression

The dynamic coefficient $\tau$ was regressed against the independent variables, i.e. water saturation, viscosity ratio, density ratio, permeability and temperature:

$$\tau = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \tag{7}$$

where $\tau$ is the dynamic coefficient; $\beta_0 \rightarrow \beta_5$ are the regression coefficients to be estimated; and $x_1 \rightarrow x_5$ are the independent variables.

Since the resulting system of equations was over-determined (William 2005), the left division method (based on Gauss elimination and least-square techniques) was used to determine the matrix coefficients which best fit the datasets. Using this technique, the data are arranged in a matrix:

$$X\beta = \tau \tag{8}$$

where

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_5 \end{bmatrix} \quad X = \begin{bmatrix} 1 & x_{11} & x_{21} & x_{31} & x_{41} & x_{51} \\ 1 & x_{12} & x_{22} & x_{32} & x_{42} & x_{52} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_{1N} & x_{2N} & x_{3N} & x_{4N} & x_{5N} \end{bmatrix} \quad \tau = \begin{bmatrix} \tau_0 \\ \tau_1 \\ \dots \\ \tau_N \end{bmatrix}$$

$x_{1i} \rightarrow x_{5i}$ and $\tau_i$ represent the data, and $i = 1,\dots, N$ where $N$ is the number of data points. The solution for the coefficients is computed:

$$\beta = X \backslash \tau \tag{9}$$

where the backslash operator performs matrix left division.

$x$ minimizes norm $(X*\beta - \tau)$, the length of the vector $X\beta - \tau$ (Demuth *et al.* 2008).

### Non-linear multiple regression

Using similar variables as in the case of linear regression, polynomials of various orders (Jain & Indurthy 2003) represented by Equations (10) and (12) were used to regress the dynamic coefficient against water saturation, viscosity ratio, density ratio, permeability and temperature:

$$\tau = \beta_0 + \beta_1(x_1)^2 + \beta_2(x_2)^2 + \beta_3(x_3)^2 + \beta_4(x_4)^2 + \beta_5(x_5)^2 \quad (10)$$

$$\tau = \beta_0 + \beta_1(x_1)^{0.05} + \beta_2(x_2)^{0.05} + \beta_3(x_3)^{0.05} + \beta_4(x_4)^{0.05} + \beta_5(x_5)^{0.05} \quad (11)$$

$$\tau = \beta_0 + \beta_1(x_1)^4 + \beta_2(x_2)^4 + \beta_3(x_3)^4 + \beta_4(x_4)^4 + \beta_5(x_5)^4 \quad (12)$$

Function based on the Gauss–Newton algorithm with Levenberg–Marquardt modifications for global convergence (Demuth *et al.* 2008) were used to determine the least-squares parameter estimates.

## RESULTS AND DISCUSSION

The reference data used in developing and training various neural network models for predictive modeling of dynamic coefficients through incorporation of dynamic effects are described in Das *et al.* (2007), Mirzaei & Das (2007) and Hanspal & Das (2012). As described in the 'Data assimilation' section, the data comprise five independent and one dependent output parameter.

### ANN models

Two different types of ANN models were developed in this work: (1) single-hidden-layer model, and (2) double-hidden-layer model. For each type of model, the number of neurons in the input layer and the output layer were kept the same. The number of neurons in the hidden layer was determined using a trial-by-error procedure proposed by Jain & Indurthy (2003). The optimal model architecture was determined by varying the number of hidden neurons from 3 to 17 and performing a post-training analysis on each of the network models. As discussed before, slope $m$, correlation coefficient $r$ and intercept $c$ values in the proximity of 1 and 0 indicate an optimal model. The model training and post-training regression analysis plots for the best ANN networks developed in this work are depicted in Figures 1–4.

