Journal of Petroleum Exploration and Production Technology (2021) 11:1885–1904
https://doi.org/10.1007/s13202-021-01148-8

ORIGINAL PAPER-PRODUCTION ENGINEERING

# Artificial neural network models for reservoir-aquifer dimensionless variables: influx and pressure prediction for water influx calculation

Anietie Ndarake Okon[1] · Idongesit Bassey Ansa[1]

Received: 2 January 2021 / Accepted: 12 March 2021 / Published online: 25 March 2021

## Abstract

Calculation of water influx into petroleum reservoir is a tedious evaluation with significant reservoir engineering applications. The classical approach developed by van Everdingen–Hurst (vEH) based on diffusivity equation solution had been the fulcrum for water influx calculation in both finite and infinite-acting aquifers. The vEH model for edge-water drive reservoirs was modified by Allard and Chen for bottom-water drive reservoirs. Regrettably, these models solution variables: dimensionless influx ($W_{\mathrm{eD}}$) and dimensionless pressure ($P_D$) were presented in tabular form. In most cases, table look-up and interpolation between time entries are necessary to determine these variables, which makes the vEH approach tedious for water influx estimation. In this study, artificial neural network (ANN) models to predict the reservoir-aquifer variables $W_{\mathrm{eD}}$ and $P_D$ was developed based on the vEH datasets for the edge- and bottom-water finite and infinite-acting aquifers. The overall performance of the developed ANN models correlation coefficients ($R$) was 0.99983 and 0.99978 for the edge- and bottom-water finite aquifer, while edge- and bottom-water infinite-acting aquifer was 0.99992 and 0.99997, respectively. With new datasets, the generalization capacities of the developed models were evaluated using statistical tools: coefficient of determination ($R^2$), $R$, mean square error (MSE), root-mean-square error (RMSE) and absolute average relative error (AARE). Comparing the developed finite aquifer models predicted $W_{\mathrm{eD}}$ with Lagrangian interpolation approach resulted in $R^2$, $R$, MSE, RMSE and AARE of 0.9984, 0.9992, 0.3496, 0.5913 and 0.2414 for edge-water drive and 0.9993, 0.9996, 0.1863, 0.4316 and 0.2215 for bottom-water drive. Also, infinite-acting aquifer models (Model-1) resulted in $R^2$, $R$, MSE, RMSE and AARE of 0.9999, 0.9999, 0.5447, 0.7380 and 0.2329 for edge-water drive, while bottom-water drive had 0.9999, 0.9999, 0.2299, 0.4795 and 0.1282. Again, the edge-water infinite-acting model predicted $W_{\mathrm{eD}}$ and Edwardson et al. polynomial estimated $W_{eD}$ resulted in the $R^2$ value of 0.9996, $R$ of 0.9998, MSE of $4.740\times10^{-4}$, RMSE of 0.0218 and AARE of 0.0147. Furthermore, the developed ANN models generalization performance was compared with some models for estimating $P_D$. The results obtained for finite aquifer model showed the statistical measures: $R^2$, $R$, MSE, RMSE and AARE of 0.9985, 0.9993, 0.0125, 0.1117 and 0.0678 with Chatas model and 0.9863, 0.9931, 0.1411, 0.3756 and 0.2310 with Fanchi equation. The infinite-acting aquifer model had 0.9999, 0.9999, 0.1750, 0.0133 and $7.333\times10^{-3}$ with Edwardson et al. polynomial, then 0.9865, 09,933, 0.0143, 0.1194 and 0.0831 with Lee model and 0.9991, 0.9996, $1.079\times10^{-3}$, 0.0328 and 0.0282 with Fanchi model. Therefore, the developed ANN models can predict $W_{\mathrm{eD}}$ and $P_D$ for the various aquifer sizes provided by vEH datasets for water influx calculation.

**Keywords** Dimensionless influx · Dimensionless pressure · Edge- and bottom-water drive reservoirs · Finite and infinite-acting aquifer · Artificial neural network models

## Abbreviations

| | |
|---|---|
| AARE | Absolute average relative error |
| ANN | Artificial neural network |
| CNN | Convolution neural network |
| FFBP | Feed-forward back-propagation |
| FFNN | Feed-forward neural network |
| GRNN | Generalized regression neural network |
| MIMO | Multiple-inputs multiple-outputs |
| MISO | Multiple-inputs single-output |
| MLP | Multilayer perceptron |
| MSE | Mean square error |
| $P_D$ | Dimensionless pressure |
| $P'_D$ | Dimensionless pressure derivative |

✉ Anietie Ndarake Okon
anietieokon@uniuyo.edu.ng

[1] Department of Chemical and Petroleum Engineering, University of Uyo, Uyo, Nigeria

| | |
|---|---|
| $q_D$ | Dimensionless heat flow rate |
| $R$ | Correlation coefficient |
| $R^2$ | Coefficient of determination |
| RBFNN | Radial basis function neural network |
| $r_{eD}$ | Dimensionless radius |
| RMSE | Root-mean-square error |
| RNN | Recurrent neural network |
| SISO | Single-input single-output |
| $t_D$ | Dimensionless time |
| vEH | Van Everdingen-Hurst |
| $W_e$ | Water influx |
| $W_{eD}$ | Dimensionless influx |
| $z_D$ | Dimensionless vertical distance |

## Introduction

Most petroleum reservoirs are underlain by water-bearing formation aquifer, which in most cases provides the reservoir natural energy (drive) source (Okon and Appah 2018). During oil and gas production from the reservoir, the pressure drop depletion, in the reservoir, enables the encroachment of water (i.e., water influx) from the aquifer into the petroleum reservoir (Nashawi and Elkamel 1999). In other words, water influx contributes to the driving force (energy) used for the production of hydrocarbon from the reservoir to the surface (Mustafa et al. 2018). Al-Ghanim et al. (2012) reported that calculation of water influx volumes into the hydrocarbon reservoir is imperative in numerous applications, such as material balance for estimation of reserves, reservoir simulation studies for model calibration, production scheduling and setting up development strategies to optimize hydrocarbon recovery. Because of the mentioned importance of water influx calculation, a reliable model that captures the dynamics of petroleum subsurface system is of essence (Mustafa et al. 2018). In the literature, there are several models for water influx calculation which apply to different flow regimes, namely steady-state (Schilthius 1963), modified steady-state (Hurst 1943; Leung 1986a), unsteady-state (van Everdingen-Hurst 1949; Carter-Tracy 1960) and pseudo-steady-state (Fetkovitch 1971; Leung 1986b). In all the available water influx models, van Everdingen and Hurst (vEH) in 1949 presented the most reliable model for water influx calculation. The reason is that their developed model was the exact solution to the radial flow diffusivity equation, that is, the partial differential equation that describes the fluid flow in porous media. According to Allard and Chen (1988) and Al-Ghanim et al. (2012), the vEH model applies to all flow regimes provided the flow geometry is radial. Also, the model solutions are for both the constant-terminal-pressure and constant-terminal-rate cases of finite and infinite-acting edge-water aquifers. Meaning, the vEH aquifer model was one dimensional and does not cover bottom-water aquifer.

Looking at the gap in vEH solution, Coats (1962) presented a model (diffusivity equation) that considers the upward movement of water from the aquifer into the reservoir, thus, bottom-water drive model. His developed model was two dimensional that provides the solution for only the constant-terminal-rate case of an infinite-acting aquifer. Thus, Coats (1962) model was not applicable to the finite aquifer. Again, using Coats (1962) model, Allard and Chen (1988) modified the model to provide solutions for constant-terminal-pressure case in both finite and infinite-acting aquifers. These classical models for calculating water influx in edge- and bottom-water drive reservoirs are quite useful in reservoir engineering applications (Etim 2019). Regrettably, their established solutions for reservoir-aquifer variables: dimensionless influx ($W_{\mathrm{eD}}$) and dimensionless pressure ($P_D$) for the constant-terminal-rate case and constant-terminal-pressure case, respectively, are in table form. Hence, the use of these models for water influx calculation requires table look-up and interpolation between time entries (Nashawi and Elkamel 1999; Al-Ghanim et al. 2012) to determine the $W_{\mathrm{eD}}$ and $P_D$ for the appropriate aquifer size. In applying these vEH-based models in petroleum reservoir software, the task of executing table look-up and interpolation between time entries requires huge computer efforts (Ansa 2019). Okotie and Ikporo (2019) added that vEH-based approach of calculating water influx requires the principle of superposition which is not straightforward procedure. In this direction, Carter and Tracy (1960) and Fetkovitch (1971) developed approximation models for calculating water influx. Carter-Tracy model handled the tedious process of superposition involved in vEH-based approach. However, the challenge of table look-up and interpolation between time entries remains part of the approach, as they introduced dimensionless pressure ($P_D$) and its derivative ($P'_D$) with respect to dimensionless time ($t_D$) (Nashawi and Elkamel 1999). Allard and Chen (1988) maintained that these vEH-based approximate models are applicable to only a limited range of flow condition or reservoir-aquifer geometries. Considering the wide acceptability of the vEH-based approach, therefore, it is important to have models that are not limited to flow condition rather applicable to all vEH datasets ranges. Hence, the focus of this study is to develop ANN models that cover the vEH datasets for predicting $W_{\mathrm{eD}}$ and $P_D$ variables for calculation of water influx volume into the reservoirs.

### Overview of some existing models for estimating the reservoir-aquifer dimensionless variables: $W_{\mathrm{eD}}$ and $P_D$

Among the available models for calculating water influx into the reservoir, there is no doubt that the vEH-based models, which are the exact solutions of the diffusivity equations, are the most applicable. As reported by Al-Ghanim et al. (2012), these models suffer limitation because their results (i.e., $W_{eD}$ and $P_D$ variables) are in tabular forms, which significantly limits their application in computer analysis and simulation studies. For the constant-terminal-rate in finite-radial and infinite-acting edge-water drive reservoirs, the proposed Chatas (1953), Lee (1982) and Fanchi (1985) equations for estimating $P_D$ from $t_D$ and $r_{eD}$ determine this dimensionless variable. In another development, the polynomials developed by Edwardson et al. (1962) for the approximation of $q_D$, $P_D$ and $P'_D$ as a function of $t_D$ for an infinite-acting aquifer are extended to water influx dimensionless variables determination, as $W_{eD}$ replaces $q_D$ in the equations. Also, Klins et al. (1999) developed some complex polynomials for estimating water influx variables: $W_{eD}$, $P_D$ and $P'_D$ for finite and infinite-acting aquifers that apply to edge-water drive reservoirs. They maintained that these equations represent a traceable replacement to tubular listings of the vEH dimensionless functions. Again, Al-Ghanim et al. (2012) developed nonparametric optimal transformation models for $W_{eD}$ and $P_D$ for edge-water drive reservoirs that are data-driven and do not assume a priori function form as other models do. Regrettably, as reported by Nashawi and Elkamel (1999) and Al-Ghanim et al. (2012), the available models for estimating these water influx dimensionless variables are characterized by drawbacks. For instance, Chatas (1953), Edwardson et al. (1962), Lee (1982), Fanchi (1985) and Klins et al. (1999) models do not apply to the various aquifer sizes (i.e., finite and infinite-acting aquifers) and reservoir drives (edge- and bottom-water drive). Also, these models do not provide values for all the aquifer sizes presented by vEH as their estimation is correlation range limited. Again, Klins et al. (1999) and Al-Ghanim et al. (2012) equations are not easy to implement as they involve some complex transformations and computation effort. In this direction, it is necessary to have a model that will handle the mentioned drawbacks of these existing models. According to Nashawi and Elkamel (1999), intelligent models-neural network models would provide the values of the dimensionless variables for the various reservoir drive and aquifer sizes as presented by vEH. They developed ANN models for predicting $W_{eD}$ and $P_D$ in edge- and bottom-water drive finite and infinite-acting aquifers. Their developed ANN models were multiple-inputs single-output (MISO) except for edge-water drive infinite-acting aquifer model that was single-input single-output (SISO). The performance of these models was evaluated based on their training and testing data points errors (i.e., minimum, maximum and average errors) and compared to the results obtained from Fanchi (1985) and Klins et al. (1999) equations. Unfortunately, these ANN models generalization capacities were not tested with new datasets to establish their application potentials. Also, except for the finite edge-water ANN model, other models' scaled variables were further normalized by taking their natural logarithm. Meaning that the de-normalization of these models predicted outputs would not be straight forward, as they will involve 2-stage de-normalization. Besides, any ANN model predicts values in the range of 0.00001–1.0 effectively, which is not the case in edge- and bottom-water drive infinite-acting aquifers. Therefore, the potential of the Nashawi and Elkamel (1999) ANN models to predict new sets of data is in doubt. Hence, it is imperative to develop ANN models that can handle new datasets to predict $W_{eD}$ and $P_D$ that are comparable with the existing models in this study.

