



Functional networks as a new data mining predictive paradigm to predict permeability in a carbonate reservoir

Emad A. El-Sebakhy^{a,*}, Ognian Asparouhov^a, Abdul-Azeez Abdulraheem^b, Abdul-Aziz Al-Majed^b, Donghui Wu^a, Kris Latinski^a, Iputu Raharja^c

^a MEDai Inc, An Elsevier Company, Millenia Park One, 4901 Vineland Road, Suite 450, Orlando, FL 32811, USA

^b Center of Petroleum and Minerals, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

^c Information and Computer Science, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

ARTICLE INFO

Keywords:

Functional networks
Data mining
Feedforward neural networks
Fuzzy logic
Statistical regression
Permeability
Porosity
Carbonate reservoir
Minimum description length

ABSTRACT

Permeability prediction has been a challenge to reservoir engineers due to the lack of tools that measure it directly. The most reliable data of permeability obtained from laboratory measurements on cores do not provide a continuous profile along the depth of the formation. Recently, researchers utilized statistical regression, neural networks, and fuzzy logic to estimate both permeability and porosity from well logs. Unfortunately, due to both uncertainty and imprecision, the developed predictive modelings are less accurate compared to laboratory experimental core data. This paper presents functional networks as a novel approach to forecast permeability using well logs in a carbonate reservoir. The new intelligence paradigm helps to overcome the most common limitations of the existing modeling techniques in statistics, data mining, machine learning, and artificial intelligence communities. To demonstrate the usefulness of the functional networks modeling strategy, we briefly describe its learning algorithm through simple distinct examples. Comparative studies were carried out using real-life industry wireline logs to compare the performance of the new framework with the most popular modeling schemes, such as linear/nonlinear regression, neural networks, and fuzzy logic inference systems. The results show that the performance of functional networks (separable and generalized associativity) architecture with polynomial basis is accurate, reliable, and outperforms most of the existing predictive data mining modeling approaches. Future work can be achieved using different structure of functional networks with different basis, interaction terms, ensemble and hybrid strategies, different clustering, and outlier identification techniques within different oil and gas challenge problems, namely, 3D passive seismic, identification of lithofacies types, history matching, rock mechanics, viscosity, risk assessment, and reservoir characterization.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

The process of oil exploration has been drastically improved by the availability of information technology, especially artificial intelligence with data mining modeling, as well as advances in seismic technology. Advances have pushed the envelope of what is feasible, both in terms of finding the oil and figuring out how to extract it once an oil and gas company has identified a location. Cutting edge advances in information technology and computational intelligence represent a breakthrough in oil and gas exploration and production. They have helped transform the business of exploration and production, increasing its production efficiency; and generating significant environmental benefits.

Permeability is defined as the ability of porous rock to transmit fluid. It is one of the most crucial reservoir properties and is very difficult to calculate accurately. Generally, permeability is directly measured in a laboratory on rock samples. Last two decades, numerous efforts have been made to forecast permeability using well log data and available core data. The field of permeability prediction is vast, and incorporates many years of effort by expert petrophysicists, geologists and reservoir engineers. The problem with computational methodologies for permeability prediction from logs is the inherent inability of this method to integrate the geological and petrophysical controls on single phase flow. Carbonate reservoirs in particular are characterized by a very wide range of measured permeability for a given porosity, which reflects heterogeneity in pore size, geometry and connectivity.

The relationships for forecasting permeability using wireline logs and core permeability data are based on statistical regression, feed-forward neural networks (FFNs), and fuzzy logic (FL) or adaptive neuro-fuzzy inference systems (ANFISs). Based on these ap-

* Corresponding author.

E-mail address: e.el-sebakhy@elsevier.com (E.A. El-Sebakhy).

proaches, better estimates have been reported compared to that of conventional techniques. Nevertheless, the application of neural networks to reservoir characteristics prediction is quite limited. Attempts were made to apply neural networks and fuzzy logic to permeability prediction, reservoir characteristics, pressure–volume–temperature prediction, and flow regimes and liquid-hold-up (Abdulraheem et al., 2007; Bhatt & Helle, 2002; El-Sebakhy, 2009; El-Sebakhy, Hadi, & Faisal, 2007; Fangming, Zhongyuan, Aming, & Xiaoxia, 2009; Lucia, Ruppel, Jennings, & Laubach, 2001; Nikraves & Aminzadeh, 2000, 2003). The findings of these works clearly indicate the potential of neural networks, which generally produces better results than those empirical formulae, derived from conventional multiple regression analysis.

The results obtained from neural networks models are reasonably accurate. However, there is a need for further improvement when this method is applied to a complex structure.

The use of combination of neural networks and fuzzy logic has also been reported in literature by Balan, Mohaghegh, and Ameri (1995), Cuddy (1997), Tamhane, Wong, Aminzadeh, and Nikraves (2000), Lucia et al. (2001), Bhatt and Helle (2002), Nikraves and Aminzadeh (2000), Nikraves and Aminzadeh (2003), Kamali and Mirshady (2004), Lim (2005), Sears and Lucia (2006), Kadkhodaie-Ilkhchi, Rezaee, and Moallemi (2006), Chen and Lin (2006), Jeirani and Mohebbi (2006), Lucia (2008), Abdulraheem et al. (2007), El-Sebakhy et al. (2007), Fangming et al. (2009), El-Sebakhy (2009), and Cross et al. (2010). Unfortunately, both neural networks and fuzzy logic inference systems are heuristic modeling approaches and suffer from a number of drawbacks, such as overfitting and local optima. In some cases, these techniques do not perform well, because the parameters in a training algorithm are based on initial guess of random weights, learning rate, and momentum. Therefore, there is a need for predictive modeling framework to estimate permeability and lithofacies from well logs in a carbonate reservoir.

Recently, functional network has been proposed as a new intelligence data mining predictive model in solving numerous of prediction/classification problems, namely, *pattern recognition, bioinformatics, engineering, software engineering, and business applications* (Bruen & Yang, 2005; Castillo, Cobo, Gutierrez, & Hadi, 1999; Castillo, Gutiérrez, Hadi, & Lacruz, 2001, 2008; El-Sebakhy, 2004; El-Sebakhy, 2009; El-Sebakhy, 2010; El-Sebakhy et al., 2007; El-Sebakhy et al., 2010). It has been only utilized in solving two oil and gas industry problems, namely, predicting reservoir fluids PVT properties, Al-Bokhitan (2007) and rock mechanical parameters for hydrocarbon reservoirs, El-Sebakhy et al. (2010). The main motivation of this research is to investigate the strengths and capabilities of functional networks in forecasting the permeability from well logs in a hydrocarbon reservoir and compares its performance with the one of the most common statistics and artificial intelligence modeling techniques.

The workflow of this research is designed as follows: Section 2 of this paper provides a brief literature review. Functional networks intelligent data mining system methodology and examples are described in detail in Section 3. The most common statistical quality measures in predictive modeling are proposed in Section 4. The use of functional networks in identifying the permeability from well logs implementations process and comparative studies are carried out in Section 5. Section 6 contains discussions of results, while Section 7 presents conclusions and recommendations.