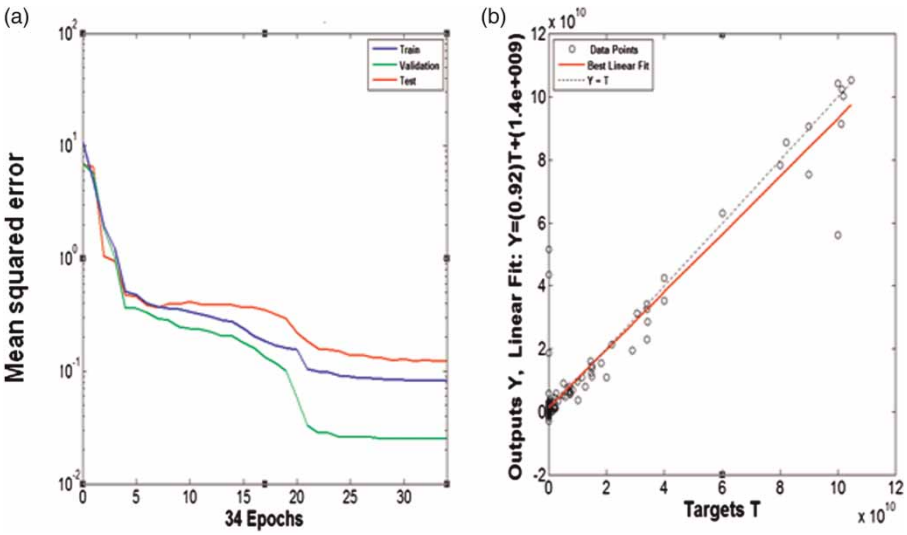


**Figure 1** | Trained network and post-training regression analysis for [5-7-1] ANN model.

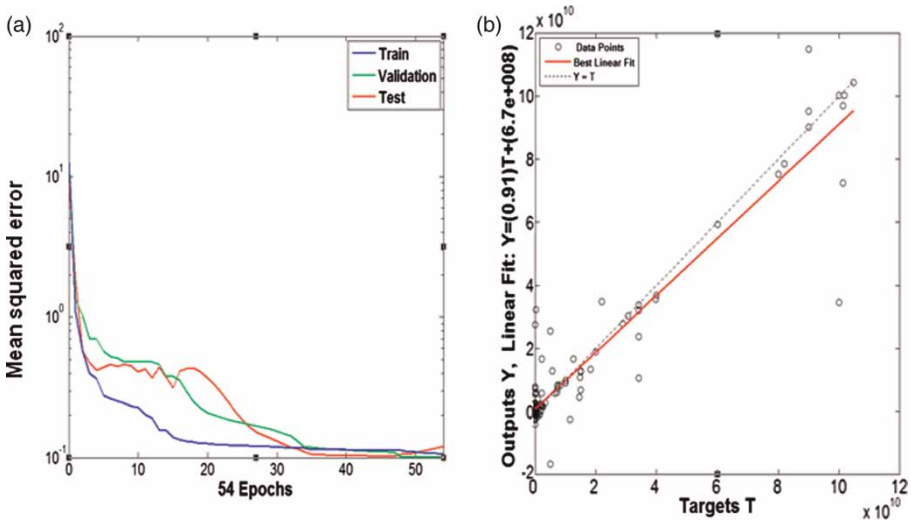


**Figure 2** | Trained network and post-training regression analysis for [5-9-1] ANN model.

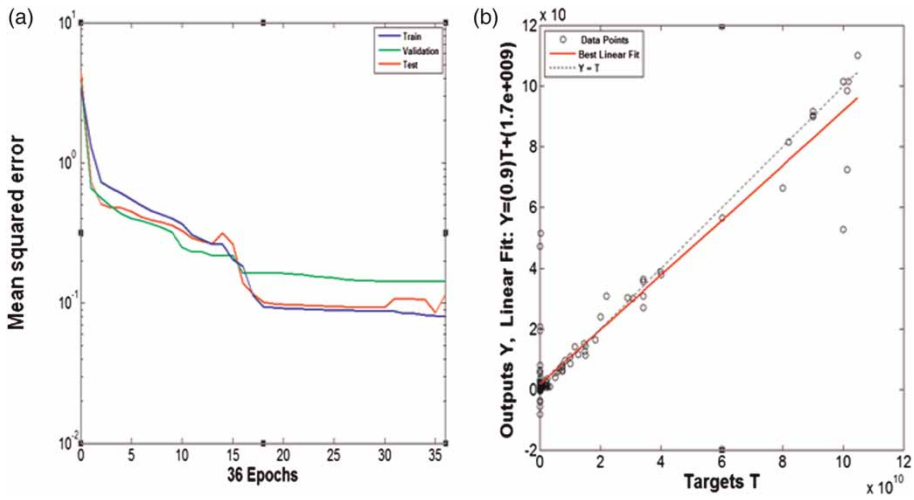


**Figure 3** | Trained network and post-training regression analysis for [5-9-11-1] ANN model.

The mean-squared errors in Figures 1(a)–4(a) gradually decrease as the learning and the training process continues. The performance goal indicates that the convergence of the mean-squared error training, validation and test plots was set to the default mode of zero. The number of iterations (epochs) were different in all the distinct models reported, since the validation test stops the network training when the peak performance is attained. Figures 1(b)–4(b) represent the post-training regression analyses of the network models depicting the perfect line, outputs = targets ($Y = T$) and the best linear regression line for the data points. The best linear regression line is then used for evaluating the slope, correlation coefficient and the intercept. Table 2 lists the performance values of the ANN models on the basis of post-training line regression plots used in the determination of the number of hidden neurons which produce the most accurate fit.

From the slopes and the correlation coefficients listed in Table 2, it is clear that the network is best trained when there are seven or nine hidden neurons for the single-hidden-layer model structure and a combination of [9 11], [13 15] or [11 13] hidden neurons for the double-hidden-layer models.