## Overview of artificial neural network (ANN)

According to Zou et al. (2008), the artificial neural network (ANN), often just called a neural network, is a machine learning method evolved from the idea of simulating the human brain. Therefore, ANN is modeled on the concept of biological neural network with ANN as the interconnection nodes or neurons. An ANN consists of several artificial neurons (i.e., nonlinear processing unit) connected through weights (Krenker et al. 2011). Zou et al. (2008) reported that ANN has three major components, namely, node character, network topology and learning rules. The node or neuron character signals processes by the neuron, such as the number of inputs and output, and the activation (transfer) function. The network topology controls the manners neurons are array and linked to the network. Again, the learning rules determine how the weights and biases (threshold) are initialized and adjusted in the network. There are several types of neural networks, namely, feed-forward neural network (FFNN), multilayer perceptron (MLP), generalized regression neural network (GRNN), convolution neural network (CNN), radial basis function neural network (RBFNN), recurrent neural network (RNN), etc. A typical ANN topology or architecture has three layers: input layer, hidden layer and output layer (Jiang et al. 2018; Han et al. 2018). Figure 1 depicts a simplified topology of an ANN, which can be represented mathematically as in Eq. 1 (Anifowose et al. 2012).

$$y = f\left[\sum_{i=1}^{n}\left(x_i W_i + b_i\right)\right] \tag{1}$$

where $x_i$ are the inputs to the neuron, $W_i$ are the weight attached to the inputs to the neuron, $b_i$ is the bias (or threshold), $f$ is the network transfer function, and $y$ is the output of the neuron. Krenker et al. (2011) mentioned that the major unknown variable in Eq. 1 is the transfer function, which is chosen based on the nature of the problem to be solved by the artificial neuron. The various transfer or activation functions available in the literature are linear, nonlinear, piecewise linear, sigmoidal, tangent, hyperbolic and polynomial

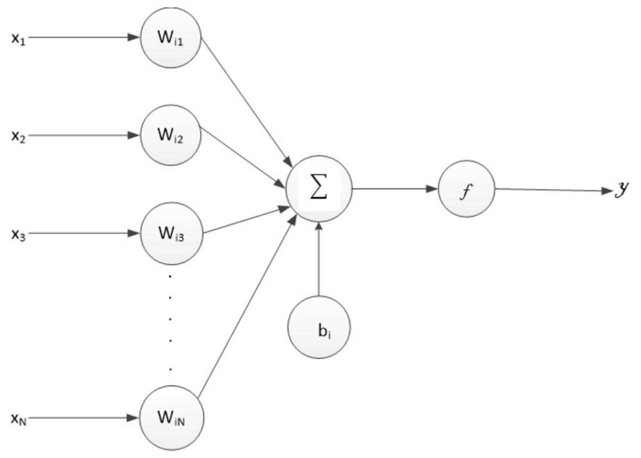


**Fig. 1** A simplified ANN topology (architecture) (Anifowose et al. 2012)

functions (Anifowose et al. 2012). In any case, the most used transfer functions in a neural network are linear function "*purelin*" and nonlinear (sigmoid) function "*tansig*." The sigmoid function is as shown in Eq. 2;

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{2}$$

where $z$ represents the node summed variables and $\sigma(z)$ denotes the transformed node output. The value processed by the sigmoid function is the network node output value. An artificial neural network learns a task by adjusting its weights (Musa and Hamisu 2019). The higher the weight of an artificial neuron is, the stronger the input which is multiplied by it will be. The types of network learning or training are supervised and unsupervised learning approaches. Supervised training requires the output data to learn the target data, while unsupervised learning does not need the output data to predict the target outcome (Krenker et al. 2011). There are several ANN learning algorithms available in the literature. The importance of any training algorithm is to minimize the mean square error (MSE) between the predicted output (target) datasets of the model and the observation outputs datasets used in the network training (Okon et al. 2020). Examples of the available training algorithms include Levenberg–Marquardt, Bayesian regularization, scaled conjugate gradient, Quasi-Newton, etc. Among these ANN learning algorithms, Levenberg–Marquardt algorithm is the most efficient (Konate et al. 2015), as it is faster and has more convergence stability than other learning algorithms (Hagan and Menhaj 1994). So far, ANNs have been applied in numerous fields, like medical, environmental, software engineering, control engineering, etc. In petroleum engineering, the most common type of ANN is the MLP which is train with a feed-forward back-propagation (FFBP) approach (Wood 2019). Some applications of ANN in petroleum industry include prediction of hydrocarbon reserve (Ma and Gomez 2015); reservoir characterization (Long et al. 2016); mud loss treatment (Cristofaro et al. 2017); relative permeability interpolation (Dang et al. 2018); water saturation prediction of sandstone reservoirs (Khan et al. 2018); development of screening tool for $CO_2$ injection in naturally fractured reservoirs (Hammam and Ertekim 2018), among others**.**

## Materials and methods

### Data acquisition and preparation

The edge-water van Everdingen-Hurst (1949) and bottom-water Allard and Chen (1988) dimensionless datasets of time ($t_D$), radius ($r_{\mathrm{eD}}$), vertical distance ($z_D$) and water influx ($W_{\mathrm{eD}}$) required for the finite (bounded) and infinite-acting aquifers were extracted from Ahmed and McKinney (2005). These vEH datasets were based on analytical solution (using Laplace transformation) to the radial diffusivity equation, which assumed there was step change between the reservoir and the aquifer pressure. The constant reservoir-aquifer boundary pressure solution was presented in the form of dimensionless water influx ($W_{\mathrm{eD}}$) as a function of $t_D$ and $r_{\mathrm{eD}}$. Also, the corresponding edge-water dimensionless pressure ($P_D$) for the two reservoir-aquifer configurations: finite (bounded) and infinite-acting, was evaluated using Chatas (1953) and Edwardson et al. (1962) models (Eqs. 3 and 4). In the bounded aquifer, the edge-water dimensionless influx ($W_{\mathrm{eD}}$) and dimensionless pressure ($P_D$) are functions of dimensionless time ($t_D$) and dimensionless radius ($r_{\mathrm{eD}}$) while in the bottom-water, they are functions of $t_D$, $r_{\mathrm{eD}}$ and dimensionless vertical distance ($z_D$). Also, in the infinite-acting aquifer, $W_{\mathrm{eD}}$ and $P_D$ are a function of $t_D$ in edge-water drive and the functions of $t_D$ and $z_D$ in the bottom-water drive reservoir. Regrettably, there is no available empirical or analytical model(s) in the literature for the estimation of $P_D$ in the bottom-water drive reservoir type for bounded and infinite-acting aquifers.

For the finite (bounded) aquifer, the Chatas (1953) model for predicting $P_D$ in edge-water drive reservoir-aquifer configuration is given as:

$$P_D = \frac{0.5 + 2t_D}{r_{\mathrm{eD}}^2 - 1} - \frac{r_{\mathrm{eD}}^4\left[3 - 4\ln\left(r_{\mathrm{eD}}\right)\right] - 2r_{\mathrm{eD}}^2 - 1}{4\left(r_{\mathrm{eD}}^2 - 1\right)^2} \tag{3}$$

For the infinite-acting aquifer, the Edwardson et al. (1962) model for predicting $P_D$ in bottom-water drive reservoir-aquifer configuration is expanded as:

$$P_D = \frac{370.529\sqrt{t_D} + 137.582t_D + 5.69549\left(t_D\right)^{1.5}}{328.834 + 265.488\sqrt{t_D} + 45.2157t_D + \left(t_D\right)^{1.5}} \tag{4}$$

when the $t_D > 100$; thus, $P_D = 0.5\left[\ln\left(t_D\right) + 0.80907\right]$

Therefore, for the bounded aquifer, the edge-water drive consists of 516 datasets with $t_D$ and $r_{eD}$ as input data and $W_{eD}$ and $P_D$ as output variables. The bottom-water drive reservoir type has 1218 data points with $t_D$, $r_{eD}$ and $z_D$ as input data and $W_{eD}$ as output data. Again, for the infinite-acting aquifer, the datasets were 549 for an edge-water drive reservoir with $t_D$ as an input variable and $W_{eD}$ and $P_D$ as output data. The bottom-water drive consists of 4098 datasets with $t_D$ and $z_D$ as input data and $W_{eD}$ as output data. Tables 1 and 2 present the input and output variables' minimum and maximum values and the statistical description of these variables' values for the various reservoir-aquifer configurations. As observed in Table 2, especially for the infinite-acting aquifer, the differences between the variables maximum and minimum values (i.e., range) are large, and these values will affect the network training process if not scaled down. Hence, the input and output variables were normalized to 0–1 using the maximum–minimum normalization equation (Eq. 5). This approach ensures that the neural network training algorithm will adjust the network weights and biases adequately. Again, scaling the input and output data to 0–1 reduces the sensitivity of the neural networks' sigmoidal (i.e., activation) function to large data values (Okon et al. 2020).

$$y_{\text{normalized}} = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \tag{5}$$

**Table 1** Minimum and maximum values of the input and output variables for ANN models development

| Reservoir drive type | Datasets | | Input data | | | Input data | |
|---|---|---|---|---|---|---|---|
| | | Bounded reservoir | | | | | |
| | | | Min | Max | | Min | Max |
| Edge-water drive | 516 | $t_D$ | 0.05 | 500 | $W_{eD}$ | 0.276 | 49.36 |
| | | $r_{eD}$ | 1.5 | 10.0 | $P_D$ | 0.2438 | 22.101 |
| Bottom-water drive | 1218 | $t_D$ | 2.0 | 1200 | | | |
| | | $r_{eD}$ | 4.0 | 10.0 | $W_{eD}$ | 2.398 | 49.96 |
| | | $z_D$ | 0.05 | 1.0 | | | |
| | | Infinite-acting reservoir | | | | | |
| Edge-water drive | 549 | $t_D$ | 0.0 | $2.0\times10^{12}$ | $W_{eD}$ | 0.0 | $1.42\times10^{11}$ |
| | | | | | $P_D$ | 0.0 | 14.5666 |
| Bottom-water drive | 4098 | $t_D$ | 0.10 | $2.0\times10^{12}$ | $W_{eD}$ | 0.176 | $1.4\times10^{11}$ |
| | | $z_D$ | 0.5 | 1.0 | | | |

**Table 2** Statistical description of the input and output variables used for the ANN models development

| Reservoir type | Aquifer geometry | Statistical tool | Input and Output Variables | | | | |
|---|---|---|---|---|---|---|---|
| | | | $t_D$ | $r_{eD}$ | $z_D$ | $W_{eD}$ | $P_D$ |
| Bounded Reservoir | Edge-water drive | $\bar{x}$ | 41.4825 | 5.1124 | – | 10.4993 | 2.9211 |
| | | SD | 79.5594 | 2.6406 | – | 10.9456 | 2.7628 |
| | | Range | 499.95 | 8.50 | – | 49.084 | 21.8570 |
| | Bottom-water drive | $\bar{x}$ | 136.2644 | 7.2989 | 0.5071 | 18.8677 | – |
| | | SD | 202.9565 | 2.2160 | 0.3489 | 13.2108 | – |
| | | Range | 1198 | 6.0 | 0.95 | 48.453 | – |
| Infinite-acting Reservoir | Edge-water drive | $\bar{x}$ | $1.81\times10^{10}$ | – | – | $1.34\times10^{9}$ | 4.4650 |
| | | SD | $1.35\times10^{11}$ | – | – | $9.78\times10^{9}$ | 2.8416 |
| | | Range | $2.0\times10^{12}$ | – | – | $1.42\times10^{11}$ | 14.5666 |
| | Bottom-water drive | $\bar{x}$ | $1.89\times10^{11}$ | – | 0.5071 | $1.35\times10^{9}$ | – |
| | | SD | $1.38\times10^{11}$ | – | 0.3490 | $9.69\times10^{9}$ | – |
| | | Range | $2.0\times10^{12}$ | – | 0.95 | $1.4\times10^{11}$ | – |

$\bar{x}$: mean; SD: Standard deviation

where $y_{\text{normalized}}$ is the normalized input or output variable,$y$ is the actual variable value, and $y_{\min}$ and $y_{\max}$ are the minima and maxima values of the variables, respectively. It is worth mentioning that after normalizing the infinite-acting aquifer edge-water and bottom-water datasets, the scaled $t_D$ and $W_{\text{eD}}$ datasets were in the range of $1.0\times10^{-12}$–1.0. These extreme values would affect the generalization capacity of the ANN models predictions. In this case, the edge-water and bottom-water infinite-acting aquifer datasets were grouped into five sets, and then scaled based on the minimum and maximum values in Table 3 to reduce these datasets range.