2. Literature review

Recently, researchers in both petroleum engineering and computer science have achieved considerable success in forecasting permeability from wireline logs based on statistical regression,

standard neural networks, and fuzzy logic (FL) inference systems, see Lucia et al. (2001), Bhatt and Helle (2002), Nikraves and Aminzadeh (2000, 2003), Kamali and Mirshady (2004), Lim (2005), Sears and Lucia (2006), Kadkhodaie-Ilkhchi et al. (2006), Chen and Lin (2006), Jeirani and Mohebbi (2006), Lucia (2008), Abdulraheem et al. (2007), El-Sebakhy et al. (2007), Fangming et al. (2009), El-Sebakhy (2009), and Cross et al. (2010) for more details. Most of these empirical and computational correlations are often determined using log-derived parameters, such as, *porosity, density and Gamma ray*. Bruce et al. (2000) and Nikraves and Aminzadeh (2000, 2003) presented state-of-the-art reviews of the use of neural networks for predicting permeability from well logs. They had also developed a *nonlinear regression* tool to obtain transformation between well logs and core permeability. The tool can be used for estimating permeability in un-cored intervals of the well. Though the results that obtained from neural networks and regression models are reasonably accurate, there is room for further improvement in both efficiency and stability, especially in regions with complex geology. In addition, both neural networks and fuzzy logic inference systems are heuristic modeling approaches and suffer from a number of drawbacks, such as overfitting and local optima. Furthermore, in some cases, these techniques do not perform well because the parameters in the training algorithm are based on initial guess of random weights, learning rate, and momentum. Therefore, there is a need for new intelligence predictive modeling frameworks to estimate both permeability and lithofacies from well logs in a carbonate reservoir.

In the last decade, application of fuzzy logic in petroleum engineering field has received considerable attention and it has been successfully applied to address problems on various oil and gas reservoirs such as estimation of lithofacies and permeability using wireline logs, see Cuddy (1997), Tamhane et al. (2000), and Abdulraheem et al. (2007) for more details. For predicting these properties, generally *Gaussian membership* and *fuzzy clustering algorithm* are applied. Based on these approaches, better estimates have been reported compared to that of conventional techniques. The use of a combination of neural networks and fuzzy logic has also been reported in the literature by Nikraves and Aminzadeh (2000, 2003). This combined approach can be utilized in developing an optimum set of rules for nonlinear mapping between various input parameters. Such rules developed for the training set can be used to predict permeability for a new dataset and are very useful in the case when some of the inputs are missing. Predictive modeling techniques based on fuzzy logic were successful for most of the cases, but were not satisfactory in some cases due to the heuristic approach in dealing with the data.

2.1. Statistical linear/nonlinear regression

Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent variable y . The regression line for p explanatory variables x_1, x_2, \dots, x_p is defined to be $\mu_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$. This equation describes how the mean response μ_y changes with the explanatory variables. The observed values for y vary about their means μ_y and are assumed to have the same standard deviation, σ . Formally, the model for multiple linear regression given n observations is $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \varepsilon_i$, for $i = 1, \dots, n$, where ε_i is the model deviation. One approach to simplifying multiple regression equations is to use the stepwise procedures. These include *forward selection, backwards elimination, and stepwise regression*, which add or remove variables one-at-a-time until some stopping rule is satisfied.

2.2. Multilayer perceptron and feedforward neural networks

Artificial neural network is an information processing system that simulates the human brain. The topology of neural networks is a graphical structure in which the nodes are the predictors and directed edges with weights are connections between neuron outputs and neuron inputs of the next layer. Neural networks are characterized by the network architecture, that is: the number of layers, the number of nodes in each layer, and how the nodes are connected.

Multilayer perceptron (MLP) network is the most popular form of neural networks, where all nodes and layers are arranged in a feedforward manner. The first layer is called the input layer, where external information is received. The last layer is called the output layer, where the network produces the model solution. In between, there are one or more hidden layers, which are critical for the performance of the neural network. All nodes in adjacent layers are connected by acyclic arcs from a lower layer to a higher layer.

There are numerous books and review papers in the Computer Science, Engineering, and Business communities which summarize the multilayer perceptron and feedforward neural networks, most commonly used with the backpropagation (BP) algorithm, see Arbogast and Franklin (1999), Duda, Hart, and Stock (2001), and Badarinadh (2002), and the references therein for more details. An example of such neural networks with two hidden layers is shown in Fig. 1.

In the multilayer perceptron neural networks, the basic predictive model that provides a nonlinear transformation of a pattern $x \in R^p$ to $g(x) \in R$ or $g(x) \in \{1, \dots, c\}$ can be written as:

$$g_j(x) = \sum_{k=1}^m \omega_{jk} f\left(\sum_{i=1}^p \omega_{ki} x_i + \omega_{k0}\right) + \omega_{j0}; \quad j = 1, 2, \dots, c,$$

where m represents the number of hidden nodes, w_{jk} and w_{ki} are weights on links, w_{j0} and w_{k0} are biases on nodes. The nonlinear function $f(\bullet)$ is usually of the form of the sigmoid (*s-curve*) function: $(1 + e^{(-\alpha z)})^{-1}$, where $z \in R$ and $\alpha > 0$ is a constant called the gain parameter. Often in practice, the number of outputs, c , is taken as the number of classes, with each output corresponding to one class. A test sample is assigned to the class with the largest output value.

In the multilayer perceptron neural networks model, the optimal weights and biases are found by optimizing a criterion function. Least squared error is often used to minimize the function in Eq. (1):

$$\text{Min} \left\{ J = \frac{1}{2} \left[\sum_{i=1}^n \|t_i - g(x_i)\|^2 \right] \right\}, \quad (1)$$

where $g(x_i)$ is the output vector for the input x_i and $t_i = (t_{i1}, \dots, t_{ic})^T$ is the corresponding target vector. In the MLP model, the procedure is to find the optimal weights and biases using a backpropagation

optimization algorithm. Since the objective function is not convex with respect to its parameters, the backpropagation algorithm often gets trapped at a local optimum. Often in practice, to find a good solution the backpropagation algorithm needs to be run several times, each time with different initial weights. This creates a heavy computational load and is often considered to be the major drawback of the multilayer perceptron networks model.

The feedforward multilayer perceptron network is composed of layers of interconnected neurons. Usually, for sufficient number of hidden units, it can approximate any continuous static input–output mapping to any desired degree of approximation, as documented by Simon (1994) and White and Stinchcombe (1989). The learning process involves updating networks architecture and connection weights; so that a network can do both prediction and classification tasks using some efficient learning algorithms, such as, back propagation algorithm (Hopfield, 1982; Jacob, 1988; Simon, 1994; White & Stinchcombe, 1989). Several improvements of the backpropagation algorithm were proposed in the literature, e.g. Delta–Bar–Delta algorithm, quick-propagation, Fahlman (1988), and more recently the resilient back-propagation, Riedmiller and Braun (1993). Several types of neural networks have also been proposed for function approximation, forecasting, and classification, such as radial basis function networks, probabilistic neural network, and abductive (polynomial) neural networks. For the sake of simplicity, we propose only the radial basis function networks and its implementation process for predictions.