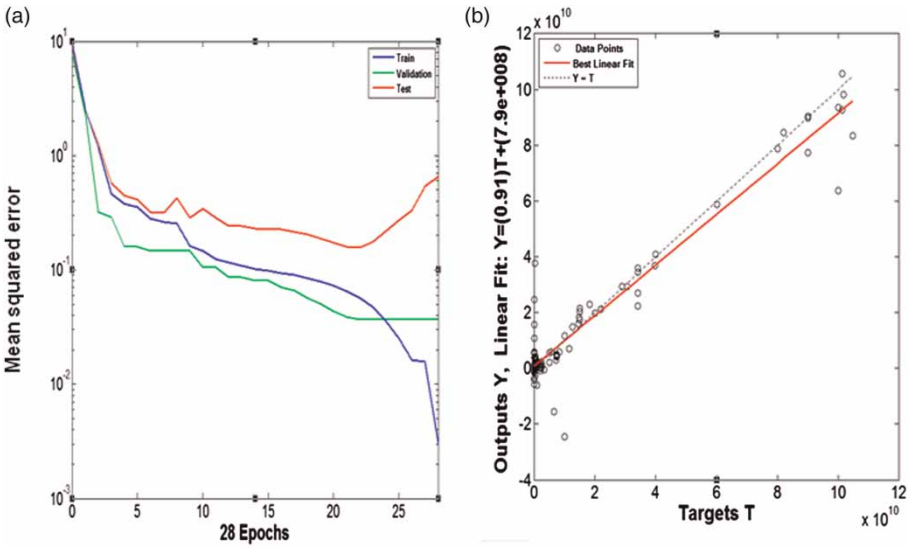


**Figure 4** | Trained network and post-training regression analysis for [5-11-13-1] ANN model.

**Table 2** | ANN model performance determined using post-training regression analysis

| Number of hidden layers | Artificial neural network model | Correlation coefficient *r* | Slope *m* |
|---|---|---|---|
| 1 | ANN (5-3-1) | 0.8396 | 0.7363 |
| 1 | ANN (5-5-1) | 0.8950 | 0.8359 |
| 1 | ANN (5-7-1) | 0.9578 | 0.9183 |
| 1 | ANN (5-9-1) | 0.9439 | 0.9056 |
| 1 | ANN (5-11-1) | 0.8634 | 0.7134 |
| 1 | ANN (5-13-1) | 0.8752 | 0.7715 |
| 2 | ANN (5-3-5-1) | 0.9455 | 0.8705 |
| 2 | ANN (5-5-7-1) | 0.9499 | 0.8786 |
| 2 | ANN (5-7-9-1) | 0.9492 | 0.8722 |
| 2 | ANN (5-9-11-1) | 0.9489 | 0.9025 |
| 2 | ANN (5-11-13-1) | 0.9632 | 0.9077 |
| 2 | ANN (5-13-15-1) | 0.9243 | 0.9595 |
| 2 | ANN (5-15-17-1) | 0.9248 | 0.8938 |

Only these models have their correlation coefficients and slopes above 0.9. In terms of the correlation coefficient, the double-hidden-layer structure ANN [11 13] performed best with a value of 0.9632 while the ANN [13 15] model structure performed best with respect to the slope with a value of 0.9595. In comparison to the double-hidden-layer networks, the single-layer networks ANN [7] and ANN [9] models have correlation coefficient values of 0.9578 and 0.9439, respectively.

## Regression models

The values of the regression coefficients for the linear and non-linear regression models used in this work are listed in Table 3. The performances of the regression models were further evaluated and compared with the ANN model performances, enlisted in the subsequent section using the criteria described in the 'ANN model performance testing and calibration' section.

## Model performance criteria evaluation

Model performance parameters (AARE, SSE, *R*, *E* and TS) were computed to determine the performance of the ANN and regression models. Plots depicted in Figures 5–8 provide a better means to evaluate and compare the performance of the ANN and regression models.

From the absolute average relative error (AARE) tests, characterized in Figure 5, it can be seen that the regression models performed badly; the linear regression model performs the worst. The best of the single-layer and double-hidden-layer structures performed almost identically.

Figure 6 illustrates the comparison of the sum squared errors (SSE), signifying that the regression models perform poorly with the non-linear regression model-2 performing the worst (7.13333). The ANN [7] and ANN

**Table 3** | Linear and non-linear regression coefficients

| | | | Non-linear multiple regression (NLR) models | | | |
|---|---|---|---|---|---|---|
| **Regression parameter** | **Independent variable** | **Linear multiple regression (LR) model** | **Model 1 (NLR 1)** | **Model 2 (NLR 2)** | **Model 3 (NLR 3)** | **Model 4 (NLR 4)** |
| $\beta_0$ | Constant | 0.5223 | 9.1625 | 0.1785 | −0.6614 | 6.9875 |
| $\beta_1$ | $x_1$ | −0.8281 | −9.5799 | −0.6248 | 1.4945 | −4.1225 |
| $\beta_2$ | $x_2$ | −0.0983 | −1.4919 | 0.0051 | 0.2166 | −11.7713 |
| $\beta_3$ | $x_3$ | 0.0791 | 0.5146 | 0.0930 | −0.1092 | 0.8174 |
| $\beta_4$ | $x_4$ | −0.0009 | −0.0248 | 0.0060 | 0.0044 | −2.0980 |
| $\beta_5$ | $x_5$ | 0.1610 | 1.2602 | 0.1471 | −0.22358 | 1.1004 |

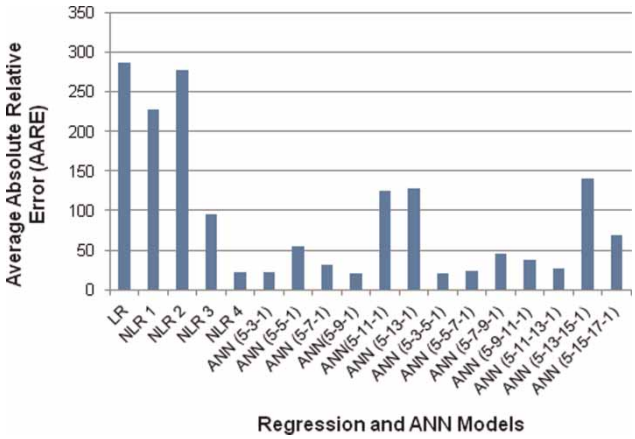


**Figure 5** | Average absolute relative errors (AARE): comparison of various regression (linear LR and non-linear NLR) and ANN models.