## Artificial neural network development

The artificial neural network (ANN) models for predicting the dimensionless influx ($W_{\text{eD}}$) and the dimensionless pressure ($P_D$) of finite (bounded) and infinite-acting aquifers were developed using the neural fitting tool (nftool) in MATLAB (Matrix Laboratory) 2015a mathematical software. The normalized input (i.e., $t_D$, $r_{\text{eD}}$ and $z_D$) and output (i.e.,$W_{\text{eD}}$ and $P_D$) datasets for the various reservoir-aquifer configurations were exported to the MATLAB nftool environment and randomly divided into three parts, namely, training set (70%), validation set (15%) and testing set (15%), as shown in Table 4. The basic settings of the MATLAB software used to train and developed the ANN models are visible in Table 5. The neural network training was supervised learning as the target data ($W_{\text{eD}}$ and $P_D$) were provided to the network as outputs. In the training phase, the network learning algorithm: Levenberg–Marquardt (*trainlm* in MATLAB which takes less time) adjusts the network neurons' weights and biases until the neural network learns the appropriate input–output pattern of the datasets (Okon et al. 2020). Again, during the training stage, the predictions performance of the neural network was assessed using the validation and test datasets. Then, the network training truncated based on the stopping criteria, that is, mean square error (MSE) and the number of the epoch (iteration). The MATLAB software default values for these stopping criteria are given in Table 5. In this software, the training automatically stopped when

**Table 3** Ranges of the grouped edge- and bottom-water drive infinite-acting aquifer variables

| Reservoir-aquifer configurations | Dataset group | $t_D$ | | $W_{eD}$ | | $P_D$ | |
|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max |
| Infinite-acting Edge-water | SET-1 | 0 | 1000 | 0 | 293.514 | 0 | 3.8584 |
| | SET-2 | 1000 | $1.0\times10^{5}$ | 293.514 | 17,586.28 | 3.8584 | 6.1610 |
| | SET-3 | $1.0\times10^{5}$ | $1.0\times10^{8}$ | 17,586.28 | $1.095\times10^{7}$ | 6.1610 | 9.6149 |
| | SET-4 | $1.0\times10^{8}$ | $2.0\times10^{10}$ | $1.095\times10^{7}$ | $8.75\times10^{8}$ | 9.6149 | 11.9175 |
| | SET-5 | $1.0\times10^{10}$ | $2.0\times10^{12}$ | $8.75\times10^{8}$ | $1.42\times10^{11}$ | 11.9175 | 14.5666 |
| | | $t_D$ | | $z_D$ | | $W_{eD}$ | |
| Infinite-acting Bottom-water | SET-1 | 0 | 1000 | 0.5 | 1.0 | 0.176 | 274 |
| | SET-2 | 1000 | $1.0\times10^{5}$ | 0.5 | 1.0 | 240 | 16,870 |
| | SET-3 | $1.0\times10^{5}$ | $1.0\times10^{8}$ | 0.5 | 1.0 | 3920 | $1.07\times10^{7}$ |
| | SET-4 | $1.0\times10^{8}$ | $2.0\times10^{10}$ | 0.5 | 1.0 | $1.01\times10^{7}$ | $8.58\times10^{8}$ |
| | SET-5 | $1.0\times10^{10}$ | $2.0\times10^{12}$ | 0.5 | 1.0 | $1.61\times10^{8}$ | $1.40\times10^{11}$ |

**Table 4** Training, validation and testing datasets of the finite and infinite-acting aquifers

| Reservoir-aquifer geometry | | Datasets | | Training | Validation | Testing |
|---|---|---|---|---|---|---|
| Finite aquifer | Edge-water | 516 | | 362 | 77 | 77 |
| | Bottom-water | 1218 | | 852 | 183 | 183 |
| Infinite-acting aquifer | Edge-water | 549 | Model-1 | 178 | 38 | 38 |
| | | | Model-2 | 151 | 32 | 32 |
| | | | Model-3 | 24 | 5 | 5 |
| | | | Model-4 | 16 | 3 | 3 |
| | | | Model-5 | 16 | 4 | 4 |
| | Bottom-water | 4098 | Model-1 | 1171 | 251 | 251 |
| | | | Model-2 | 1293 | 277 | 277 |
| | | | Model-3 | 170 | 37 | 37 |
| | | | Model-4 | 112 | 24 | 24 |
| | | | Model-5 | 122 | 26 | 26 |

**Table 5** The basic ANN models training parameters

| Parameters | | Values |
|---|---|---|
| Number of input neurons, Finite (Bounded) aquifer | Edge-water | 2 ($t_D$, $r_{eD}$) |
| | Bottom-water | 3 ($t_D$, $r_{eD}$, $z_D$) |
| Number of input neurons, Infinite-acting aquifer | Edge-water | 1 ($t_D$) |
| | Bottom-water | 2 ($t_D$, $z_D$) |
| Number of hidden layer | | 1 |
| Number of neurons in hidden layer, Finite (Bounded) aquifer | Edge-water | 10 |
| | Bottom-water | 6 |
| Number of neurons in hidden layer, Infinite-acting aquifer | Edge-water | 5 |
| | Bottom-water | 8 |
| Number of output neuron, Finite (Bounded) aquifer | Edge-water | 2 ($W_{eD}$, $P_D$) |
| | Bottom-water | 1 ($W_{eD}$) |
| Number of output neuron, Infinite-acting aquifer | Edge-water | 2 ($W_{eD}$, $P_D$) |
| | Bottom-water | 1 ($W_{eD}$) |
| Input activation function | | *tansig* |
| Output activation function | | *purelin* |
| Learning algorithm (*trainlm*) | | Levenberg–Marquardt |
| Mean square error (MSE) | | 1.0e−05 |
| Number of epochs | | 1000 |
| Training rate | | 0.7 |

the validation dataset generalization (i.e., MSE value) stops improving. According to Okon et al. (2020), the weights and biases that yield the lowest error from the supervised datasets are the network best generalization. Also, this least MSE and correlation coefficient (R) values were the basis for selecting the best ANN architecture. In other words, a trial-and-error approach was used to achieve the number of neurons in the network hidden layer. Hence, the ANN architecture training was several times repeated to ensure that its random weights initialization did not result in the network's inconsistent performance correlation.

# Results and discussion

## Performance of the ANN models

### Finite (Bounded) aquifer

The developed ANN models for predicting the dimensionless influx ($W_{\mathrm{eD}}$) and dimensionless pressure ($P_D$) of edge-water and bottom-water drive finite (bounded) aquifer are three-layer feed-forward back-propagation networks. These developed ANN models architectures are established based on their best generalization performance with the training, validation and testing datasets. Table 6 shows the models architecture as well as their performance indexes, namely, mean square error (MSE) and correlation coefficient ($R$) values, for this reservoir-aquifer configuration. From this table, the ANN model for the edge-water drive has two neurons at the input layer, ten neurons at the hidden layer and two neurons at the output layer, thus, represented as 2–10–2. Also, the bottom-water drive ANN model has three neurons at the input layer, six neurons at the hidden layer and one neuron at the output layer (i.e., 3–6–1). Again, the developed ANN models performance plots that showed the training, validation, testing and overall predictions of these models to the edge-water and bottom-water drives datasets are visible in Figs. 2 and 3, respectively. For the edge-water drive ANN model, its predictions were very close to the actual $W_{\mathrm{eD}}$ and $P_D$ datasets. This assertion is because of the MSE and $R$ values obtained (Table 6) and the diagonal alignment of the output and target from the training, validation and testing datasets (Fig. 2). Besides, the overall performance of the edge-water ANN model prediction of $W_{\mathrm{eD}}$ and $P_D$ data resulted in an $R$-value of 0.99983. This

**Table 6** Training, validation and testing performance of the finite aquifer ANN models

| Reservoir drive | ANN topology | Training | | Validation | | Testing | |
|---|---|---|---|---|---|---|---|
| | | MSE | $R$ | MSE | $R$ | MSE | $R$ |
| Edge-water | 2–10–2 | 1.030e−5 | 0.99985 | 1.532e−5 | 0.99979 | 1.388e−5 | 0.99983 |
| Bottom-water | 3–6–1 | 3.202e−5 | 0.99978 | 3.791e−5 | 0.99975 | 2.723e−5 | 0.99983 |

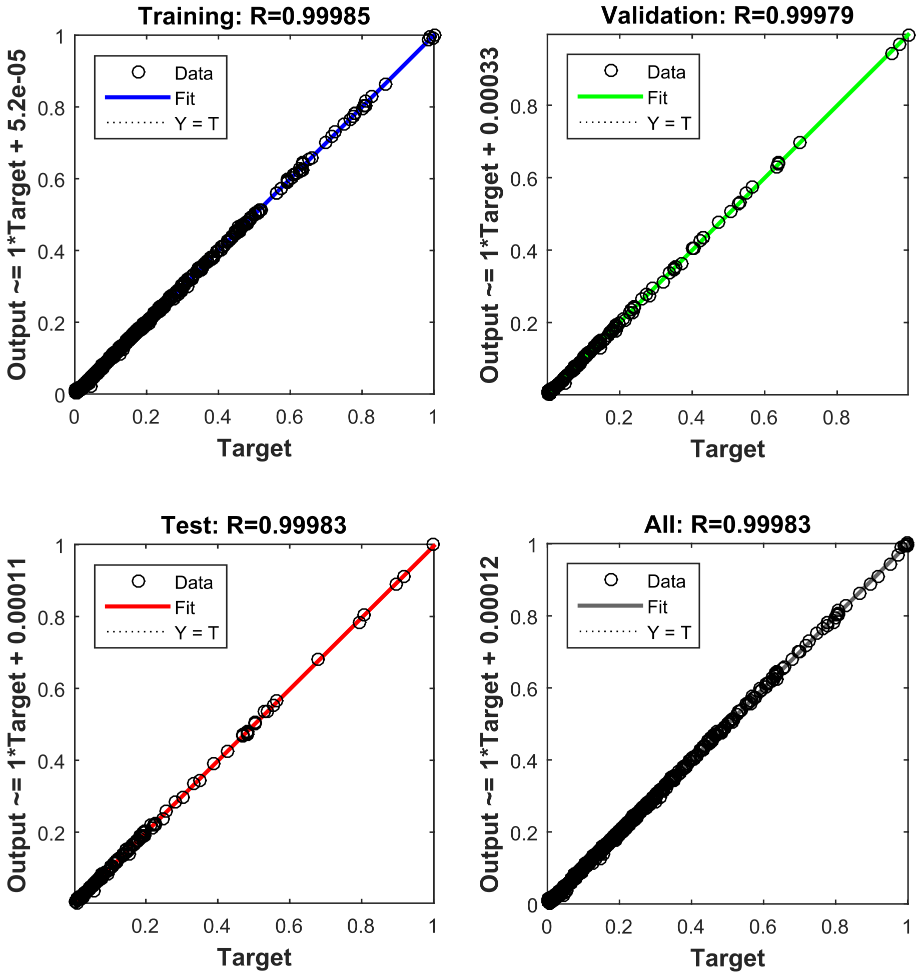


**Fig. 2** Edge-water drive finite aquifer ANN model performance plot

$R$-value further supported that the model predictions fit very close to the actual $W_{\mathrm{eD}}$ and $P_D$ data for the edge-water drive finite aquifer. Just as in the developed edge-water model, the performance plots of the bottom-water drive ANN model as depicted in Fig. 3 implied that the neural network predictions were near-exact values of the $W_{\mathrm{eD}}$ datasets. The efficient predictions of this ANN model are visible in the MSE and $R$ values obtained (Table 6), as well as the diagonal trend of the output-target datasets from the training, validation and testing in Fig. 3. Also, the robust predictions of this ANN model are visible in its overall performance $R$-value of 0.99978. Therefore, these empirical ANN models for the edge-water and bottom-water $W_{\mathrm{eD}}$ and $P_D$ predictions based on the Levenberg–Marquardt learning algorithm are expanded in Eqs. 6 and 7. These equations predict the $W_{\mathrm{eD}}$ and $P_D$ in the normalized forms which would be de-normalized using Eq. 8;

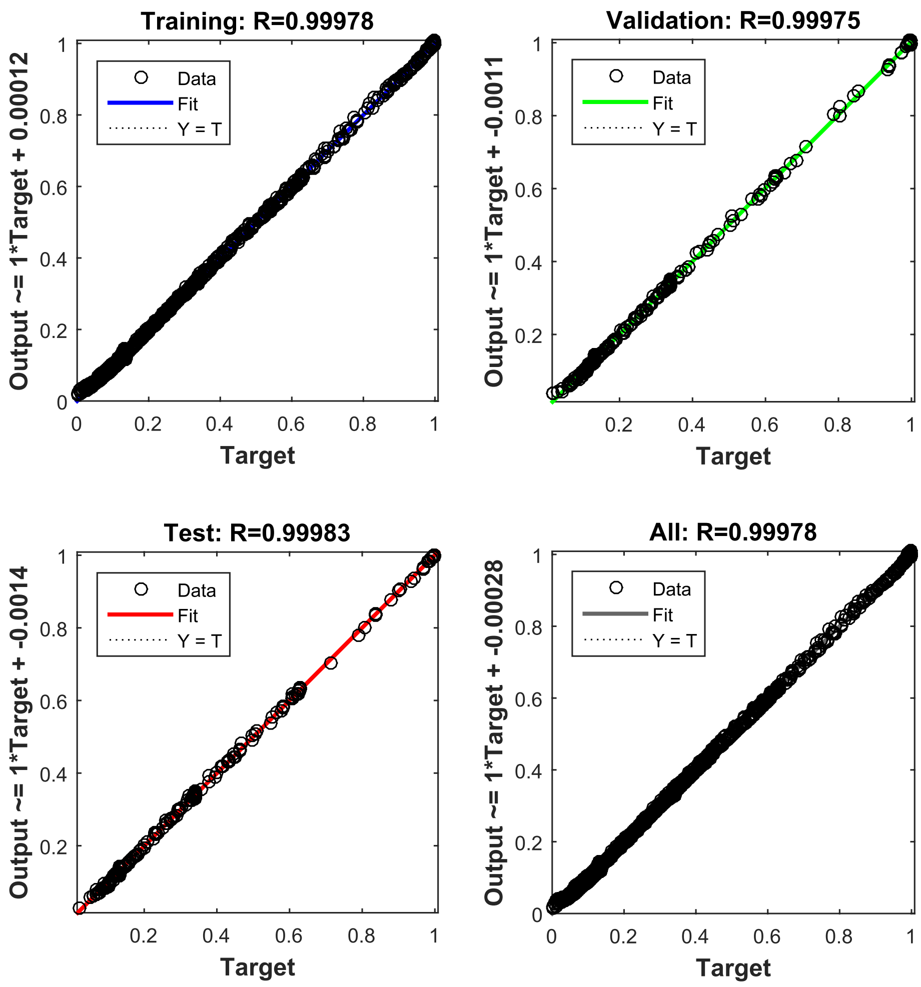
Edge-water drive ANN model for finite aquifer:

$$(W_{\mathrm{eD}}, P_D) = \sum_{j=1}^{2}\left[purelin\left\{\sum_{i=1}^{10}\sum_{j=1}^{2} tansig\left[\left(t_D j_1 + r_{\mathrm{eD}} j_2\right)_i + b_i\right]\right\} \times Lw_{i,j} + b_{k_i}\right] \tag{6}$$

Bottom-water drive ANN model for finite aquifer:

$$(W_{\mathrm{eD}}) = \sum_{j=1}^{1}\left[purelin\left\{\sum_{i=1}^{6}\sum_{j=1}^{3} tansig\left[\left(t_D j_1 + r_{\mathrm{eD}} j_2 + z_D j_3\right)_i + b_i\right]\right\} \times Lw_{i,j} + b_{k_i}\right] \tag{7}$$

**Fig. 3** Bottom-water drive finite aquifer ANN model performance plot



$$\left(W_{eD}, P_D\right)_{\text{denorm.}} = \left(W_{eD}, P_D\right)_{\text{Min.}} + \left(W_{eD}, P_D\right)\left[\left(W_{eD}, P_D\right)_{\text{Max.}} - \left(W_{eD}, P_D\right)_{\text{Min.}}\right] \tag{8}$$

where $\left(W_{eD}, P_D\right)_{\text{denorm.}}$ would be the de-normalized $W_{\text{eD}}$ or $P_D$ value(s) predicted by the developed ANN models, $\left(W_{eD}, P_D\right)_{\text{Min.}}$ and $\left(W_{eD}, P_D\right)_{\text{Max.}}$ are the actual minimum and maximum values of $W_{\text{eD}}$ or $P_D$, and $\left(W_{\text{eD}}, P_D\right)$ is the normalized value(s) of $W_{\text{eD}}$ or $P_D$ predicted by the ANN models.