2.3. Radial basis function networks

Radial basis function (RBF) network is similar to the multilayer perceptron model, the basic RBF model provides a nonlinear transformation of a pattern $x \in R^p$ to $g(x) \in R$ or $g(x) \in \{1, \dots, c\}$, that is,

$$g_j(x) = \sum_{i=1}^m \omega_{ji} \varphi\left(\frac{\|x - \mu_i\|}{h}\right) + b_j; \quad j = 1, \dots, c; \quad (2)$$

where m is a constant representing the number of basis functions, ω_{ji} is a weight, b_j is a bias, $\varphi(\cdot)$ is a radial symmetric basis function, $\mu_i \in R^p$ is called a center vector, and $h \in R$ is a smoothing parameter. We note that RBF based on Eq. (2) has almost the same mathematical form as that of MLP networks, the key difference being that the logistic function is replaced by a radial basis function, which is often taken to be the Gaussian function $\varphi(z) = e^{-z^2}$, where $z = h^{-1} \|x - \mu\|$. The RBF adopted least-squared errors optimization criterion to estimate the weights ω_{ji} within the selected architecture of the RBF networks. Unlike the multilayer perceptron model, where all parameters are optimized at the same time, in the radial basis function networks model, both *center vectors* and *weights* are found in two separate steps (Hopfield, 1982; Jacob, 1988; White & Stinchcombe, 1989). In the first step, the center vectors are found using some existing pattern recognition techniques (such as *k-means clustering* or the *Gaussian mixture*). In the second step, a set of linear equations is solved to find the optimal weights and biases. Consequently, the training of the RBF networks model does not run into the problem of local optima as with the MLP model. Therefore, the training time for RBF networks models is reported to be shorter. However, since there is often some correlation between the locations of center vectors and the weights, the RBF approach may lead to a suboptimal solution.

2.4. Neuro-fuzzy inference systems

Fuzzy logic is an excellent tool for modeling the kind of uncertainty associated with vagueness, imprecision, and/or a lack of information regarding a particular element of the problem at hand. It performs well on uncertain information, very similar to the way

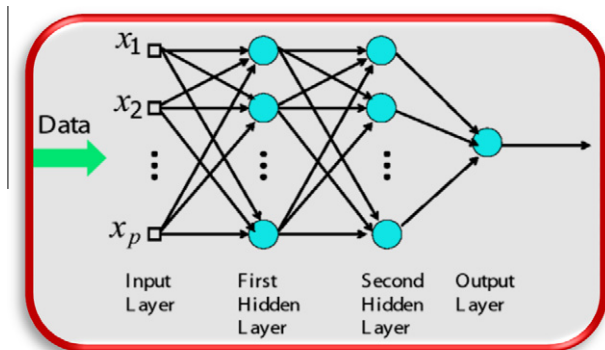


Fig. 1. Multilayer feedforward neural networks.

human reasoning does. Brief details and the most common basic concepts, such as concepts of fuzzy set theory, including: fuzzy relations, fuzzification and defuzzification, construction of membership functions, and fuzzy arithmetic are given in Duda et al. (2001) and LeCun et al. (1995). Fuzzy sets are defined through their membership functions, μ which map the elements of the considered universe to the unit interval $[0, 1]$ as in Eq. (3), see Amabeoku et al. (2005) for more details. The membership of an element x in the crisp set A is represented by the characteristic function μ_A of A , that is

$$\mu_A(x) = \begin{cases} 1; & \text{if } x \in A, \\ 0; & \text{if } x \notin A. \end{cases} \quad (3)$$

Generally, the rule-based fuzzy modeling technique can be classified into three categories, namely the *linguistic (Mamdani-type)*, the *relational equation*, and the *Takagi, Sugeno and Kang (TSK)*, see Cuddy (1998), Mizutani and Jang (1997), Hambalek and Reinaldo (2003), Abdulraheem et al. (2007), and El-Sebakhy (2009). In linguistic models, both the antecedent and the consequence are fuzzy sets, while in the TSK model the antecedent consists of fuzzy sets but the consequence is made up of linear equations. We are going to focus on the developed neuro-fuzzy permeability systems with the TSK model, because TSK needs fewer rules and its parameters can be estimated from numerical data using optimization methods such as least-square algorithms, see Liu, Nakashimam, Sakom, and Fujisawam (2003), Abdulraheem et al. (2007), and El-Sebakhy (2009).

The basic architecture of a type-1 FLS with crisp inputs and output is shown in Fig. 2. The TSK fuzzy modeling method was proposed by Takagi and Sugeno as a framework for generating fuzzy if-then rules from numerical data. A TSK fuzzy model consists of a set of fuzzy rules, each describing a local linear input–output relationship:

REL_i : if x_1 is A_{i1} and x_2 is A_{i2}, \dots , and if x_p is A_{ip} , then

$$y_i = a_{i0} + a_{i1}x_1 + \dots + a_{ip}x_p; i = 1, 2, \dots, n;$$

where REL_i is the i th rule; x_1, \dots, x_p are the input variables; A_{i1}, \dots, A_{ip} are the fuzzy sets assigned to corresponding input variables; y_i represents the value of the i th local output; and a_{i0}, \dots, a_{ip} are the model

consequent parameters. The final global output of the TSK fuzzy model for a crisp input vector $\mathbf{x} = (x_1, \dots, x_p)$ is calculated using the fuzzy mean-weight formula $\hat{y} = \sum_{i=1}^n \beta_i(x_i) y_i / [\sum_{i=1}^n \beta_i(x_i)]^{-1}$, where $\beta_i(x_i)$ represents the *degrees of firing (DOF) of the fuzzy rule* and is defined as $\beta_i(x_i) = \text{Min}\{\mu_{A_{i1}}(x_1), \dots, \mu_{A_{ip}}(x_p)\}$.

The construction of the TSK fuzzy model from numerical data proceeds in three steps: *fuzzy clustering*, *setting the membership functions*, and *parameter estimation*, see Mizutani and Jang (1997), Liu et al. (2003), and El-Sebakhy (2009) for more details about the Type 1 Fuzzy logic architecture and the corresponding Gustafson Kassel clustering schemes with numerous distance forms. Let X denote the matrix whose i th row is the input vector x_i and let Y denote the column vector with y_i as its i th component. Let $W_k(x_i)$ denote the $n \times n$ real diagonal matrix that represents normalized firing strength of the k th rule for the i th observation or sample, $W_k(x_i) = \beta_k(x_i) [\sum_{j=1}^K \beta_j(x_i)]^{-1}$, where $k = 1, \dots, K$; $i = 1, \dots, n$. Suppose that $\Theta_i \equiv [a_{i0}, \dots, a_{in}]$ denotes the vector of consequent parameters of the i th rule. In order to estimate the off-set term, a_{i0} , a unitary column \mathbf{I} is appended to the matrix, X , to produce the extended matrix $X_e \equiv [X, \mathbf{I}]$. Therefore, the unknown parameters Θ_i are calculated via least-square criterion, $\Theta_i = [X_e^T \cdot W_i \cdot X_e]^{-1} \cdot X_e^T \cdot W_i \cdot Y$.