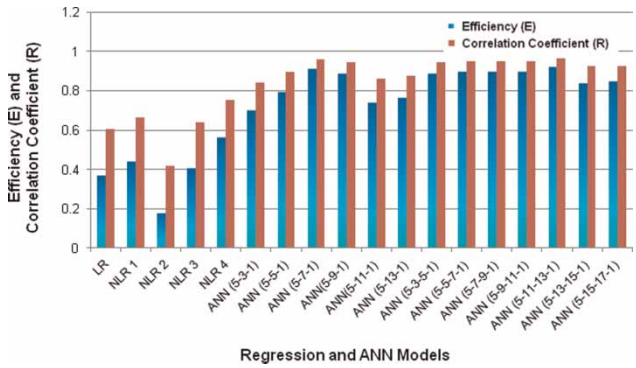


**Figure 7** | Efficiency $E$ and correlation coefficient $R$: comparison of various regression (linear LR and non-linear NLR) and ANN models.

[11 13] performed best with the latter having a better value of 0.7175.

Comparisons using efficiency $E$ and correlation coefficient $R$ presented in Figure 7 also illustrates the underperformance of regression models with the non-linear regression model-2 performing the worst. The ANN [7] and ANN [11 13] models have the best performance with the efficiency $E$ and correlation coefficient $R$ for ANN [7] being 0.9171 and 0.9578, respectively. ANN [11

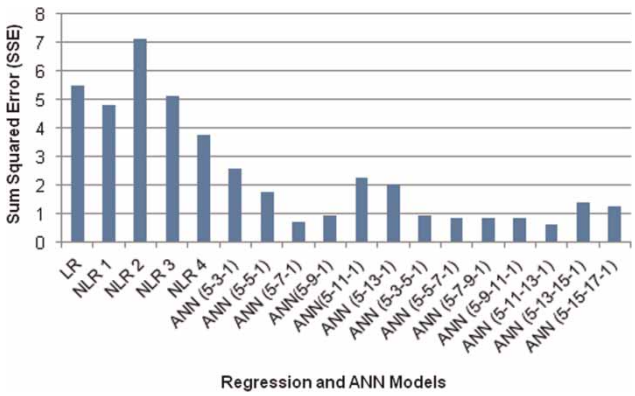


**Figure 6** | Sum squared errors (SSE): comparison of various regression (linear LR and non-linear NLR) and ANN models.

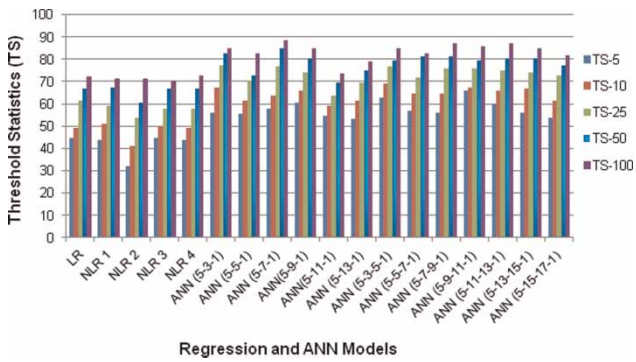


**Figure 8** | Threshold statistics (TS-5, TS-10, TS-25, TS-50 and TS-100): comparison of various regression (linear LR and non-linear NLR) and ANN models.

**Table 4** | Model performance comparisons for ANN [7] and ANN [11 13], the best-performing single- and double-hidden-layer ANN structures

| Model | *R* | *E* | SSE | AARE | TS-5 | TS-10 | TS-25 | TS-50 | TS-100 |
|---|---|---|---|---|---|---|---|---|---|
| ANN-7 | 0.9578 | 0.9171 | 0.7175 | 31.2250 | 57.58 | 63.65 | 76.52 | 84.85 | 88.64 |
| ANN-[11 13] | 0.9632 | 0.9272 | 0.6306 | 27.4732 | 59.85 | 65.91 | 75.00 | 80.30 | 87.12 |

13] performed slightly better with values of 0.9272 for $E$ and 0.9632 for $R$, respectively.

From the threshold statistics (TS) plot in Figure 8, it is can be seen that the regression models again perform poorly; the non-linear regression model-2 has the lowest value for threshold statistic (TS-5). The ANN [7] model, on the other hand, performed best for TS-100 with a value of 88.64 while the ANN [9 11] model performed best for TS-5 with a value of 65.91.

It is concluded that the regression models generally perform poorly in comparison to ANN models. The ANN [7] in the category of single-hidden-layer models and the ANN [11 13] within the class of double-hidden-layer models demonstrate the best performance. Double-hidden-layer network models performed slightly better in comparison to the single-layer-network models; they have better performance values in all the tests except for the threshold statistics, where they had lower values in comparison to the single-layer network (Table 4).

## Model simulations: Dynamic coefficient–water saturation relationship

The functional relationships between the dynamic coefficient and water saturation can typically be characterized by smooth curves, exemplifying decreasing dynamic coefficient values for increasing water saturation. Dynamic coefficients values were obtained using the best ANN and regression models developed in this work and compared against the corresponding target values obtained using the reference data (Das *et al*. 2006, 2007; Mirzaei and Das 2007; Hanspal & Das 2009, 2012). Dynamic coefficient versus water saturation plots presented in Figures 9–11 were developed using the simulated data from the single, double-hidden-layer ANN and regression model simulations. Comparisons have been made for determining the forecasting accuracy of the ANN and regression models regarding a typical inverse relation between the dynamic coefficient and water saturation.