**Table 7** Weights and biases of the developed edge-water ANN model

| $i$ | Input layer weights | | Input biases | Hidden layer weight | | Input biases |
|---|---|---|---|---|---|---|
| | $j_1(t_D)$ | $j_2(r_{eD})$ | $b_1$ | $Lw_1$ | $Lw_2$ | $b_k$ |
| 1 | −0.93605 | −1.0019441 | 1.893821 | −0.81513 | 0.05756 | −35.343549 |
| 2 | 2.337696 | −2.8786690 | −2.3113 | 11.52446 | −15.8817 | −12.135393 |
| 3 | 1.788543 | −1.0376910 | 0.760595 | 19.23305 | −3.96886 | |
| 4 | 1.44388 | −0.0278711 | 3.086581 | 27.1473 | 2.855759 | |
| 5 | −9.82893 | −6.2658930 | 0.471504 | −0.01458 | −0.0079 | |
| 6 | 1.759516 | −1.0668261 | 0.788837 | −19.7503 | 4.30937 | |
| 7 | −0.62423 | −1.1124574 | −1.33701 | 1.719126 | −0.58229 | |
| 8 | 1.623027 | −2.3682539 | −2.33772 | −37.9502 | 54.98914 | |
| 9 | −24.4916 | 0.3508711 | −26.7682 | −13.9546 | 0.521742 | |
| 10 | 1.397001 | 2.1866949 | 5.217839 | −29.0559 | 47.95144 | |

**Table 8** Weights and biases of the developed bottom-water ANN model

| $i$ | Input layer weights | | | Input biases | Hidden layer weight | Input bias |
|---|---|---|---|---|---|---|
| | $j_1(t_D)$ | $j_2(r_{eD})$ | $j_2(z_D)$ | $b_1$ | $Lw_1$ | $b_k$ |
| 1 | 1.508636 | 1.11969 | −0.18959 | −0.97293 | 2.243376 | −2.97751 |
| 2 | 1.66491 | 1.005135 | −0.21951 | −0.93007 | −1.95284 | |
| 3 | 6.89277 | −0.45576 | −0.08777 | 6.599614 | 4.996322 | |
| 4 | −0.07996 | 0.626797 | −0.00431 | 0.091816 | 1.214238 | |
| 5 | −6.82805 | 0.165655 | −0.14067 | −7.9447 | −2.29222 | |
| 6 | 7.287317 | −0.40061 | −0.07584 | 6.866409 | −4.23918 | |

Equations 6 and 7 are the trained artificial neural network (ANN) models for predicting $W_{eD}$ and $P_D$ of edge-water and bottom-water finite aquifers. In these equations, the variables $j_1$, $j_2$ and $j_3$ are input weights connected to the inputs data $t_D$, $r_{\mathrm{eD}}$ and $z_D$, respectively, from the input layer to the hidden layer neurons. The variable $Lw_{i,j}$ is hidden layer weights attached to the output layer neuron(s), as the variables $b_i$ and $b_k$ are biases connected to the hidden layer neurons and output layer neuron(s), respectively. These variables values for the developed edge-water and bottom-water finite aquifer ANN models are as shown in Tables 7 and 8, respectively. Also, in these equations, the activation function '*tansig*' transformed the combined inputs ($t_D$, $r_{\mathrm{eD}}$ and $z_D$), weights ($j_1$, $j_2$ and $j_3$) and bias ($b_i$), that is, output(s) of the hidden layer neurons to the output layer neuron(s) and the '*purelin*' function converts the estimated outcome of the output layer neuron(s) to established the linear relationship between the input and output datasets.

## Infinite-acting aquifer

The developed ANN models for edge-water and bottom-water infinite-acting aquifers are also a three-layer feed-forward back-propagation (FFBP) neural network. As earlier mentioned that because of the large values of $t_D$ and $W_{\mathrm{eD}}$ datasets of the edge-water and bottom-water infinite-acting aquifer, the data were grouped into five sets. Hence, five ANN models, as indicated in Table 9, were trained to handle the predictions of $W_{\mathrm{eD}}$ and $P_D$ for this reservoir-aquifer geometry. The developed edge-water drive ANN models have one neuron at the input layer, five neurons at the hidden layer and two neurons at the output layer (i.e., 1–5–2), while the bottom-water drive models have two neurons at the input layer, 8 neurons at the hidden layer and one neuron at the output layer (i.e., 2–8–1). Table 9 depicts the training, validation and testing performance, that is, MSE and $R$ values of the developed models. From these MSE and R values and the performance plots obtained for the various ANN models, they showed that the models' predictions were close to the actual $W_{\mathrm{eD}}$ and $P_D$ datasets. Figures 4 and 5 are the plots showing the training, validation, testing and overall performance of the edge-water ANN Model-1 and bottom-water ANN Model-1, respectively. Interestingly, the similar trends were obtained for other models (i.e., model-2–5). Therefore, the overall performance of the five ANN models for the edge-water infinite-acting aquifer resulted in $R$ values of 0.99994, 1.0, 0.99995, 0.99999 and 0.99997, while that of bottom-water resulted in $R$ values of 1.0, 1.0, 1.0, 1.0 and 0.99959. These MSE and $R$ values obtained, and the output-target diagonal trends of the ANN models overall performance implied that the developed models would handle the predictions of $W_{\mathrm{eD}}$ and $P_D$ for edge-water and bottom-water infinite-acting aquifers. The mathematical

**Table 9** Training, validation and testing performance of the infinite-acting aquifer ANN models

| Reservoir drive | ANN topology | ANN model | Training | | Validation | | Testing | |
|---|---|---|---|---|---|---|---|---|
| | | | MSE | $R$ | MSE | $R$ | MSE | $R$ |
| Edge-water | 1–5–2 | Model-1 | 7.782e–6 | 0.99996 | 3.607e–5 | 0.99984 | 7.267e–6 | 0.99997 |
| | | Model-2 | 1.047e–7 | 0.99999 | 3.846e–8 | 0.99999 | 8.211e–8 | 0.99999 |
| | | Model-3 | 1.091e–5 | 0.99996 | 1.305e–5 | 0.99995 | 1.682e–5 | 0.99991 |
| | | Model-4 | 8.780e–8 | 0.99999 | 2.082e–7 | 0.99999 | 2.078e–5 | 0.99974 |
| | | Model-5 | 1.539e–7 | 0.99999 | 3.432e–5 | 0.99964 | 3.481e–5 | 0.99998 |
| Bottom-water | 2–8–1 | Model-1 | 5.041e–7 | 0.99999 | 4.980e–7 | 0.99999 | 4.783e–7 | 0.99999 |
| | | Model-2 | 1.957e–8 | 0.99999 | 5.123e–8 | 0.99999 | 6.013e–8 | 0.99999 |
| | | Model-3 | 3.650e–7 | 0.99999 | 2.021e–7 | 0.99999 | 1.605e–6 | 0.99999 |
| | | Model-4 | 5.143e–7 | 0.99999 | 3.786e–7 | 0.99999 | 6.673e–7 | 0.99999 |
| | | Model-5 | 5.835e–5 | 0.99943 | 5.678e–5 | 0.99974 | 1.276e–5 | 0.99991 |

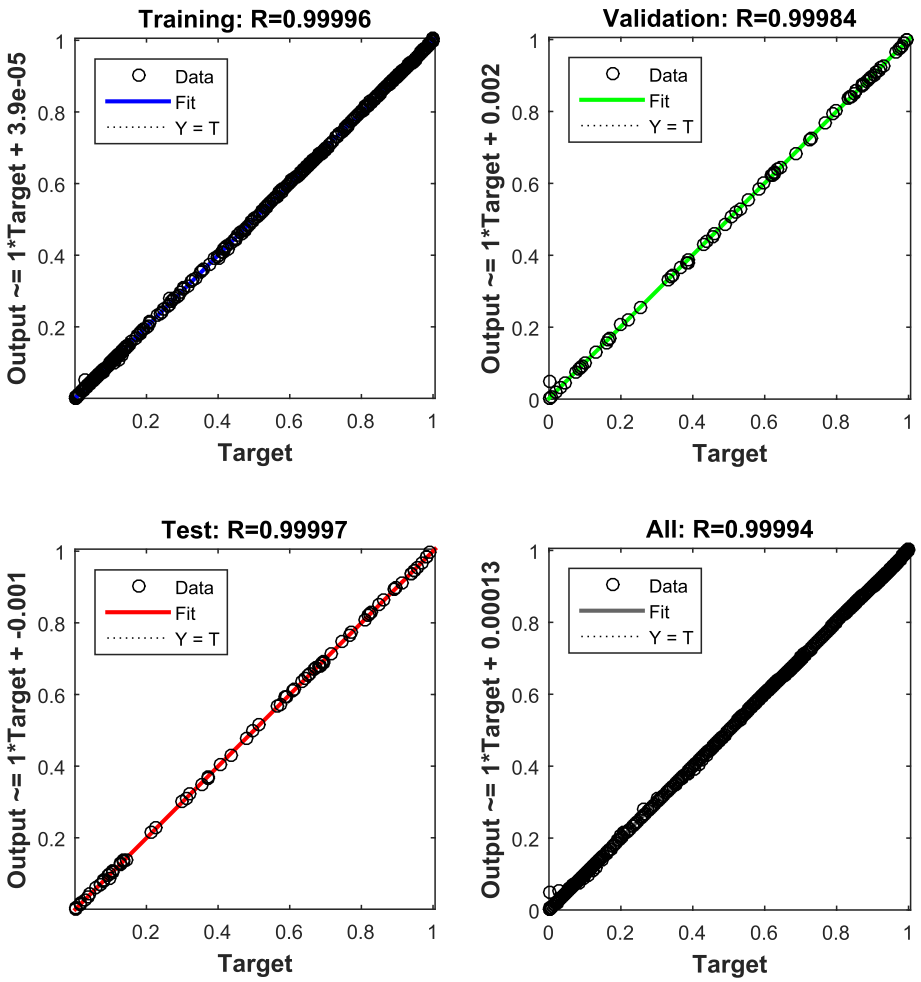


**Fig. 4** Edge-water drive infinite-acting aquifer ANN model (Model-1) performance plot

equations of the developed ANN models for edge-water and bottom-water infinite-acting aquifers are as shown in Eqs. 9 and 10. The variables $j_1$, $j_2$, $j_3$, $Lw_{i,j}$, $b_i$, $b_k$, and activation functions: *tansig* and *purelin*, in these equations are the same as described for Eqs. 6 and 7. Tables 10, 11, 12, 13 and 14 depict the values of these variables in the developed edge-water infinite-acting aquifer models, while Tables 15, 16, 17, 18 and 19 present the values for the bottom-water infinite-acting aquifer models.