3. Functional networks intelligence paradigm: methodology and implementations

3.1. Introduction and overview

The functional network is a generalization of the standard neural network; it deals with generalized functional models instead of sigmoidal standard types. The initial architecture of functional networks is problem driven, i.e., it is designed based on the problem at hand. The final topology of the functional network is chosen according to knowledge expertise, data domain, and information about the other properties of the neuron function (associativity, commutativity, and invariance). In functional networks the neuron functions associated with each neuron are not fixed but are learned from the available data. Hence, there is no need to include weights

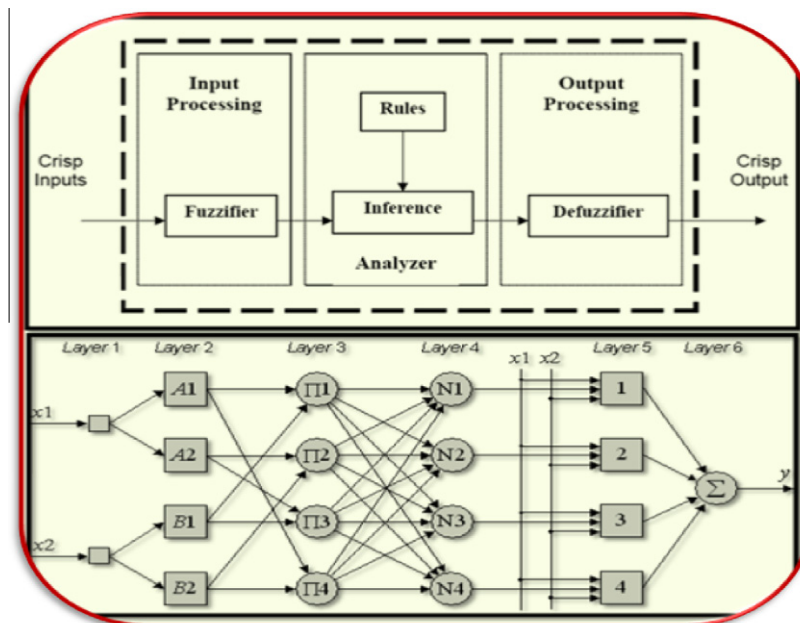


Fig. 2. Neuro-fuzzy inference system: Two rules Sugeno with crisp inputs and output architecture.

associated with links, since the neuron functions subsume the effect of weights. In addition, functional networks allow neurons to be multi-argument, multivariate, and different learnable functions, instead of fixed functions. Furthermore, functional networks allow converging neuron outputs, forcing them to be coincident. This leads to a system of functional equations, which requires some compatibility conditions on the neuron functions, see Castillo et al. (1999, 2001), Castillo, Hadi, Lacruz, and Pruneda (2008), Bruen and Yang (2005), El-Sebakhy (2004, 2009, 2010), and El-Sebakhy et al. (2007, 2010) for more details.

The functional networks modeling scheme has been utilized in solving numerous prediction/classification problems, such as regression, interpolation, pattern recognition, medical, function approximation, floodforecasting, bioinformatics, engineering, software engineering, and business applications. Functional network outperforms the most common existing data mining schemes with stable and reliable performance, see Castillo et al. (1999, 2001, 2008), El-Sebakhy (2004, 2009, 2010), and El-Sebakhy et al. (2007, 2009), and the references therein for more details.

Yet, functional networks intelligence system framework has been only utilized in some specific oil and gas industry applications, namely, predicting reservoir fluids PVT properties, Al-Bokhitan (2007) and rock mechanical parameters for hydrocarbon reservoirs, El-Sebakhy et al. (2010). The motivation behind this paper is to propose functional networks as a new intelligence paradigm for permeability prediction from well logs in a hydrocarbon reservoir. The built calibration model will be developed and tested based on the use of distinct known reservoir characteristics databases. The methodology and implementation processes of the functional networks framework and the related notions with explanatory examples are briefly summarized below:

3.2. Functional networks predictive modeling: methodology and implementations

The goal of predictive data mining is to develop an intelligent dynamical model for either continuous or categorized outcomes to determine the relationship among the set of input and output variables of a given dataset $[X_1, X_2, \dots, X_p; Y]$, where X is n -by- p matrix of input variables and Y is n -by-1 vector of the output variables. Let us consider the nonlinear model, $g(Y) = f(X_1, X_2, \dots, X_p)$. Assuming that g is an invertible (one-to-one and onto) function, then this nonlinear model can be written as: $Y = g^{-1}(f(X_1, X_2, \dots, X_p)) = h(X_1, X_2, \dots, X_p)$. Therefore, we are interested in (i) Discovering the structure of both functions: f and g in the first part or (ii) Modeling Y as a nonlinear function of $[X_1, \dots, X_p]$ in the second part.

The objective in the case of developing a predictive model for a continuous outcome, is to model Y using a nonlinear function of numerous predictor variables, which is equivalent to approximate the functions $f(X_1, \dots, X_p)$ or $h(X_1, \dots, X_p)$ by linear combinations of set of basis (set of independent families), that is,

$$h(X_1, X_2, \dots, X_p) = \sum_{r_1}^{q_1} \sum_{r_2}^{q_2} \dots \sum_{r_p}^{q_p} C_{r_1 r_2 \dots r_p} \phi_{r_1}(X_1) \phi_{r_2}(X_2) \dots \phi_{r_p}(X_p), \quad (4)$$

where $C_{r_1 r_2 \dots r_p}$ are the parameters in the model; $\Phi_s = \{\phi_{r_s}(X_s) : r_s = 1, 2, \dots, q_s\}$ are families of independent families, and q_s is the number of elements in Φ_s for $s = 1, 2, \dots, p$.

To design a functional network that corresponds to the problem at hand, we define the neuron functions as: Assume that we have a set of nodes, $\mathbf{X} = \{X_1, \dots, X_p\}$ and each node X_i is associated with a variable X_i . The neuron function over a set of nodes X is a tuple $U = \langle X, f, Z \rangle$, where X is a set of the input nodes, f is a processing function and Z is the set of output nodes, such that $Z = f(X)$, where X and Z are two non-empty subsets of \mathbf{X} .

To make it simple to the reader, we discuss the functional networks example in Fig. 3. This graph illustrates the representation of a functional network for data set $\mathbf{X} = \{X_1, \dots, X_7\}$ of seven nodes, see Castillo et al. (1999, 2001, 2008), El-Sebakhy et al. (2007, 2010), and El-Sebakhy (2004, 2009, 2010) for more details. This functional network consists of the following:

- Several layers of storing units, like a layer having the input data set (x_i ; $i = 1, 2, 3, 4$) while another contains the output data (x_7). There can be one or several layers for storing intermediate information (x_5 and x_6).
- One or several layers for the processing units that evaluate the set of input values and deliver a set of output values (f_i).
- A set of directed links.

In general, functional networks extend the standard neural networks by allowing neuron functions f_i to be not only true multi-argument and multivariate functions, but to be different and learnable, instead of fixed functions. In addition, the neuron functions in functional networks can be any basis functions (family of linearly independent functions), namely, polynomial, Volterra polynomial, exponential, radial basis, B-spline, and Fourier, that have to be estimated during the learning process. According to the universal approximation theorem, Haykin (1994), there are a finite number of linearly independent functions (basis), so that functional networks can approximate the nonlinear function precisely. Furthermore, functional networks allow connecting neuron outputs, forcing them to be coincident, Castillo et al. (1999, 2001, 2008), El-Sebakhy (2004, 2009, 2010), and El-Sebakhy et al. (2007, 2010). The functional networks use two types of learning: (a) structural learning and (b) parametric learning. In the structural learning, the initial topology of the networks based on some properties available to the designer is arrived at, and finally a simplification is made using functional equations. In parametric learning, usually the activation functions are estimated considering the combination of “basis” functions based on the least square, steepest descent and mini-max methods, Cuddy (1997). In this study, the least square method for estimating activation functions has been used. One can choose different optimization criteria based on his interest. Development of a typical model based on functional networks involves several steps described below:

- Define the problem by specifying the initial topology based on the domain of the problem in hand.
- Simplify the chosen initial architecture using functional equations and the equivalence concept. Check the uniqueness

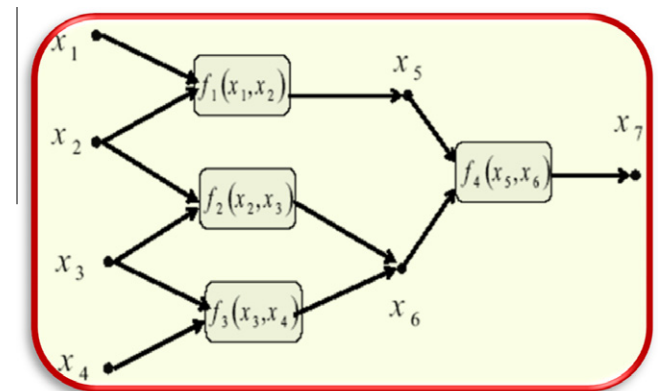


Fig. 3. Typical functional networks architecture.

condition of the desired architecture, Castillo et al. (1999, 2001, 2008), and El-Sebakhy (2004, 2009, 2010).