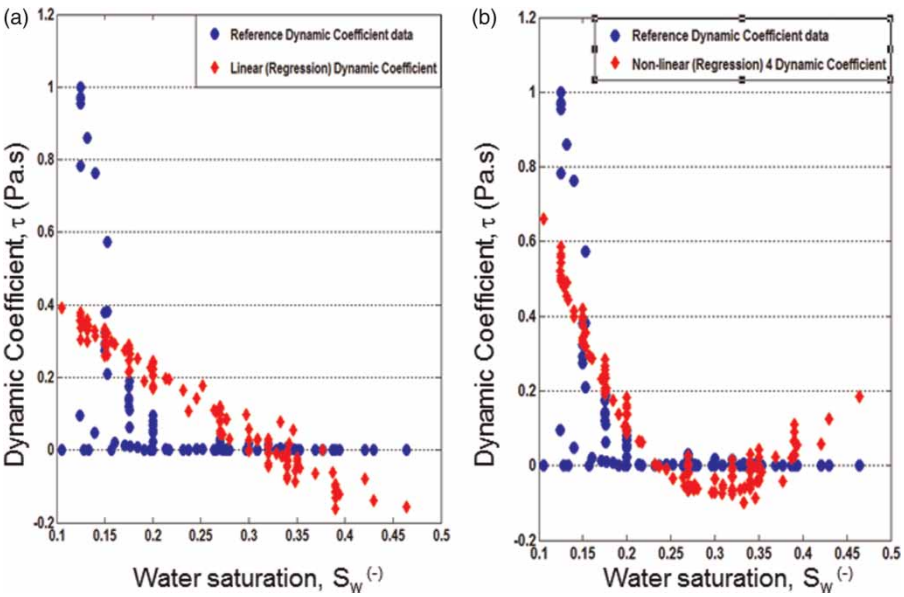


**Figure 9** | Regression model performance for predicting dynamic coefficient–water saturation relationship.

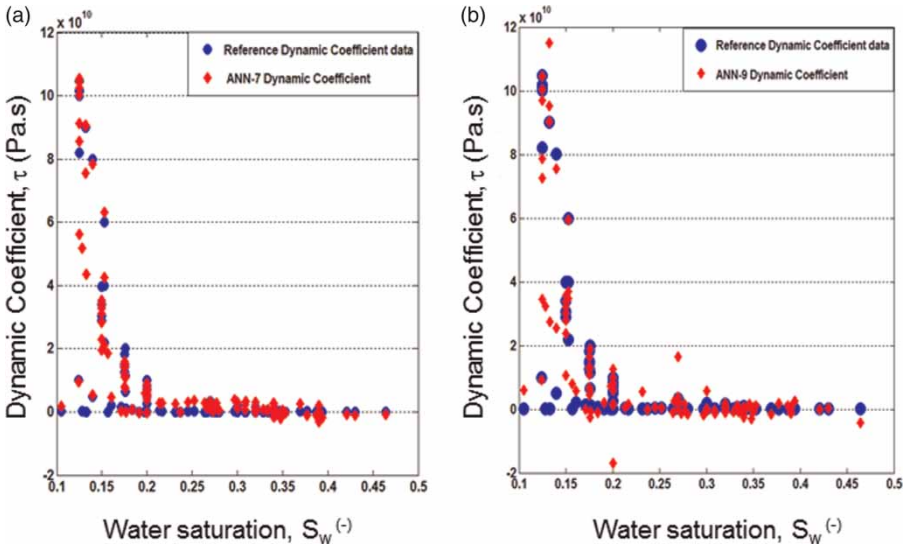


**Figure 10** | Single-hidden-layer ANN model performance for predicting dynamic coefficient–water saturation relationship. The ANN structure with [7] neurons in the hidden layer seems to perform better.

Figures 9(a) and (b) demonstrate the predictive capabilities of the linear and non-linear regression models, which are deemed to be poor. The non-linear model performs better in comparison to the linear regression model, but still fails to accurately represent the characteristic behavior of the reference data. Single-layer ANN model predictions compare very well against the reference data illustrated in Figures 10(a) and (b).

The network model with seven hidden neurons successfully simulates the dynamic coefficient values, which fall in close proximity to the reference data with fewer errors in comparison to the ANN model with nine neurons in the hidden layer. The simulations were carried out using the double-hidden-layer ANN models presented in Figures 11(a) and (b).

The simulated data represent predictions of the dynamic coefficient values and characteristics of the