Edge-water drive ANN model for infinite-acting aquifer:

$$(W_{\mathrm{eD}}, P_D) = \sum_{j=1}^{2}\left[purelin\left\{\sum_{i=1}^{5}\sum_{j=1}^{1} tansig\left[\left(t_D j_1\right)_i + b_i\right]\right\} \times Lw_{i,j} + b_{k_i}\right] \tag{9}$$

Bottom-water drive ANN model for infinite-acting aquifer:

$$\left(W_{\mathrm{eD}}\right) = \sum_{j=1}\left[purelin\left\{\sum_{i=1}^{8}\sum_{j=1}^{2} tansig\left[\left(t_D j_1 + z_D j_2\right) + b_i\right]\right\} \times Lw_{i,j} + b_k\right] \tag{10}$$

In all the developed empirical ANN models, the general steps of the models to predict $W_{\mathrm{eD}}$ and $P_D$ for the finite (bounded) and infinite-acting aquifers require providing the model(s) input parameters: $t_D$, $r_{\mathrm{eD}}$ and $z_D$, values. For example, in the developed edge-water finite aquifer ANN model to predict $W_{\mathrm{eD}}$ and $P_D$, the input parameters (i.e., $t_D$ and $r_{\mathrm{eD}}$) values provided to the input layer neurons will multiply with their input layer weights: $j_1$ and $j_2$, respectively. In the first neuron of the hidden layer, the sum of these inputs to the neuron and its bias ($b_i$), that is, $\sum_{i=1}\sum_{j=1}^{2}\left(t_D j_1 + r_{eD} j_2\right)_1 + b_1$ is estimated using the values for $i = 1$ in Table 7, where $j_1 = -0.93605$, $j_2 = -1.0019441$ and $b_1 = 1.893821$, and their

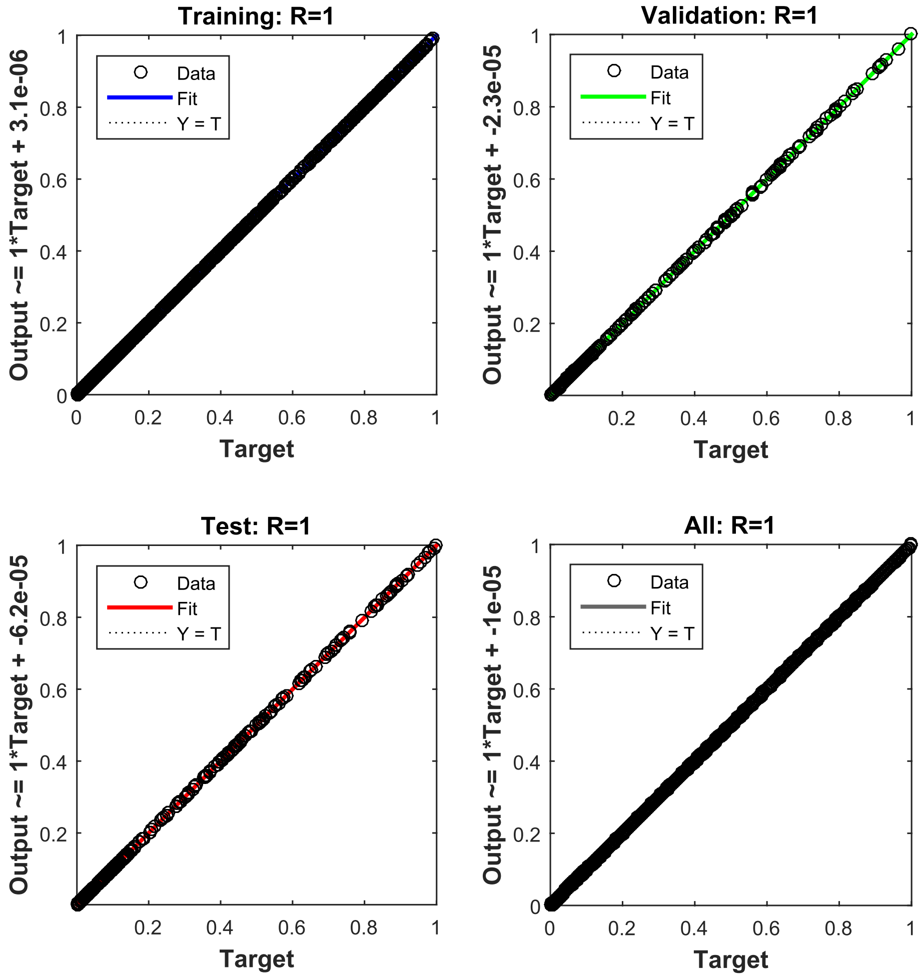


**Fig. 5** Bottom-water drive infinite-acting aquifer ANN model (Model-1) performance plot

**Table 10** Weights and biases of the developed edge-water ANN Model-1

| | Input weights | Input biases | Hidden layer weight | | Output biases |
|---|---|---|---|---|---|
| $i$ | $j_1(t_D)$ | $b_1$ | $Lw_1$ | $Lw_2$ | $b_k$ |
| 1 | 0.592815 | − 11.3895 | 8.110947 | 64.87528428 | − 5.72214 |
| 2 | − 0.20069 | − 0.19554 | − 4.82257 | − 1.00243202 | − 74.0469 |
| 3 | 2.038073 | 4.586331 | 10.00812 | 48.78773477 | |
| 4 | 19.03442 | 21.65537 | 2.30217 | 50.41387002 | |
| 5 | 315.6025 | 318.244 | 0.701886 | 40.35465944 | |

**Table 11** Weights and biases of the developed edge-water ANN Model-2

| $i$ | Input weights | Input biases | Hidden layer weight | | Output biases |
|---|---|---|---|---|---|
| | $j_1(t_D)$ | $b_1$ | $Lw_1$ | $Lw_2$ | $b_k$ |
| 1 | 0.419221 | − 0.52614 | 1.829332 | 0.180915 | − 0.38807 |
| 2 | 0.563567 | 0.835056 | 1.251269 | 1.090775 | − 12.2435 |
| 3 | − 2.18135 | − 3.32332 | − 0.34596 | − 3.39457 | |
| 4 | 8.870327 | 10.15374 | 0.017444 | 2.900897 | |
| 5 | 32.41138 | 34.54604 | 0.111826 | 6.001599 | |

**Table 12** Weights and biases of the developed edge-water ANN Model-3

| | Input weights | Input biases | Hidden layer weight | | Output biases |
|---|---|---|---|---|---|
| $i$ | $j_1(t_D)$ | $b_1$ | $Lw_1$ | $Lw_2$ | $b_k$ |
| 1 | − 6.87383 | 5.630295 | − 0.01146 | 0.008828 | − 0.30312 |
| 2 | 0.326209 | 0.019295 | 2.994527 | 0.634385 | − 15.2071 |
| 3 | − 1.86067 | − 2.62919 | − 0.24116 | − 1.72483 | |
| 4 | 14.97676 | 14.80108 | 0.003235 | 0.357384 | |
| 5 | 71.56132 | 73.50562 | 0.053277 | 13.92153 | |

appropriate $t_D$ and $r_{\mathrm{eD}}$ values. This summation is transformed by the transfer function '*tansig*' and connected to the output layer neurons. Again, the outcome from the first neuron in the hidden layer, $\sum_{i=1} \sum_{j=1}^{2} \tan sig\left(t_D j_1 + r_{eD} j_2\right)_1 + b_1$ is multiplied with the hidden layer weights ($Lw_{i,j}$) and connected to the output neurons. Thus, this input $\sum_{i=1} \sum_{j=1}^{2} \left[\tan sig\left(t_D j_1 + r_{eD} j_2\right)_1 + b_1\right] \times Lw_{i,j}$ combined with the output biases ($b_{k_i}$) in the output neurons. The outlined computations procedures are followed by other neurons (i.e., $i = 2, 3 \ldots 10$) in the hidden layer using their corresponding input weights ($j_1$, $j_2$) and biases ($b_i$) from the input layer neurons, and weights ($Lw_{i,j}$) to link with the output layer neurons. The total summations in the output layer neurons, that is, $\sum_{j=1}^{2} \left[purelin\left\{ \sum_{i=1}^{10} \sum_{j=1}^{2} tansig\left[\left(t_D j_1 + r_{eD} j_2\right)_i + b_i\right] \right\} \times Lw_{i,j} + b_{k_i}\right]$ $\sum_{j=1}^{2} \left[purelin\left\{ \sum_{i=1}^{10} \sum_{j=1}^{2} tansig\left[\left(t_D j_1 + r_{eD} j_2\right)_i + b_i\right] \right\} \times Lw_{i,j} + b_{k_i}\right]$ are transformed by the output layer transfer function '*purelin*' and presented as the ANN model prediction(s) in the normalized form. Then, the ANN model predicted outputs (i.e., $W_{\mathrm{eD}}$ and $P_D$) would be de-normalized using Eq. 8.

**Table 13** Weights and biases of the developed edge-water ANN Model-4

| | Input weights | Input biases | Hidden layer weight | | Output biases |
|---|---|---|---|---|---|
| $i$ | $j_1(t_D)$ | $b_1$ | $Lw_1$ | $Lw_2$ | $b_k$ |
| 1 | − 5.34342 | 7.616427 | − 0.19361 | 0.198985 | 0.31164 |
| 2 | 0.28226 | − 0.11831 | 3.390538 | 0.775696 | − 5.842 |
| 3 | 1.180184 | 1.657614 | 0.270743 | 1.307581 | |
| 4 | 5.204942 | 5.433135 | 0.003435 | 0.585594 | |
| 5 | − 17.3724 | − 18.9369 | − 0.05467 | −4.63686 | |

**Table 14** Weights and biases of the developed edge-water ANN Model-5

| $i$ | Input weights | Input biases | Hidden layer weight | | Output biases |
|---|---|---|---|---|---|
| | $j_1(t_D)$ | $b_1$ | $Lw_1$ | $Lw_2$ | $b_k$ |
| 1 | 3.052444 | − 10.8808 | 1.248138 | 11.25223 | 0.13485 |
| 2 | 0.344166 | − 0.14649 | 2.79575 | 0.520675 | − 11.1293 |
| 3 | − 1.06562 | − 2.45062 | − 1.62776 | − 6.29203 | |
| 4 | − 5.51924 | − 7.24091 | 0.163029 | − 8.44282 | |
| 5 | − 26.1864 | − 28.0842 | − 0.10585 | − 8.5565 | |

**Table 15** Weights and biases of the developed bottom-water ANN Model-1

| $i$ | Input weights | Input biases | Hidden layer weight | | Output bias |
|---|---|---|---|---|---|
| | $j_1(t_D)$ | $j_2(z_D)$ | $b_1$ | $Lw_1$ | $b_k$ |
| 1 | 0.921793 | 2.558332 | − 3.35425 | − 0.05011644 | − 2.77674 |
| 2 | 0.124696 | − 0.00178 | 0.236951 | 7.920760262 | |
| 3 | − 0.23814 | − 1.44314 | 0.806048 | 0.445979348 | |
| 4 | 1.658264 | 0.031662 | 3.395981 | 1.912592839 | |
| 5 | 3.273171 | 3.005842 | 1.707205 | − 0.00158299 | |
| 6 | − 16.7121 | − 0.09035 | − 19.0599 | − 1.43222378 | |
| 7 | − 0.35034 | 0.706197 | − 2.80193 | 2.498670008 | |
| 8 | − 0.18928 | − 1.64078 | 0.865056 | − 0.31372102 | |

**Table 16** Weights and biases of the developed bottom-water ANN Model-2

| $i$ | Input weights | Input biases | Hidden layer weight | | Output bias |
|---|---|---|---|---|---|
| | $j_1(t_D)$ | $j_2(z_D)$ | $b_1$ | $Lw_1$ | $b_k$ |
| 1 | 1.805079 | 2.076887 | − 3.30911 | − 0.03915503 | − 2.02391 |
| 2 | 0.436752 | − 0.01212 | − 0.30621 | 2.123066693 | |
| 3 | − 0.91130 | − 1.42462 | 0.950519 | 0.026203499 | |
| 4 | 1.561208 | 0.04151 | 1.165645 | 0.208771991 | |
| 5 | 0.138143 | 3.278276 | 0.079355 | − 0.00335107 | |
| 6 | − 3.05957 | − 0.02748 | − 5.11476 | − 2.49343042 | |
| 7 | − 1.69774 | 0.814585 | − 2.80323 | 0.042488494 | |
| 8 | 0.466702 | − 4.55605 | 3.51433 | 0.001793503 | |

**Table 17** Weights and biases of the developed bottom-water ANN Model-3

| $i$ | Input weights | Input biases | Hidden layer weight | | Output bias |
|---|---|---|---|---|---|
| | $j_1(t_D)$ | $j_2(z_D)$ | $b_1$ | $Lw_1$ | $b_k$ |
| 1 | 3.821876 | 0.616475379 | −3.25721 | 0.031639 | −0.967766 |
| 2 | 2.530793 | −0.41886614 | −3.18143 | 0.125974 | |
| 3 | −0.90416 | −0.02989468 | 0.143353 | −0.90781 | |
| 4 | −3.42025 | 0.444806106 | 2.24222 | −0.02876 | |
| 5 | −0.59980 | 0.853563377 | −0.01835 | −0.08026 | |
| 6 | 1.656769 | 0.180457588 | 1.232131 | 0.202108 | |
| 7 | 2.129224 | 0.131789769 | 3.499738 | 1.082554 | |
| 8 | −1.17360 | 1.101690488 | −2.1765 | −0.04998 | |

**Table 18** Weights and biases of the developed bottom-water ANN Model-4

| $i$ | Input weights | Input biases | Hidden layer weight | | Output bias |
|---|---|---|---|---|---|
| | $j_1(t_D)$ | $j_2(z_D)$ | $b_1$ | $Lw_1$ | $b_k$ |
| 1 | 2.908838 | 3.559277 | −2.8956477 | 0.011586 | 0.330811951 |
| 2 | 0.382018 | −0.0190 | −0.2043294 | 2.443329 | |
| 3 | 0.888585 | 2.901874 | −1.7850305 | −0.01397 | |
| 4 | 2.737426 | 2.006805 | −0.1870357 | 0.017398 | |
| 5 | −2.46803 | −0.60677 | −1.4202537 | −0.06095 | |
| 6 | −2.54998 | −2.55896 | −1.979025 | −0.00039 | |
| 7 | −2.1266 | 1.111571 | −1.8037836 | −0.01525 | |
| 8 | 3.318732 | 0.345222 | 3.55855461 | 0.108533 | |