- Gather the required data and handle multicollinearity problem along with quality control checks.
- Develop the learning procedures and training algorithm based either on structure or on parametric learning by considering the combinations of linear independent functions, $\Phi_s = \{\phi_{r_s}(x_{is}) : r_s = 1, 2, \dots, q_s\}$, $s = 1, \dots, p$ to approximate the neuron functions, $f_t(x_{is}) \doteq \sum_{r_s=1}^{q_s} a_{r_s} \phi_{r_s}(x_{is})$, for all t ; where the coefficients a_{r_s} are the parameters of functional networks needed to be estimated. The most popular linearly independent functions in literature are: $\Phi = \{1, X, \dots, X^m\}$, or $\Phi = \{1, e^X, e^{-X}, \dots, e^{mX}, e^{-mX}\}$, or $\Phi = \{1, \sin(X), \cos(X), \dots, \cos^l(X), \sin^l(X)\}$, $m = 2l$; where m is the number of elements in the combination of sets of linearly independent function. The unknown parameters, a_{r_s} , can be learned using one of the known optimization (loss criterion) techniques, such as *least squares*, *conjugate gradient*, *iterative least squares*, *minimax*, or *maximum likelihood estimation*.
- Select the best model and validate it. The selection is based on the minimum description length and some other quality measurements, such as correlation coefficients and root-mean-squared errors. One can use the well known selection schemes, such as exhaustive selection, forward selection, backward elimination, backward forward selection, and forward backward elimination. Once the performance of functional networks model is satisfactory, the model is ready for use in predicting unseen datasets from real-world applications.

If we use the lower case letters $x_i \equiv (x_{i1}, \dots, x_{ip})$ to refer to the values of the i th observation of the input variables and the desired output may be: (i) $Y \in R$ for continuous prediction (forecasting problems) and (ii) $Y \subseteq R$ for classification problems, then for predictor variables, $X_j \equiv (x_{1j}, \dots, x_{nj})^T$ for $j = 1, \dots, p$, and the response variable, $Y = (y_1, \dots, y_n)^T$, the goal is one of the following main problems:

1. Forecasting continuous outcome is to determine the linear/non-linear relationship among the output and the input variables using one of the following equations:

$$g(y_i) = f(x_{i1}, \dots, x_{ip}) + \varepsilon_i; \quad i = 1, 2, \dots, n; \quad (5)$$

or

$$y_i = f(x_{i1}, \dots, x_{ip}) + \varepsilon_i; \quad i = 1, 2, \dots, n. \quad (6)$$

2. Classification problem, we use $y_i = k$ to refer to group A_k , where the symbol π_{ik} is used as the probability that observation i falls in group A_k , $\pi_{ik} = P(\mathbf{x}_i \in A_k | x_{i1}, \dots, x_{ip})$ or $\pi_{ik} = P(y_i = k | x_{i1}, \dots, x_{ip})$, where $\pi_{ik} \geq 0$; $k = 1, \dots, c$; $i = 1, \dots, n$; and $\sum_{k=1}^c \pi_{ik} = 1$. The goal in estimating the category outcome is to determine the linear/nonlinear relationship among the output and the input variables, that is,

$$g(\pi_{ik}) = f(x_{i1}, \dots, x_{ip}) + \varepsilon_i; \quad k = 1, \dots, c; \quad i = 1, 2, \dots, n; \quad (7)$$

or

$$\pi_{ik} = f(x_{i1}, \dots, x_{ip}) + \varepsilon_i; \quad k = 1, \dots, c; \quad i = 1, 2, \dots, n. \quad (8)$$

Therefore, in predictive data mining modeling for continuous prediction, we are interested in two different problems, namely, (i) Discovering the structure of transformations f and g in Eq. (5) and (ii) Modeling Y as a nonlinear function, $f(x_{i1}, \dots, x_{ip})$ in (6) or (8). If we propose some simple mathematical calculus, the function $f(x_{i1}, \dots, x_{ip})$ in model (6) can be rewritten as:

$$f(x_{i1}, \dots, x_{ip}) = \sum_{j=1}^p h_{ij}(x_{ij}) + \sum_{k=LL_1}^{UL_2} h_{kj_1}(x_{ij_1}) h_{kj_2}(x_{ij_2}) + \dots + \sum_{k=LL_{p-1}}^{UL_p} h_{kj_1}(x_{ij_1}) h_{kj_2}(x_{ij_2}) \dots h_{kj_p}(x_{ij_p});$$

where $j_1; j_2; \dots; j_p = 1, 2, \dots, p$ and $j_1 \neq j_2 \neq \dots \neq j_p$;

$$LL_s = \sum_{l=0}^s \binom{p}{l} \text{ and } UL_s = \sum_{m=1}^t \binom{p}{m};$$

for $s = 1, \dots, p-1$ and $t = 2, \dots, p$. (9)

For illustrative purposes, but without loss of generality, we consider the explanatory (predictor) variables ($p = 2$) and using polynomial families, $\Phi_s = \{1, X_s, \dots, X_s^q\}$; $s = 1, 2$ (where number of functions in these families, $q_s = q + 1$); then $f(x_{i1}, x_{i2}) = \sum_{j=1}^2 h_{ij}(x_{ij}) + \sum_{k=3}^3 h_{k1}(x_{i1}) h_{k2}(x_{i2})$. Therefore, model (6) can be written as:

$$y_i = h_{11}(x_{i1}) + h_{22}(x_{i2}) + h_{31}(x_{i1}) h_{32}(x_{i2}) + \varepsilon_i; \quad i = 1, 2, \dots, n. \quad (10)$$

where $h_{12}(x_{i2}) = 1$ and $h_{21}(x_{i1}) = 1$. Fig. 6a shows the architecture of the functional networks that is corresponding to model (10) with two explanatory variables. Suppose that each one of the neuron functions is approximated using the polynomial functions, $\Phi_s = \{1, X_s, X_s^2, \dots, X_s^q\}$, such that the neuron function $h_{11}(x_{i1})$ is the only neuron that contains the constant term. The rest of the neurons do not contain the constant term.