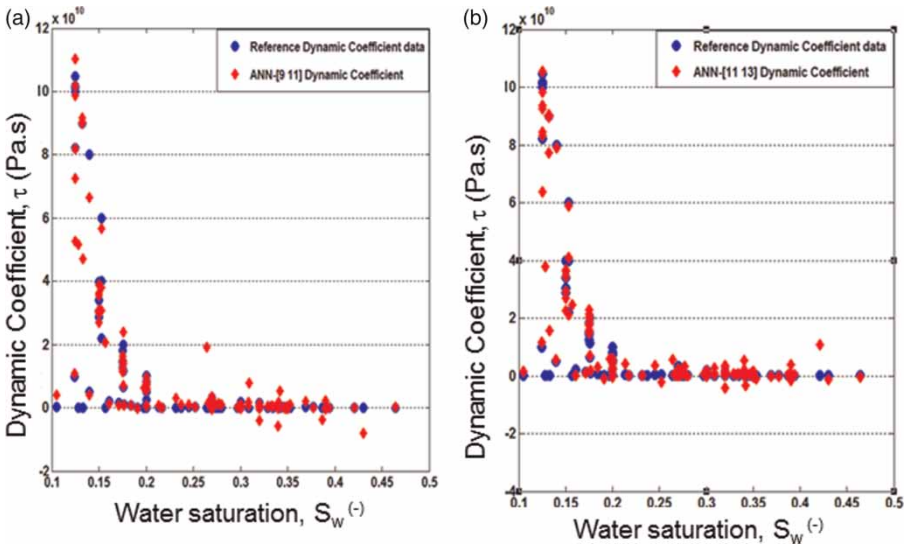


**Figure 11** | Double-hidden-layer ANN model performance for predicting dynamic coefficient–water saturation relationship. The ANN structure with [11 13] neurons in the hidden layers seem to perform better.

inverse dynamic coefficient–water saturation relationship, which again compare well against the reference data. However, there are more wayward data points in comparison with the single-hidden-layer ANN model predictions. Simulations resulting from ANN [11 13] hidden neuron model structure compare much better in comparison to the ANN [9 11] model. Finally, the best performing single-hidden-layer ANN [7] and double-hidden-layer ANN [11 13] models from all the simulations and performance analysis are compared in Figure 12.

Observing Figure 12, it can be concluded that the double-layer-hidden ANN [11 13] model contains more prediction errors in comparison to the single-hidden-layer ANN [7] model. Closer inspection reveals that the double-hidden-layer ANN network performs better for low water saturation values, enabling the prediction of high dynamic coefficient values which closely resemble the reference data (as determined by immiscible displacement experiments and complex three-dimensional flow-physics-based CFD computations; Hanspal & Das 2012). As the water saturation values start to increase, the single-hidden-layer ANN [7] model better predicts the dynamic coefficient variations than the double-hidden-layer ANN [11 13] model. The ANN [11 13] model had a great many wayward values; this was the same for all the double-hidden-layer models. The network structure of the best-performing models include: (1) single-layer ANN model containing five input, seven hidden layer and one output neuron; and (b) double-layer ANN model containing five input, [11 13] hidden layer and one output neuron. These can be used reliably for predicting dynamic coefficient–water saturation relationships, as depicted in Figures 13 and 14.

## CONCLUSIONS

We have demonstrated the successful application of ANN (single- and multiple-hidden layered) and regression modeling techniques (linear and non-linear) for determining complex relationships between the dynamic coefficient and the physical parameters characterizing the porous medium and the fluid properties. The data deployed for model development, network training, performance evaluation and subsequent analysis was acquired from computational physics-based modeling studies (Mirzaei & Das 2007; Hanspal & Das 2009, 2012). It has been demonstrated that significant computational savings can be obtained by using the ANN models (Figures 13 and 14) in

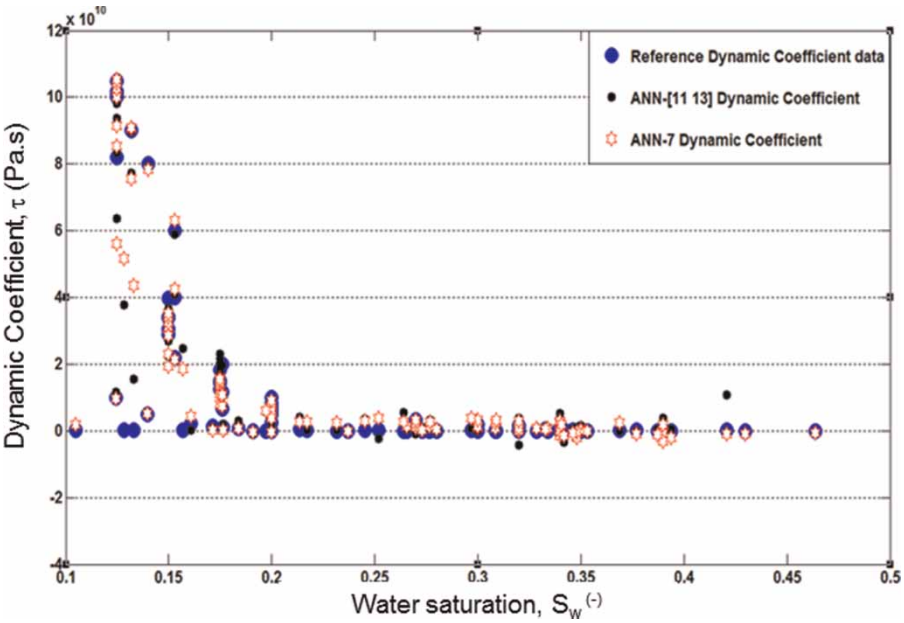


**Figure 12** | Comparisons of the best-performing single- and double-hidden-layer ANN models for predicting dynamic coefficient–water saturation relationship simulations. The performances of the two structures seem to be similar, suggesting that a single-hidden-layer ANN model may be chosen.