**Table 19** Weights and biases of the developed bottom-water ANN Model-5

| $i$ | Input weights | Input biases | Hidden layer weight | | Output bias |
|---|---|---|---|---|---|
| | $j_1(t_D)$ | $j_2(z_D)$ | $b_1$ | $Lw_1$ | $b_k$ |
| 1 | 1.67080288 | 3.266971 | 4.54356235 | −0.00531 | −0.5781 |
| 2 | 3.58887932 | 0.26768 | −2.3116437 | 0.066026 | |
| 3 | −1.5257419 | −3.45198 | 1.4390306 | −0.0037 | |
| 4 | −2.9644605 | −2.43589 | 0.49762271 | 0.02396 | |
| 5 | −0.9191691 | 1.868856 | 0.72593364 | −0.06854 | |
| 6 | −0.3809961 | −0.03627 | −0.2496443 | −2.61315 | |
| 7 | 0.98418142 | −4.13483 | 2.16249157 | 0.024557 | |
| 8 | 2.28647923 | −2.4174 | 4.08816674 | 0.032978 | |

## Generalization of the developed ANN models predictions and comparison with other existing models

The robustness of any developed model is its ability to predict with new datasets to justify its generalization as well as application. As earlier pointed out, the used of vEH approach for estimation of water influx into the reservoir(s) is achieved by table look-up. Most times, this required interpolation between two data points to obtain the corresponding $W_{\mathrm{eD}}$, as the calculated $t_D$ (even $r_{\mathrm{eD}}$ and $z_D$) may not be the exact value as provided in the vEH table. Therefore, the developed ANN models generalization potential was determined using randomly generated $t_D$ datasets based on their corresponding $r_{\mathrm{eD}}$ and $z_D$. The ANN models predicted $W_{\mathrm{eD}}$ were compared with the Lagrangian interpolation approach (Eq. 11) estimated $W_{\mathrm{eD}}$. Again, the edge-water infinite-acting models predicted $W_{\mathrm{eD}}$ were compared with the Edwardson et al. (1962) polynomial (Eqs. 12 and 13) estimated $W_{\mathrm{eD}}$. Also, the developed models (for finite and infinite-acting edge-water aquifers) predicted $P_D$ were compared with some existing models, namely, Chatas (1953), Edwardson et al. (1962), Lee (1982) and Fanchi (1985) estimations. The generalization performance of these developed models was determined using some statistical measures, like, coefficient of determination ($R^2$), $R$, MSE, root-mean-square error (RMSE) and average absolute relative error (AARE) and cross-plots.

**Table 20** Statistical performance of the developed finite aquifer ANN models with other models

| Finite aquifer type | Parameter/Model comparison | Statistical measures | | | | |
|---|---|---|---|---|---|---|
| | | $R^2$ | $R$ | MSE | RMSE | AARE |
| Edge-water | $W_{eD}$: Lang_ANN | 0.9984 | 0.9992 | 0.3496 | 0.5913 | 0.2414 |
| | $P_D$: Chatas_ANN | 0.9985 | 0.9993 | 0.0125 | 0.1117 | 0.0678 |
| Bottom-water | $P_D$: Fanchi_ANN | 0.9863 | 0.9931 | 0.1411 | 0.3756 | 0.2310 |
| | $W_{eD}$: Lang_ANN | 0.9993 | 0.9996 | 0.1863 | 0.4316 | 0.2215 |

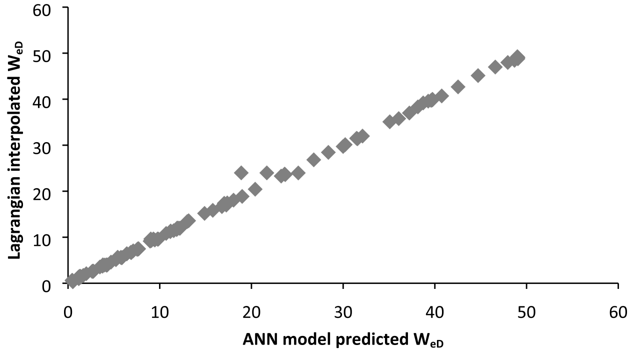


**Fig. 6** Comparison of edge-water finite aquifer ANN model predicted $W_{eD}$ with Lagrangian interpolation

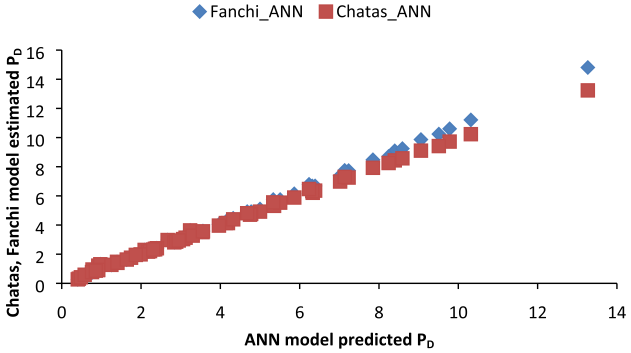


**Fig. 7** Comparison of edge-water finite aquifer ANN model predicted $P_D$ with other models

$$W_{\mathrm{eD}}(t_D) = W_{\mathrm{eD}}(t_{D_1})\left[\frac{t_{D_2} - t_D}{t_{D_2} - t_{D_1}}\right] + W_{\mathrm{eD}}(t_{D_2})\left[\frac{t_D - t_{D_1}}{t_{D_2} - t_{D_1}}\right] \tag{11}$$

where $W_{\mathrm{eD}}(t_D)$ is the required dimensionless influx at estimated $t_D$, $W_{\mathrm{eD}}(t_{D_1})$ and $W_{\mathrm{eD}}(t_{D_2})$ are the look-up dimensionless influx at $t_{D_1}$ and $t_{D_2}$, respectively, from vEH table.

Edwardson et al. (1962) polynomial:

(i). For $0.01 < t_D < 200$

$$W_{\mathrm{eD}} = \frac{\left(1.2838\sqrt{t_D} + 1.19328 t_D + 0.269872\left(t_D\right)^{1.5} + 8.55294\times 10^{-3}\left(t_D\right)^2\right)}{\left(1 + 0.616599\sqrt{t_D} + 0.0413008 t_D\right)} \tag{12}$$

(ii). For $t_D > 200$

$$W_{\mathrm{eD}} = \frac{-4.29881 + 2.02566 t_D}{\ln\left(t_D\right)} \tag{13}$$

Lee (1982) equation:

$$P_D = a_1 + a_2\ln(t_D) + a_3[\ln(t_D)]^2 + a_4[\ln(t_D)]^3 + a_5 t_D + a_6(t_D)^2 + a_7(t_D)^3 + \frac{a_8}{t_D} \tag{14}$$

where the values for $a_1$ through $a_8$ are: $a_1 = 0.8085064$, $a_2 = 0.29302022$, $a_3 = 0.035264177$, $a_4 = -1.4036304\times10^{-3}$, $a_5 = -4.7722225\times10^{-4}$, $a_6 = 5.1240532\times10^{-7}$, $a_7 = 2.3033017\times10^{-10}$ and $a_8 = -2.6723117\times10^{-3}$.

Fanchi (1985) equation:

$$P_D = a_0 + a_1 t_D + a_2\ln(t_D) + a_3[\ln(t_D)]^2 \tag{15}$$

where the values for $a_0$ through $a_3$ in the Fanchi (1985) equation varied depending on the $r_{\mathrm{eD}}$ value (i.e., 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 8.0, 10.0 and ∞) considered for the equation. The values for the constants $a_0$ through $a_3$ are available in Ahmed and McKinney (2005).

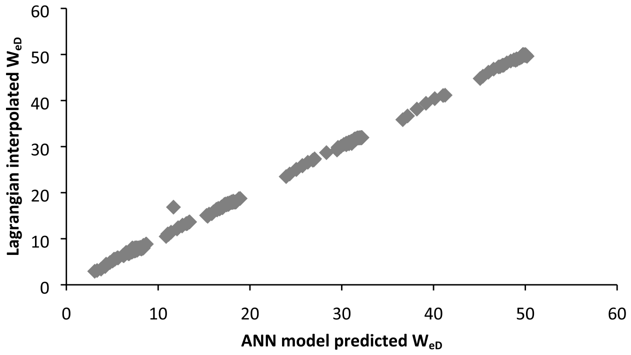


**Fig. 8** Comparison of bottom-water finite aquifer ANN model predicted $W_{eD}$ with Lagrangian interpolation

**Table 21** Statistical performance of the developed infinite-acting aquifer ANN models

| Reservoir drive | Parameter | ANN model | Statistical measures | | | | |
|---|---|---|---|---|---|---|---|
| | | | $R^2$ | $R$ | MSE | RMSE | AARE |
| Edge-water | $W_{eD}$ | Model-1 | 0.9999 | 0.9999 | 0.5447 | 0.7380 | 0.2329 |
| | $P_D$ | | 0.9996 | 0.9998 | $4.740\times10^{-4}$ | $2.178\times10^{-2}$ | $1.470\times10^{-2}$ |
| | $W_{eD}$ | Model-2 | 0.9999 | 0.9999 | 36.5811 | 6.0482 | 3.6096 |
| | $P_D$ | | 0.9999 | 0.9999 | $4.627\times10^{-7}$ | $6.802\times10^{-4}$ | $5.548\times10^{-4}$ |
| | $W_{eD}$ | Model-3 | 0.9999 | 0.9999 | $2.353\times10^{7}$ | $4.850\times10^{3}$ | $4.039\times10^{3}$ |
| | $P_D$ | | 0.9997 | 0.9999 | $2.180\times10^{-4}$ | $1.476\times10^{-2}$ | $1.177\times10^{-2}$ |
| | $W_{eD}$ | Model-4 | 0.9999 | 0.9999 | $8.576\times10^{9}$ | $9.261\times10^{4}$ | $8.459\times10^{3}$ |
| | $P_D$ | | 0.9999 | 0.9999 | $1.722\times10^{-6}$ | $1.312\times10^{-3}$ | $1.0\times10^{-3}$ |
| | $W_{eD}$ | Model-5 | 0.9998 | 0.9999 | $3.531\times10^{17}$ | $5.942\times10^{8}$ | $2.010\times10^{8}$ |
| | $P_D$ | | 0.9999 | 0.9999 | $5.546\times10^{-5}$ | $7.447\times10^{-3}$ | $4.246\times10^{-3}$ |
| Bottom-water | $W_{eD}$ | Model-1 | 0.9999 | 0.9999 | 0.2299 | 0.4795 | 0.1282 |
| | | Model-2 | 0.9999 | 0.9999 | 67.1527 | 8.1947 | 3.8671 |
| | | Model-3 | 0.9999 | 0.9999 | $3.914\times10^{8}$ | $1.978\times10^{4}$ | $4.617\times10^{3}$ |
| | | Model-4 | 0.9999 | 0.9999 | $2.921\times10^{11}$ | $5.410\times10^{5}$ | $4.240\times10^{5}$ |
| | | Model-5 | 0.9999 | 0.9994 | $3.160\times10^{17}$ | $5.620\times10^{8}$ | $3.870\times10^{8}$ |

**Table 22** Statistical performance of the overall edge- and bottom-water infinite-acting aquifer ANN models with other models

| Infinite-acting aquifer type | Parameter/model comparison | Statistical measures | | | | |
|---|---|---|---|---|---|---|
| | | $R^2$ | $R$ | MSE | RMSE | AARE |
| Edge-water | $W_{eD}$: Lang_ANN | 0.9999 | 0.9999 | $6.329\times10^{16}$ | $2.516\times10^{8}$ | $3.601\times10^{7}$ |
| | $W_{eD}$: Edwardson_ANN | 0.9999 | 0.9999 | $8.367\times10^{15}$ | $9.147\times10^{7}$ | $2.702\times10^{7}$ |
| | $P_D$: Edwardson_ANN | 0.9999 | 0.9999 | $1.75\times10^{-4}$ | $1.325\times10^{-2}$ | $7.333\times10^{-3}$ |
| | $P_D$: Fanchi_ANN | 0.9991 | 0.9996 | $1.079\times10^{-3}$ | $3.284\times10^{-2}$ | $2.823\times10^{-2}$ |
| Bottom-water | $P_D$: Lee_ANN | 0.9865 | 0.9933 | $1.425\times10^{-2}$ | $1.194\times10^{-1}$ | $8.31\times10^{-2}$ |
| | $W_{eD}$: Lang_ANN | 0.9999 | 0.9999 | $5.718\times10^{16}$ | $2.390\times10^{8}$ | $7.0\times10^{7}$ |