Assuming that in the range of the observation studies, the non-linear Eq. (4) provide an acceptable approximation to the true relation between the response and predictor variables, then each observation in D and model (6) can be written as:

$$y_i = \sum_{r_1=1}^{q_1} \sum_{r_2=1}^{q_2} \dots \sum_{r_p=1}^{q_p} C_{r_1 r_2 \dots r_p} \phi_{r_1}(x_{i1}) \phi_{r_2}(x_{i2}) \dots \phi_{r_p}(x_{ip}) + \varepsilon_i; \quad i = 1, 2, \dots, n; \quad (11)$$

where $C_{r_1 r_2 \dots r_p}$ are the parameters in the model, $\Phi_s = \{\phi_{r_s}(x_{is}) : r_s = 1, 2, \dots, q_s\}$ are families of linearly independent functions, and q_s is the number of elements in Φ_s for $s = 1, 2, \dots, p$. If the families of linearly independent functions $\Phi_s = \{\phi_{r_s}(x_{is}) : r_s = 1, 2, \dots, q_s\}$ are known, we only need to learn the coefficients $C_{r_1 r_2 \dots r_p}$. Since the errors in models (11) are additives, then we use a discrepancy measure to minimize the errors function, such as least squares and minimax. In this research, we apply the least squares optimization technique, that is

$$Q = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n \left[y_i - \sum_{r_1=1}^{q_1} \sum_{r_2=1}^{q_2} \dots \sum_{r_p=1}^{q_p} C_{r_1 r_2 \dots r_p} \phi_{r_1}(x_{i1}) \phi_{r_2}(x_{i2}) \dots \phi_{r_p}(x_{ip}) \right]^2. \quad (12)$$

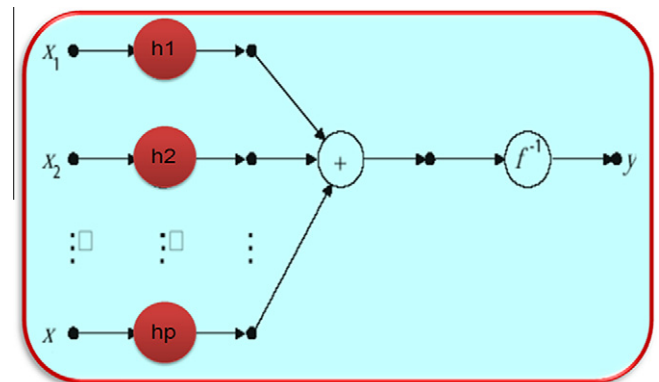


Fig. 4. Additive functional networks architecture.

The minimum is obtained by solving the following linear system of equations, where $C_{r_1 r_2 \dots r_p}$ are the unknowns, for $r_s = 1, 2, \dots, q_s$ and $s = 1, 2, \dots, p$:

$$\frac{\partial Q}{\partial C_{r_1 r_2 \dots r_p}} = -2 \sum_{i=1}^n \left[y_i - \sum_{r_1=1}^{q_1} \sum_{r_2=1}^{q_2} \dots \sum_{r_p=1}^{q_p} C_{r_1 r_2 \dots r_p} \phi_{r_1}(x_{i1}) \phi_{r_2}(x_{i2}) \dots \phi_{r_p}(x_{ip}) \right] \times \phi_{r_1}(x_{i1}) \phi_{r_2}(x_{i2}) \dots \phi_{r_p}(x_{ip}) = 0; \quad r_1 = 1, 2, \dots, q_1, \\ r_2 = 1, 2, \dots, q_2; \dots; r_p = 1, 2, \dots, q_p. \quad (13)$$

We note that the number of parameters in the model (11) is $\prod_{s=1}^p q_s$. In order to reduce the number of parameters, we substitute the formula (7) for the function $h(x_{i1}, \dots, x_{ip})$ in model (6). Therefore, model (6) can be rewritten as:

$$y_i = \sum_{j=1}^p h_{ij}(x_{ij}) + \sum_{k=LL_1}^{UL_2} h_{kj_1}(x_{ij_1}) h_{kj_2}(x_{ij_2}) + \dots + \dots \\ + \sum_{k=LL_{p-1}}^{UL_p} h_{kj_1}(x_{ij_1}) h_{kj_2}(x_{ij_2}) \dots h_{kj_p}(x_{ij_p}) + \varepsilon_i; \\ \text{where } j_1, j_2, \dots, j_p = 1, 2, \dots, p \text{ and } j_1 \neq j_2 \neq \dots \neq j_p; \\ i = 1, 2, \dots, n; \quad LL_s = \sum_{l=0}^s \binom{p}{l} \text{ and } UL_s = \sum_{m=1}^t \binom{p}{m}; \\ \text{for } s = 1, \dots, p-1 \text{ and } t = 2, \dots, p. \quad (14)$$

In this research, we present two distinct types of functional networks: (i) *Generalized Associativity* and (ii) *Separable* models to approximate the unknown functions, $f(x_{i1}, \dots, x_{ip})$. A brief description of these two types of functional networks is given below; Castillo et al. (1999), and El-Sebakhy (2004, 2009).

1. The Generalized Associativity: $f(x_{i1}, \dots, x_{ip}) = \sum_{j=1}^p h_j(x_{ip})$ which makes model (5) to lead to the additive model:

$$g(y_i) = \sum_{j=1}^p h_j(x_{ip}) + \varepsilon_i; i = 1, 2, \dots, n. \quad (15)$$

The corresponding architecture is shown in Fig. 4.

2. The Separable Functional Networks Model is a more general form for the unknown function, $f(x_{i1}, \dots, x_{ip})$ in the form $f(x_{i1}, \dots, x_{ip}) = \sum_{r_1=1}^{q_1} \dots \sum_{r_p=1}^{q_p} C_{r_1 r_2 \dots r_p} \phi_{r_1}(x_{ir_1}) \dots \phi_{r_k}(x_{ir_p})$ which makes model (6) to lead to the additive model:

$$y_i = \sum_{r_1=1}^{q_1} \dots \sum_{r_p=1}^{q_p} C_{r_1 r_2 \dots r_p} \phi_{r_1}(x_{ir_1}) \dots \phi_{r_k}(x_{ir_p}) + \varepsilon_i; i = 1, 2, \dots, n; \quad (16)$$

where $C_{r_1 \dots r_p}$ are unknown parameters expressed in terms of functions ϕ_s , where $\Phi_s = \{\phi_{r_s} : r_s = 1, \dots, q_s\}$, and $s = 1, 2, \dots, p$, are linearly

independent. An example of the functional network for $p = 2$ and $q_1 = q_2 = q$ is shown in Fig. 5. Eqs. (15) and (16) are functional equations since their unknowns are functions and their corresponding functional networks in Figs. 4 and 5 are their graphical representations.

3. The General functional networks model: Several other alternatives are possible for selecting the initial topology of functional networks; see Castillo (1998) for more details. Nevertheless, when the user has no information about the problem at hand, a general model is model (5) with $f(x_{i1}, \dots, x_{ip}) = \sum_{r_1=1}^{q_1} \dots \sum_{r_p=1}^{q_p} C_{r_1 r_2 \dots r_p} \phi_{r_1}(x_{ir_1}) \dots \phi_{r_k}(x_{ir_p})$, which leads to the model:

$$g(y_i) = \sum_{r_1=1}^{q_1} \dots \sum_{r_p=1}^{q_p} C_{r_1 r_2 \dots r_p} \phi_{r_1}(x_{ir_1}) \dots \phi_{r_k}(x_{ir_p}) + \varepsilon_i; i = 1, 2, \dots, n; \quad (17)$$

where $C_{r_1 r_2 \dots r_p}$ in (17) are the unknown parameters of the functional networks modeling system. The corresponding functional network architecture for $p = 2$ and $q_1 = \dots = q_p = q$, is shown in Fig. 6b. Note that since there are no converging arrows in the functional networks associated with the general models (see Figs. 5 and 6), simplification is not possible in these cases.