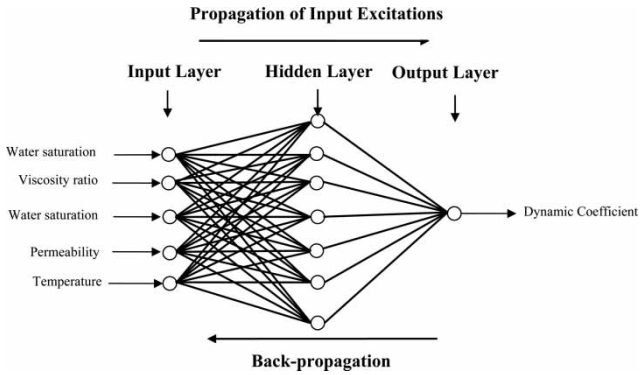


**Figure 13** | Best-performing single-layer ANN model containing five inputs, seven hidden layers and one output neuron.

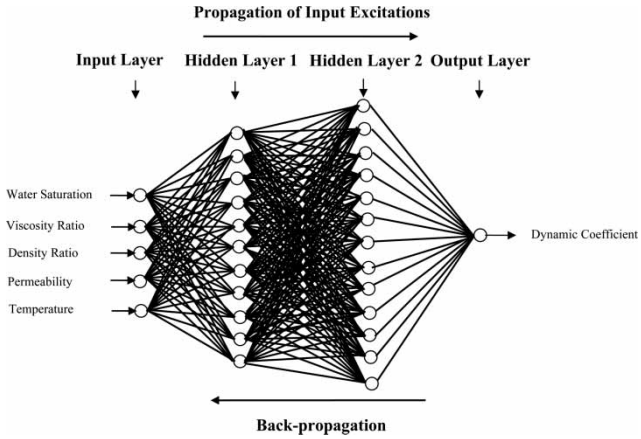


**Figure 14** | Best-performing double-layer ANN model containing five inputs, [11 13] hidden layer and one output neuron.

comparison to the flow-physics-based CFD simulators. These cost savings are indicative of the reduced simulation timescales required for determining the complex functional relationships for dynamic coefficient variations resulting from dynamic effects within multiphase flows. Excessive over-fitting of the data was also avoided.

It is concluded that ANNs can model the behavioral relationship between the changes in the media and fluid properties, reliably predicting dynamic coefficients in comparison to regression models. From the performance statistics parameters which comprise the average absolute relative error, sum squared errors and the efficiency, the double-hidden-layer ANN model seems to perform better in comparison to the single-hidden-layer ANN model with similar threshold statistics. However, from the simulation plots it was determined that single-hidden-layer ANN [7] model is a better predictor for high water saturation content in comparison to the double-hidden-layer ANN [11 13] model, while at low water saturation ANN [11 13] performs more reliably. In most cases, however, the differences in the model predictions were small and it can be concluded that, for most practical work, a well-trained and validated single-layer ANN structure should suffice.

Results from this work demonstrate that ANN models operating within a hybrid framework of both single- and double-hidden-layer neurons for a range of water saturation contents provide a boost to parameter estimation and simulation.

## ACKNOWLEDGEMENTS

We acknowledge the financial support from the EPSRC, UK for funding the project GR/S94315/01, Micro-Heterogeneity Effects on Dynamic Capillary Pressure–Saturation Relationships in Porous Media, which enabled us to conduct the CFD modeling studies to generate the reference data used for the ANN simulations in this work. DBD is very grateful to Professor Ashu Jain (IIT Kharagpur, India) for many helpful communications.

## REFERENCES

Boadu, F. K. 2001 Predicting oil saturation from velocities using petrophysical models and artificial neural networks. *Journal of Petroleum Science and Engineering* **30**, 143–154.

Bottero, S., Hassanizadeh, S. M., Kleingeld, P. J. & Heimovaara, T. J. 2011 Non-equilibrium capillarity effects in two-phase flow through porous media at different scales. *Water Resources Research* **47**, Article Number: W10505.

Brooks, R. H. & Corey, A. T. 1964 Hydraulic properties of porous media. Hydrology Paper 3, Colorado State University, Fort Collins.

Coelho, L. S., Freire, R. Z., Santos, G. H. & Mendes, N. 2009 Identification of temperature and moisture content fields using a combined neural network and clustering method approach. *International Communications in Heat and Mass Transfer* **36**, 304–313.

Das, D. B. & Mirzaei, M. 2012 Dynamic effects in capillary pressure relationships for two-phase flow in porous media: Experiments and Numerical Analyses. *AIChE Journal* **58**, 3891–3903.

Das, D. B., Gauldie, R. & Mirzaei, M. 2007 Dynamic effects in capillary pressure relationships for two-phase flow in porous media: Implications of fluid properties. *AIChE Journal* **53** (10), 2505–2520.

Das, D. B., Mirzaei, M. & Widdows, N. 2006 Non-uniqueness in capillary pressure-saturation-relative permeability relationships for two-phase flow in porous media: Interplay between distribution and intensity of micro-scale heterogeneities. *Chemical Engineering Science* **61**, 6786–6803.

Demuth, H., Beale, M. & Hogan, M. 2008 *Neural Network Toolbox 6: User's Guide*. Math Works, Inc., Natick, CO.

Godini, H. R., Ghadrdan, M., Omidkhah, M. R. & Madaeni, S. S. 2011 Part II: Prediction of the dialysis process performance using Artificial Neural Network (ANN). *Desalination* **265**, 1–3.

Gray, W. G. & Miller, C. T. 2011 TCAT analysis of capillary pressure in non-equilibrium, two-fluid phase, porous medium systems. *Advances in Water Resources* **34** (6), 770–778.