## Finite aquifer models

The generalization performance of the developed edge-water and bottom-water finite aquifer models was determined using 105 and 213, respectively, randomly generated $t_D$ datasets. Table 20 and Figs. 6, 7 and 8 present the generalization and comparison performance of these developed ANN models for the finite aquifer. From the table, the statistical measure results and the cross-plots in Figs. 6 and 8, the developed ANN models predicted $W_{\mathrm{eD}}$ were very close to the Lagrangian interpolated $W_{\mathrm{eD}}$. These models close predictions with the Lagrangian interpolation are visible in the diagonal trends of the data points along a unit slope (i.e., 45° from the origin). According to Al-Bulushi et al. (2009), this observation means a good agreement between the ANN models predicted $W_{eD}$ and Lagrangian interpolated $W_{\mathrm{eD}}$. Hence, the statistical measures of these developed models performance resulted in $R^2$, $R$, MSE, RMSE and AARE of 0.9984, 0.9992, 0.3496, 0.5913 and 0.2414 for edge-water ANN model and 0.9993, 0.9996, 0.1863, 0.4316 and 0.2215 for bottom-water model. This implies that the developed ANN models would conveniently predict $W_{\mathrm{eD}}$ for the edge-water and bottom-water finite aquifer with about 99% accuracy without any vEH table look-up. Also, comparing the developed edge-water ANN model predicted $P_D$ with some existing models: Chatas (1953) and Fanchi (1985) equations for evaluating $P_D$ showed that the developed ANN model predictions fit close to the Chatas (1953) and Fanchi (1985) models estimated $P_D$. A comparison of the ANN

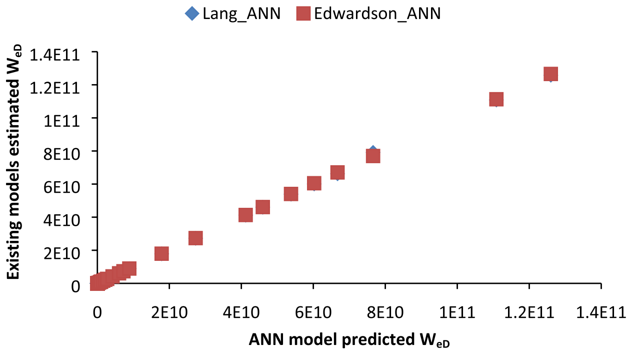


**Fig. 9** Comparison of edge-water infinite-acting aquifer ANN model predicted $W_{eD}$ with other models

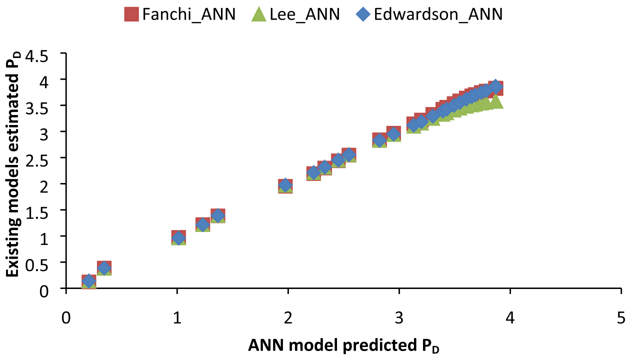


**Fig. 10** Comparison of edge-water infinite-acting aquifer ANN model predicted $P_D$ with other models

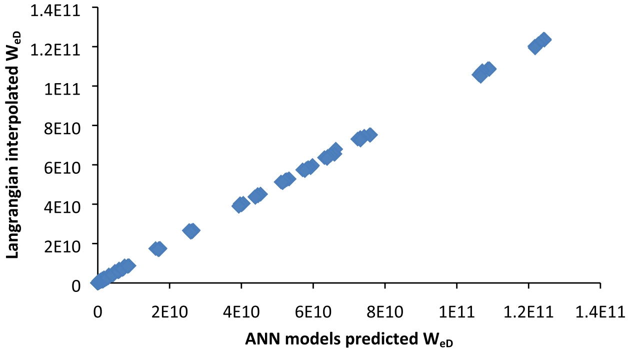


**Fig. 11** Comparison of bottom-water infinite-acting aquifer ANN model predicted $W_{eD}$ with Lagrangian interpolation

model predicted $P_D$ with the existing models estimated $P_D$ resulted in the statistical performance of $R^2$ of 0.9985, $R$ of 0.9993, MSE of 0.0125, RMSE of 0.1117 and AARE of 0.0678 with Chatas (1953) model and $R^2$, $R$, MSE, RMSE and AARE of 0.9863, 0.9931, 0.1411, 0.3756 and 0.2310, respectively, with Fanchi (1985) equation. Therefore, these developed ANN models can predict $W_{\mathrm{eD}}$ and $P_D$ for edge-water and the bottom-water finite aquifer that could be used to estimate water influx into the reservoir using van Everdingen–Hurst approach or Carter–Tracy method.

### Infinite-acting aquifer

The developed edge-water and bottom-water infinite-acting aquifer ANN models generalization capacity was determined using 106 and 735 randomly generated $t_D$ datasets. These generated $t_D$ datasets were divided to fall within the five models (Model-1–5) data ranges in Table 3. Table 21 presents the statistical performance of the developed edge-water and bottom-water infinite-acting aquifer models predictions (i.e., $W_{\mathrm{eD}}$ and $P_D$) with the new $t_D$ datasets when compared with Lagrangian interpolation for $W_{\mathrm{eD}}$ and Chatas (1953) estimation for $P_D$. The $R^2$ and $R$ values obtained for these ANN models indicate close predictions between the ANN models and these estimation methods. Furthermore, the developed infinite-acting aquifer ANN models overall generalization potentials were compared with other existing models. The results of this evaluation are visible in Table 22 as well as cross-plots in Figs. 9, 10 and 11. For the edge-water, the developed ANN model predictions were very close to Lagrangian interpolation and Edwardson et al. (1962) polynomial estimation of $W_{\mathrm{eD}}$; this closeness is noted in the unit slope in Fig. 9. Again, the efficient performance of this ANN model is seen in its predicted $P_D$ values, as they are comparable to Edwardson et al. (1962), Fanchi (1985) and Lee (1982) models estimated $P_D$. Also, in Table 22, the overall performance of the bottom-water infinite-acting model is close to the Lagrangian interpolation for $W_{\mathrm{eD}}$. In other words, these developed ANN models will predict $W_{\mathrm{eD}}$ and $P_D$ that could be functional for the calculation of water influx in edge-water and bottom-water infinite-acting reservoir.

In all, the advantages and novelty of the developed ANN models over some existing models are as follows:

1. Compared to Nashawi and Elkamel (1999) ANN models that are multiple-inputs single-output (MISO), the developed models in this study for $W_{\mathrm{eD}}$ and $P_D$ prediction for edge-water drive finite and infinite-acting aquifers are multiple-inputs multiple-outputs (MIMO) ANN models.
2. Unlike existing models, namely Lee (1982), Fanchi (1985), etc., that are correlation range limited and do not provide values for all the aquifer sizes presented by vEH, the developed ANN models cover the entire vEH datasets. Again, the ANN models can predict the $W_{\mathrm{eD}}$ and $P_D$ values for aquifer sizes that are not provided by vEH but are within the $r_{eD}$ and $z_D$ data ranges.
3. The challenges of table look-up and interpolation between time entries and preloading of the vEH data table into reservoir engineering software are taken care of with the developed ANN models for water influx calculation.

## The application of the developed ANN models in water influx calculations in reservoir engineering software

Any model be it empirical, analytical or numerical is developed to drive home some or specific problems. According to Okon et al. (2020), any developed model is useless if it lacks an area of application. Therefore, these developed ANN models would be useful for the prediction of dimensionless influx ($W_{\mathrm{eD}}$) and dimensionless pressure ($P_D$) for the calculation of water influx ($W_e$) into the reservoir when incorporated into the software. Worthy to note that the available reservoir engineering software, like, ECLIPSE-100,

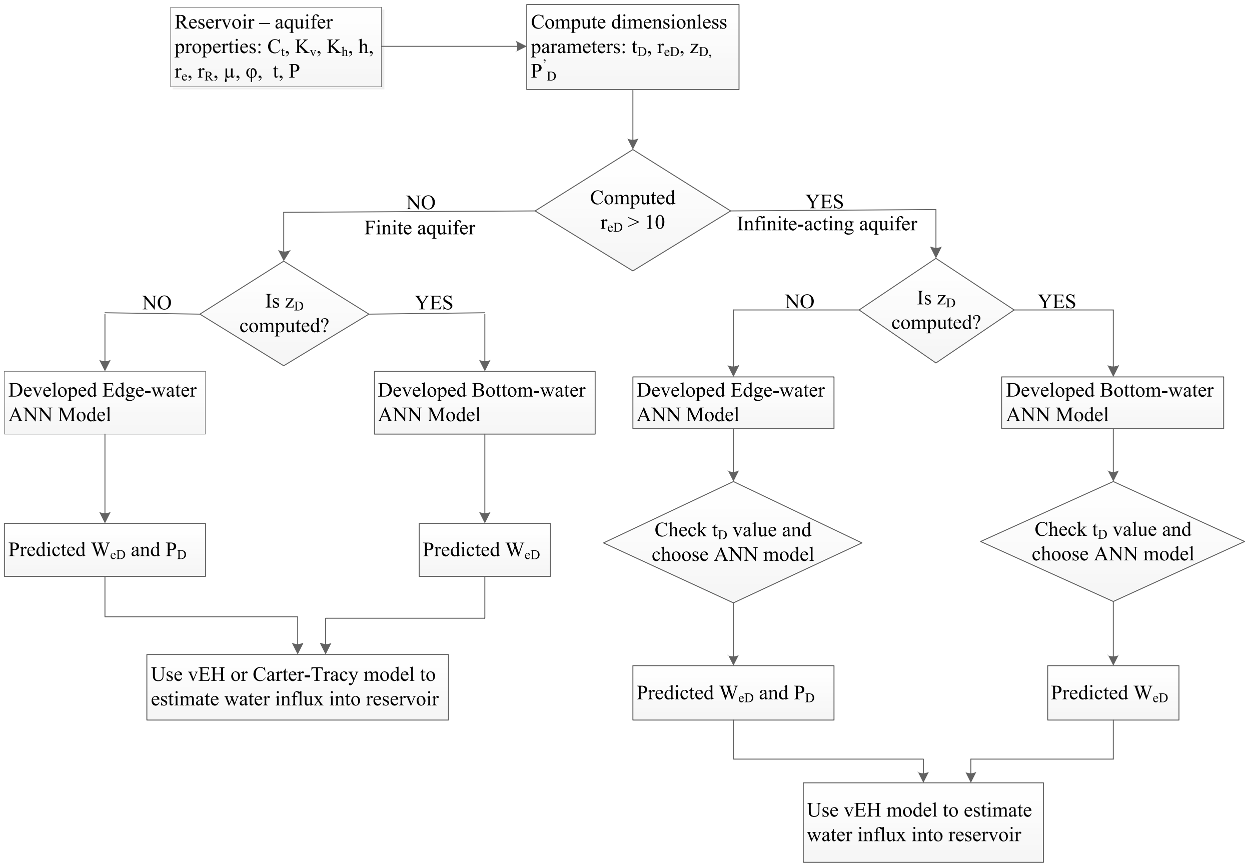

Reservoir – aquifer properties: $C_t$, $K_v$, $K_h$, h, $r_e$, $r_R$, μ, φ, t, P
Compute dimensionless parameters: $t_D$, $r_{eD}$, $z_D$, $P'_D$
Computed $r_{eD} > 10$
NO
Finite aquifer
YES
Infinite-acting aquifer
Is $z_D$ computed?
NO
YES
Developed Edge-water ANN Model
Developed Bottom-water ANN Model
Predicted $W_{eD}$ and $P_D$
Predicted $W_{eD}$
Use vEH or Carter-Tracy model to estimate water influx into reservoir
Is $z_D$ computed?
NO
YES
Developed Edge-water ANN Model
Developed Bottom-water ANN Model
Check $t_D$ value and choose ANN model
Check $t_D$ value and choose ANN model
Predicted $W_{eD}$ and $P_D$
Predicted $W_{eD}$
Use vEH model to estimate water influx into reservoir

**Fig. 12** Software conceptual schematic of the developed ANN models

PETREL, RESOLVE, etc., calculate water influx into the reservoir using an approximate solution to diffusivity equation—Carter–Tracy model or pseudo-state aquifer productivity index—Fetkovich model. The limitations of these approaches are that Fetkovich model applies to finite aquifer edge-water and bottom-water drive reservoirs, as Carter-Tracy aquifer model applies to edge-water drive reservoir finite-acting and infinite-acting aquifers (Okotie and Ikporo 2019). Again, Fetkovich approach underestimates the water influx into the reservoir, while Carter–Tracy method overestimates the water influx. Regrettably, the exact solution to diffusivity equation—van Everdingen–Hurst model that applies for both finite and infinite-acting aquifers edge-water and bottom-water drive reservoirs is limited in the reservoir engineering software. The reason is because of the challenge of table look-up in most $t_D$ entries to interpolate for $W_{\mathrm{eD}}$. In this direction, intelligent models that predict $W_{\mathrm{eD}}$ for the calculation of $W_e$ using vEH approach were necessary. The conceptual schematic of the developed ANN models for reservoir engineering software for the prediction of $W_{\mathrm{eD}}$ and $P_D$, and then the calculation of $W_e$ using vEH or Carter-Tracy method is as shown in Fig. 12. The software takes basic reservoir-aquifer properties and computes reservoir-aquifer dimensionless variables, namely, $t_D$, $r_{\mathrm{eD}}$, $z_D$ and $P'_D$. It then checks the computed $r_{\mathrm{eD}}$ to determine the aquifer outer boundary to establish the aquifer type, that is, finite aquifer or infinite-acting aquifer. The computer program further assesses the computed $z_D$ to determine the reservoir drive type: edge-water or bottom-water, and the ANN model to be used for the prediction. For instance, if the mentioned conditions established finite aquifer edge-water drive, then the developed ANN model takes input variables (i.e., computed $t_D$ and $r_{\mathrm{eD}}$) to predict $W_{\mathrm{eD}}$ and $P_D$. Then, vEH or Carter–Tracy model could beused to estimate the $W_e$ into the reservoir. Again, if the conditions established infinite-acting aquifer bottom-water drive, the developed ANN model takes the computed input variables (i.e., $t_D$ and $z_D$) and used the $t_D$ value to check for the appropriate ANN model's weights and biases to predict the $W_{\mathrm{eD}}$. Hence, the $W_e$ into the reservoir can be determined using vEH method.