In order to obtain h_1, h_2, \dots, h_p in (15) or (17), $g(y_i)$ and each $h_j(x_j)$ for $j = 1, 2, \dots, p$ is approximated by a linear combination of sets of linearly independent functions ϕ_{js} defined above, that is

$$h_j(x_j) = \sum_{s=1}^{q_j} a_{js} \phi_{js}(x_j); j = 1, 2, \dots, p, g(y_i) = \sum_{s_0=1}^{q_0} b_{s_0} \phi_{s_0}(y_i). \quad (18)$$

The problem is then reduced to estimate the parameters, b_{s_0} and a_{js} , for all j and s . Parameters $C_{r_1 r_2 \dots r_p}$ are linked with a_{js} and evaluated subsequently; see Castillo et al. (1999, 2008), El-Sebakhy et al.

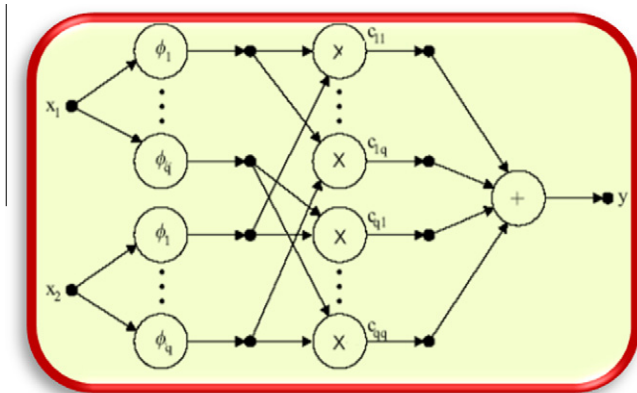


Fig. 5. Separable functional networks model with two inputs and q neuron functions.

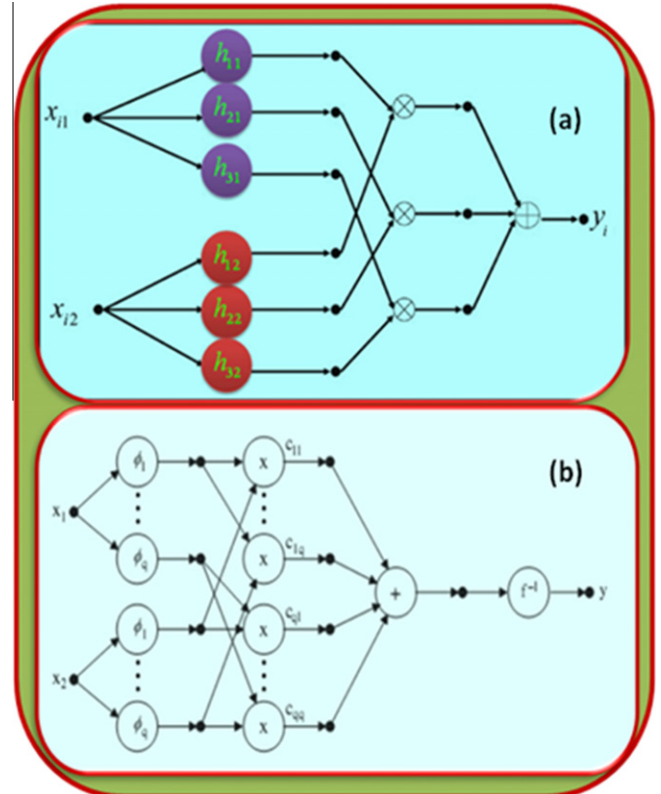


Fig. 6. (a) General separable functional networks architecture and (b) general functional networks model (17) involving transformation of the response target.

(2007), and El-Sebakhy (2009, 2009) for more details. In general, by feeding appropriate input data and applying system identification to study the defect prone classes identification, one can customize the characteristics of input values according to the desired output. In this paper, the least squares criterion is used to estimate the parameters. However, an additive model requires additional constraints to guarantee uniqueness. Alternatively, one can choose a different optimization criterion based on his interest. The main advantage in choosing the least squares method is that it leads to a linear system of equations.

3.3. Non-Polynomial fitting example based on functional networks

Suppose that we generate data of size 200 from the model $\log(y) = x_1 + x_2 + \varepsilon$, where x_1 and x_2 are uniformly distributed in $[0, 1]$, $x_1 \sim U[0, 1]$, $x_2 \sim U[0, 1]$, and ε is normal with zero mean and 0.01 scatter, $\varepsilon \sim N[0, 0.01]$ or $\varepsilon \sim N[0, 0.1]$. The scatter plot of y against $x_1 + x_2$ is shown in Fig. 7a with 50 observations for both models. To determine the estimated model using functional networks using the proposed training algorithm with separable functional networks model, we choose the polynomial families of

linearly independent functions $\phi_1 = \{1, y, y^2\}$, $\phi_2 = \{1, x_1, x_1^2, x_1^3\}$, and $\phi_3 = \{1, x_2, x_2^2, x_2^3, x_2^4\}$, and tolerance values equal to 0.1 or 1. Therefore, by applying the proper training algorithm on the initial architecture of the corresponding functional networks, we obtain the estimated functional networks model with $\varepsilon \sim N[0, 0.01]$ with backward selection criterion: $-0.9498 + 1.1930y - 0.2071y^2 + 0.0141y^3 = 0.1380x_1 - 0.1532x_1^3 + x_2$. The value of the mean-squared errors is 0.00000672.

When data are simulated without error (no uncertainty associated with the data sample), the true functional networks model is selected at the end of backward procedure if tolerance = 1. The predictive functional networks model with polynomial function is written as: $-0.8280 + 1.1930y - 0.2071y^2 + 0.0141y^3 = x_1 + x_2$. Fig. 7b shows the scatter plots of both $\log(y)$ and the predicted model: $-0.8280 + 1.1930y - 0.2071y^2 + 0.0141y^3$ against $x_1 + x_2$. We note that both graphs are identical.

We conclude that the estimated form of the functional networks is simpler than its given model. The output values layer does not utilize all the input nodes. This is reasonable because if additional different terms are considered in these models, then the tolerance value would influence the results significantly.