Hanspal, N. & Das, D. B. 2009 Dynamic effects for two-phase flow in porous media: temperature effects. AIChE Summer Meeting, 2009, Tampa, USA.

Hanspal, N. & Das, D. B. 2012 Dynamic effects on capillary pressure-saturation relationships for two-phase porous flow: implications of temperature. *AIChE Journal* **58**, 1951–1965.

Haykin, S. 1999 *Neural Networks – A Comprehensive Foundation*. 2nd edn. Prentice-Hall, Englewood Cliffs, NJ.

Jain, A. & Ormsbee, L. E. 2002 Evaluation of short-term water demand forecast modelling techniques: Conventional v/s artificial intelligence. *Journal of American Water Works Association* **94** (7), 64–72.

Jain, A. & Indurthy, P. 2003 Comparative analysis of event-based rainfall-runoff modelling techniques – deterministic, statistical and artificial neural networks. *Journal of Hydrological Engineering* **9** (6), 1084–0699.

Jain, A., Varshney, A. K. & Joshi, U. C. 2001 Short-term water demand forecast modelling at IIT Kanpur using artificial neural networks. *Water Resources Management* **15** (5), 299–321.

Joekar-Nisar, V. & Hassanizadeh, S. M. 2011 Effect of fluids properties on non-equilibrium capillarity effects: Dynamic pore-network modeling. *International Journal of Multiphase Flow* **37**, 198–214.

Johnson, V. M. & Rogers, L. L. 2000 Accuracy of neural network approximators in simulation-optimization. *Journal of Water Resources Planning and Management (ASCE)* **126**, 48–56.

Karimpouli, S., Fathianpour, N. & Roohi, J. 2010 A new approach to improve neural networks' algorithm in permeability prediction of petroleum reservoirs using supervised committee machine neural network (SCMNN). *Journal of Petroleum Science and Engineering* **73**, 227–232.

Khataee, A. R. & Kasiri, M. B. 2010 Artificial neural networks modeling of contaminated water treatment processes by homogeneous and heterogeneous nanocatalysis. *Journal of Molecular Catalysis A: Chemical* **331**, 86–100.

Kumbur, E. C., Sharp, K. V. & Mench, M. M. 2008 A design tool for predicting the capillary transport characteristics of fuel cell diffusion media using an artificial neural network. *Journal of Power Sources* **176**, 191–199.

Lallahem, S. & Mania, J. 2003 A nonlinear rainfall-runoff model using neural network technique: example in fractured porous media. *Mathematical and Computer Modelling* **37**, 1047–1061.

Lefik, M., Boso, D. P. & Schrefler, B. A. 2009 Artificial neural networks in numerical modelling of composites. *Computational Methods in Applied Mechanics and Engineering* **198**, 1785–1804.

Lobato, J., Canizares, P., Rodrigo, M. A., Piuleac, C. G., Curteanu, S. & Linaresa, J. J. 2010 Direct and inverse neural networks modelling applied to study the influence of the gas diffusion layer properties on PBI-based PEM fuel cells. *International Journal of Hydrogen Energy* **35**, 7889–7897.

Mirzaei, M. & Das, D. B. 2007 Dynamic effects in capillary pressure saturations relationships for two-phase flow in 3D porous media: implications of micro-heterogeneities. *Chemical Engineering Science* **62**, 1927–1947.

Nichols, W. E., Aimo, N. J., Oostrom, M. & White, M. D. 1997 *STOMP Application Guide*. PNNL-11216, UC-2010. Pacific Northwest National Laboratory, Richland, Washington.

O'Carroll, D. M., Mumford, K. G., Abriola, L. M. & Gerhard, J. I. 2010 Influence of wettability variations on dynamic effects in capillary pressure. *Water Resources Research* **46**, 578.

Rogers, L. L. & Dowla, F. U. 1994 Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. *Water Resources Research* **30**, 457–481.

Silva, I. N. & Flauzino, R. A. 2008 An approach based on neural networks for estimation and generalization of crossflow filtration processes. *Applied Soft Computing* **8**, 590–598.

Tabach, E. L., Lancelot, L., Shahrour, I. & Najjar, Y. 2007 Use of artificial neural network simulation metamodelling to assess groundwater contamination in road project. *Mathematical and Computer Modelling* **45**, 766–776.

Tsakiroglou, C. D., Theodoropoulou, M. A. & Karoutsos, V. 2003 Nonequilibrium capillary pressure and relative permeability curves of porous media. *AIChE Journal* **49** (10), 2472–2486.

van Genuchten, M. T. 1980 A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal* **44**, 892–898.

White, M. D. & Oostrom, M. 2006 *STOMP 4.0 User Guide*. PNNL-17782. Pacific Northwest National Laboratory, Richland, Washington.

Widrow, B. 1962 Generalization and information storage in networks of adaline neurons. In: *Self-Organizing Systems* (M. C. Jovitz, G. T. Jacobi & G. Goldstein, eds). Spartan Books, Washington, DC, pp. 435–461.

William III, J. P. 2005 *Introduction to MATLAB for Engineers*. Chapter 5 in *Advanced Plotting and Model Building*. McGraw-Hill, Singapore, International Edition, pp. 259–334.

Yan, S. Q. & Minsker, B. 2006 Optimal groundwater remediation design using an adaptive neural network genetic algorithm. *Water Resources Research* **42**, W05407.

First received 9 September 2011; accepted in revised form 29 August 2012. Available online 16 November 2012