## Conclusion

Water influx calculation is an indispensable evaluation in reservoir engineering as an alternative method to assess reservoir underground withdrawal performance. The most reliable approach by vEH-based models is hindered by table look-up and interpolation between time entries to determine the reservoir-aquifer dimensionless variables: influx and pressure. In this study, the feed-forward back-propagation (FFBP) artificial neural network (ANN) models were developed for the edge- and bottom-water drive finite and infinite-acting aquifers. The developed models are reproducible, as their weights and biases required to replicate the models are visible in this work. From the performance of these developed ANN models, the following conclusions can be drawn:

1. the developed ANN models predicted reservoir-aquifer dimensionless variables: $W_{eD}$ and $P_D$ were fitted to the original vEH-based model datasets, as the overall performance $R$ values obtained for the edge- and bottom-water finite aquifer models are 0.99983 and 0.99978, while edge- and bottom-water infinite-acting aquifer models are 0.99992 and 0.99997, respectively;
2. the developed finite aquifer ANN models predicted $W_{eD}$ values are comparable with the Lagrangian interpolation approach, as the obtained statistical measures between them resulted in the $R^2$ of 0.9984, $R$ of 0.9992, MSE of 0.3496, RMSE of 0.5913 and AARE of 0.2414 for edge-water drive model, while bottom-water drive model resulted in the $R^2$, $R$, MSE, RMSE and AARE of 0.9993, 0.9996, 0.1863, 0.4316 and 0.2215, respectively;
3. for the infinite-acting aquifer models, their predicted $W_{eD}$ and Lagrangian interpolated $W_{eD}$ resulted in the $R^2$, $R$, MSE, RMSE and AARE of 0.9999, 0.9999, 0.5447, 0.7380 and 0.2329 for edge-water drive model (Model-1); with an overall performance $R^2$ value of 0.9999, $R$ of 0.9999, MSE of $6.329 \times 10^{16}$, RMSE of $2.516 \times 10^8$ and AARE of $3.601 \times 10^7$. The bottom-water model (Model-1) had $R^2$ of 0.9999, $R$ of 0.9999, MSE of 0.2299, RMSE of 0.4795 and AARE of 0.1282; with an overall performance $R^2$ value of 0.9999, $R$ of 0.9999, MSE of $5.718 \times 10^{16}$, RMSE of $2.390 \times 10^8$ and AARE of $7.0 \times 10^7$;
4. again, the developed edge-water infinite-acting aquifer ANN model predicted $W_{eD}$ were comparable with Edwardson et al. (1962) polynomial estimated $W_{eD}$, as the obtained $R^2$ value of 0.9996, R of 0.9998, MSE of $4.740 \times 10^{-4}$, RMSE of $2.178 \times 10^{-2}$ and AARE of $1.470 \times 10^{-2}$ for Model-1; and overall performance of $R^2$, $R$, MSE, RMSE and AARE of 0.9999, 0.9999, $8.367 \times 10^{15}$, $9.147 \times 10^7$ and $2.702 \times 10^7$, respectively;
5. the developed edge-water ANN models predicted $P_D$ were closed to some existing models, namely Chatas (1953), Edwardson et al. (1962), Lee (1982) and Fanchi (1985) estimations, as the overall statistical performance: $R^2$, $R$, MSE, RMSE and AARE obtained for finite aquifer were 0.9985, 0.9993, $1.250 \times 10^{-2}$, 0.1117 and $6.780 \times 10^{-2}$ with Chatas (1953) model and 0.9863, 0.9931, 0.1411, 0.3756 and 0.2310 with Fanchi (1985) equation, while the infinite-acting aquifer was 0.9999, 0.9999, 0.1750, $1.325 \times 10^{-2}$ and $7.333 \times 10^{-3}$ with Edwardson et al. (1962) polynomial, then 0.9865, 09,933, $1.425 \times 10^{-2}$, 0.1194 and $8.310 \times 10^{-2}$ with Lee (1982) model, and 0.9991, 0.9996, $1.079 \times 10^{-3}$, $3.284 \times 10^{-2}$ and $2.823 \times 10^{-2}$ with Fanchi (1985) model;
6. the developed ANN models would perform better than the existing models since they are not correlation range limited and can predict the dimensionless variables: $W_{eD}$ and $P_D$, for aquifer sizes not provided in the vEH datasets.

For future work, as observed in the infinite-acting aquifer edge- and bottom-water models prediction, developing hybridize ANN models that minimize this aquifer-type dimensionless variables prediction error is important for water influx calculation in infinite-acting aquifers.

**Funding** The authors received no specific funding for this work.

**Declaration**

**Conflict of interest** The authors declare that there is no conflict of interest in the paper.

## References

Ahmed T, McKinney PD (2005) Advanced reservoir engineering. Gulf Professional Publishing, Elsevier, USA

Al-Bulushi N, King PR, Blunt MJ, Kraaijveld M (2009) Development of artificial neural network models for predicting water saturation and fluid distribution. J Pet Sci Eng 68:197–208

Al-Ghanim JA, Nashawi IS, Malallah A (2012) Prediction of water influx of edge-water drive reservoirs using nonparametric optimal transformations. Paper presented at the North Africa Technical Conference and Exhibition, Cairo, Egypt, 20–22 Feb 2012

Allard DR, Chen SM (1988) Calculation of water influx for bottom-water drive reservoirs. SPE Res Eng 3(02):369–379

Anifowose F, Ewenla A, Eludiora S (2012) Prediction of oil and gas reservoir properties using support vector machines. Paper presented at the international petroleum technical conference, Bangkok, Thailand, 7–9 Feb 2012

Ansa IB (2019) Artificial-intelligence-models-based for predicting dimensionless water influx for infinite-acting reservoirs. University of Uyo, Nigeria, BEng Project

Carter RD, Tracy GW (1960) An improved method for calculating water influx. Trans AIME 219(01):415–417. https://doi.org/10.2118/1626-G

Chatas AT (1953) A practical treatment of nonsteady-state flow problems in reservoir systems. J Pet Eng 25:44–56

Coats KH (1962) A mathematical model for water movement about bottom-water drive reservoirs. Soc Pet Eng J 2(01):44–52. https://doi.org/10.2118/160-PA

Cristofaro RA, Longhin GA, Waldmann AA, de Sá CHM, Vadinal RB, Gonzaga KA, Martins AL (2017) Artificial intelligence strategy minimizes lost circulation non-productive time in Brazilian deep-water pre-salt. In: Paper OTC 28034 presented at the offshore technology conference Brasil, Rio de Janeiro, Brazil, 24–26 https://doi.org/10.4043/28034-MS

Dang C, Nghiem L, Fedutenko E, Gorucu E, Yang C, Mirzabozorg A (2018) Application of Artificial intelligence for mechanistic modeling and probabilistic forecasting of hybrid low salinity chemical flooding. In: Paper SPE 191474 presented at the society of petroleum engineers annual technical conference and exhibition, Dallas, Texas, 24–26 Sept.https://doi.org/10.2118/191474-MS

Edwardson MJ, Girner HM, Parkison HR, Williams CD, Matthews CS (1962) Calculation of formation temperature disturbances caused by mud circulation. J Pet Technol 14(04):416–426. https://doi.org/10.2118/124-PA

Etim IO (2019) Artificial intelligence models for predicting dimensionless water influx for a bounded reservoirs. University of Uyo, Nigeria, BEng Project

Fanchi JR (1985) Analytical representation of the van Everdingen-Hurst aquifer influence functions for reservoir simulation. Soc Pet Eng J 25(03):405–406. https://doi.org/10.2118/12565-PA

Fetkovitch MJ (1971) A simplified approach to water influx calculations-finite aquifer systems. J Pet Technol 23(07):814–828. https://doi.org/10.2118/2603-PA

Hagan MT, Menhaj MB (1994) Training feed-forward techniques with Marquardt algorithm. IEEE Trans Neural Netw 5(6):989–993

Hammam H, Ertekin T (2018) A generalized varying oil compositions and relative permeability screening tool for continuous carbon dioxide injection in naturally fractured reservoirs. In: Paper SPE 192194 presented at the Society of petroleum engineers kingdom of saudi arabia annual technical symposium and exhibition, Dammam, Saudi Arabia, 23–26 April. https://doi.org/10.2118/192194-MS

Han S, Kim KW, Kim S, Youn YC (2018) Artificial neural network: understanding the basic concepts without mathematics. Dementia Neurocogniti Disorder 17(3):83–89. https://doi.org/10.12779/dnd.2018.17.3.83

Hurst W (1943) Water influx into a reservoir and its application to the equation of volumetric balance. Trans AIME 151(01):57–72. https://doi.org/10.2118/943057-G

Jiang S, Lian M, Lu C, Gu Q, Ruan S, Xie X (2018) Ensemble prediction algorithm of anomaly monitoring based on big data analysis platform of open-pit mine slope. Complexity 2018:1–13. https://doi.org/10.1155/2018/1048756

Khan MR, Tariq Z, Abdulraheem A (2018) Machine learning derived correlation to determine water saturation in complex lithologies. Paper SPE 192307 presented at the Society of Petroleum Engineers Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition, Dammam, Saudi Arabia, 23–26 April. https://doi.org/10.2118/192307-MS

Klins MA, Bouchard AJ, Cable CL (1999) A polynomial approach to the van Everdingen-Hurst dimensionless variables for water encroachment. Soc Pet Eng Res Eng 3(1):320–326. https://doi.org/10.2118/15433-PA

Konate AA, Pan H, Khan Y, Yang JH (2015) Generalized regression and feed-forward back-propagation neural networks in modeling porosity from geophysical well logs. J Petrol Explor Prod Technol 5:157–166. https://doi.org/10.1007/s13202-014-0137-7

Krenker A, Bester J, Kos A (2011) Introduction to the artificial neural networks. In: Suzuki K (ed.), Artificial neural networks—methodological advances and biomedical applications. InTech Janeza Trdine 9, 51000 Rijeka, Croatia

Lee J (1982) Well testing. SPE of AIME, Dallas, Texas, USA

Leung WF (1986a) A fast convolution method for implementing single-porosity finite/infinite aquifer models for water influx calculations. Soc Pet Eng Res Eng 1(5):490–510. https://doi.org/10.2118/12276-PA

Leung WF (1986b) A new pseudosteady-state model for dual-porosity/dual-permeability aquifers and two interconnected single-porosity aquifers. Soc Pet Eng Res Eng 1(5):511–520. https://doi.org/10.2118/12277-PA

Long W, Chai D, Aminzadeh F (2016) Pseudo density log generation using artificial neural network. In: Paper SPE 180439 presented at the Society of Petroleum Engineers Western Regional Meeting, Anchorage, Alaska, 23–26,https://doi.org/10.2118/180439-MS

Ma YZ, Gomez E (2015) Uses and abuses in applying neural networks for predictions in hydrocarbon resource evaluation. J Pet Sci Eng 133:66–75. https://doi.org/10.1016/j.petrol.2015.05.00

Musa H, Hamisu MA (2019) An artificial neural network model for predictions of impacts of MOSFET transconductance parameter on noise currents. J Res Innovations in Eng 4(1):16–24

Mustafa AA, Jada AP, Omer MM, Elhadi AM (2018) Calculation of cumulative water influx using van Everdingen model with superposition concept by MATLAB program. Sudan University of Science and Technology, Sudan, BSc Project

Nashawi IS, Elkamel A (1999) Neural network for the prediction of water aquifer dimensionless variables foe edge- and bottom-water drive reservoirs. Energy Fuel 13:88–89

Okon AN, Appah D (2018) Integrated-reservoir-model-based critical oil rate correlation for vertical wells in thin oil rim reservoirs in the Niger Delta. Int J Eng Technol 7(3):1757–1761

Okon AN, Adewole SE, Uguma EM (2020) Artificial neural network model for reservoir petrophysical properties: porosity, permeability and water saturation prediction. J Modeling Earth Syst Environ. https://doi.org/10.1007/s40808-020-01012-4

Okotie S, Ikporo B (2019) Water influx. Reservoir Engineering, Springer. https://doi.org/10.1007/978-3-030-02393-5_4

Schilthius RJ (1963) Active oil and reservoir energy. Trans AIME 118(01):33–52. https://doi.org/10.2118/936033-G

van Everdingen AF, Hurst W (1949) The application of Laplace transform to flow problems in reservoirs. J Pet Technol 1(12):305–324. https://doi.org/10.2118/949305-G

Wood DA (2019) Predicting porosity, permeability and water saturation applying an optimized nearest neighbour, machine-learning and data-mining network of well-log data. J Pet Sci Eng 184:1–17. https://doi.org/10.1016/j.petrol.2019.106587

Zou J, Han Y, So S (2008) Overview of artificial neural networks. In: Livingstone DS (ed) Artificial neural networks: methods and protocols. Humana Press. Springer. https://doi.org/10.1007/978-1-60327-101-1

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.