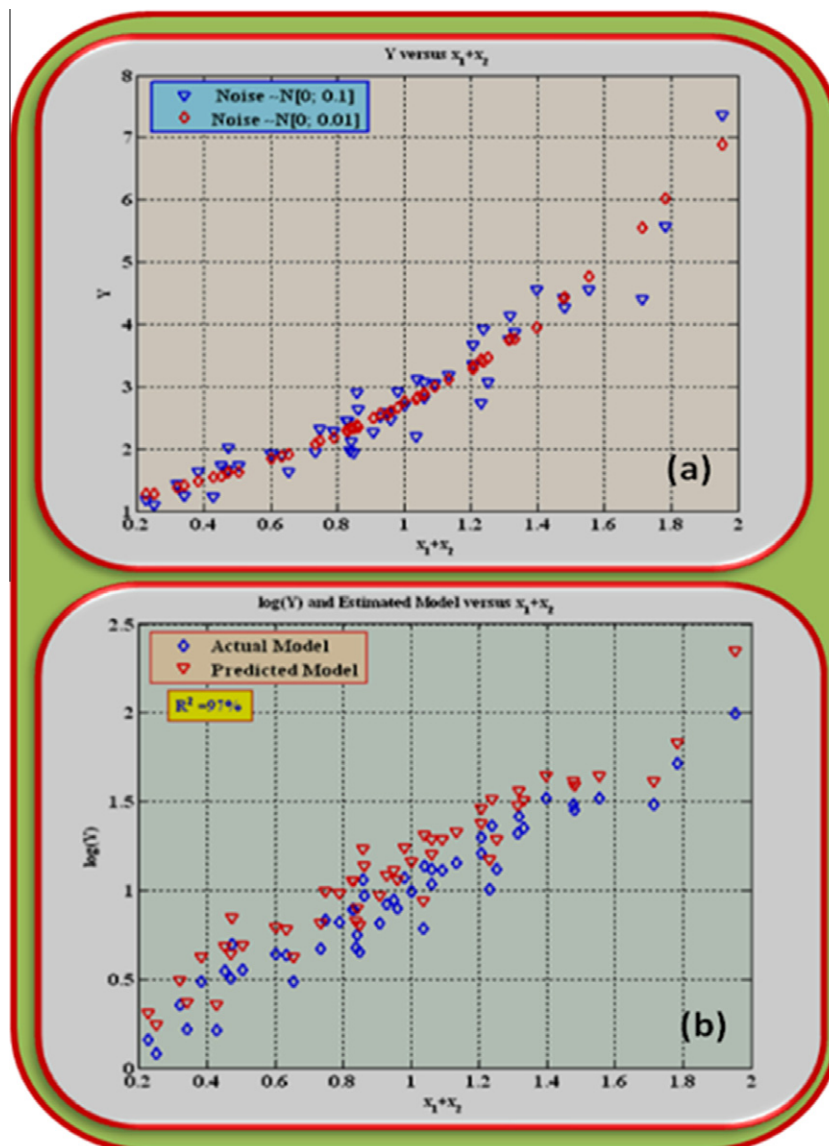


Fig. 7. (a) Generated data for the given model with noise: $\varepsilon \sim N[0, 0.1]$ and $\varepsilon \sim N[0, 0.01]$ and (b) Actual and predicted models versus $x_1 + x_2$.

4. Statistical and quality measures

4.1. Initialization

To implement the functional networks or any other statistics and data mining modeling techniques, the available dataset has to be divided into two subsets (training and the testing), which can be achieved using either leave one out or k-fold cross validation or random selection procedures. Through the implementation, a stratified sampling technique is used to make sure that the same pattern is applied as in the original data, Hush and Horne (1993). The training set is used to build up the networks model while the testing set is used to evaluate (deploy) the predictive modeling capabilities in forecasting new unseen data. The predictive performance of functional networks, statistical regression, neural networks, and fuzzy logic systems were tested using real-world permeability data.

Through the implementations we utilized three distinct functional network architectures, namely: *associativity*, *separable* or *general* functional networks model explained above with polynomial of degree at most two or three (family of linearly independent functions (basis)). The multilayer perceptron feed-forward networks is used with pure linear and sigmoid activation neuron functions and two or three hidden layers. The adaptive neuro-fuzzy inference system is utilized with both subtractive grid partition and with both Gaussian and triangular membership functions of at most two rules. The initial parameters for radial basis functions are the centers and distances: $\sigma = \frac{d_{\max}}{\sqrt{k}}$; where d_{\max} is the maximum of Euclidean distances between the centers and each observation and $k = n^{3/8}$ is the number of clusters for n observations.

The input datasets were normalized to interval of [0,1] using the formula: $x_i^{(new)} = \frac{(x_i^{(old)} - \min(x_i))}{(\max(x_i) - \min(x_i))}$; $i = 1, \dots, n$; which leads to a set of input variables that are independent of their measurement units. The normalized inputs are then used for functional networks and the utilized data mining techniques. Different quality measures can be used to judge the performance of the developed data mining models.

4.2. The common quality measures

To evaluate the developed models, we compute several quality measures, namely, the *absolute residual* (*AbsRes*) which is the absolute difference between the actual and desired target values, the *magnitude of relative error* (*MRE*), *sum of the absolute residuals* (*SumAbsRes*), the *median of the absolute residuals* (*MedAbsRes*), and the *standard deviation of the absolute residuals* (*SDAbsRes*). These measures have been chosen because they provide the user with significant quality assessment tools in developing a proper predictive permeability model. For instance, the *MedAbsRes* measures the central tendency of the residual distribution and is chosen to be a measure of the central tendency because the residual distribution is usually skewed in most of the core and wireline datasets used in reservoir characterization. In addition, the *SDAbsRes* measures the dispersion of the residual distribution. The most common statistical quality measures can be written mathematically as follows:

- **Root mean squares error (RMSE):** It measures the data dispersion around zero deviation:

$$RMSE = \left[\frac{1}{n} \sum_{i=1}^n E_i^2 \right]^{1/2} \quad (19)$$

where E_i is a relative deviation of an estimated value from an experimental input data sets.

$$E_i = \left[\frac{(y)_{\exp} - (y)_{est}}{(y)_{\exp}} \right] \times 100; i = 1, \dots, n; \quad (20)$$

- **Correlation coefficient:** It represents the degree of success in reducing the standard deviation by regression analysis, defined as:

$$r = \sqrt{1 - \frac{\sum_{i=1}^n [(y)_{\exp} - (y)_{est}]^2}{\sum_{i=1}^n [(y)_{\exp} - \bar{y}]^2}}, \quad (21)$$

where $\bar{y} = \frac{1}{n} \sum_{i=1}^n [(y)_{\exp}]_i$.

- **Mean absolute percentage error:** The mean absolute percentage error (MAPE) of the forecasted values with respect to the actual core permeability:

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left[\frac{|(y)_{est} - (y)_{act}|}{(y)_{act}} \right], \quad (22)$$

where $(y)_{est}$ is the forecasted value of permeability and $(y)_{act}$ is the actual core permeability; n is the number of cases used in the test sample.

- **Standard deviation:** The standard deviation is a measure of how widely values are dispersed from the average value (the mean). Standard deviation is based on the entire population given as arguments. The standard deviation is calculated using the “biased” or “n” method. It is defined by:

$$SD = \left(\frac{n \sum_{i=1}^n (E_i)^2 - (\sum_{i=1}^n E_i)^2}{n^2} \right)^{1/2}. \quad (23)$$

4.3. Feature selection techniques

To maximize the performance of the developed predictive modeling schemes, we utilized different feature selection criteria. The following is a brief description of each one of these utilized feature selection procedures:

- **Forward selection:** It starts with an empty model. The variable that has the smallest P value when it is the only predictor in the regression equation is placed in the model. Each subsequent step adds the variable that has the smallest P value in the presence of the predictors already in the equation. Variables are added one-at-a-time as long as their P values are small enough, typically less than 0.05 or 0.10.
- **Backward elimination:** It starts with all of the predictors in the model. The variable that is least significant that is, the one with the largest P value is removed and the model is refitted. Each subsequent step removes the least significant variable in the model until all remaining variables have individual P values smaller than some value, such as 0.05 or 0.10.
- **Stepwise regression:** This approach is similar to forward selection except that variables are removed from the model if they become non significant as other predictors are added.

The backwards elimination technique has an advantage over forward selection and stepwise regression because it is possible for a set of variables to have considerable predictive capability rather than any individual subset. Forward selection and stepwise regression will fail to identify them because sometimes variables do not predict well individually. Given that backward elimination starts with everything in the model, the joint predictive capability will be seen.

5. Implementations and comparative studies

Based on the investigated literature, there is no established correlation for permeability with the wireline log data. Therefore, it is

mandatory to develop a predictive intelligence modeling scheme to forecast such relationship using data mining from the available input data. The functional networks (FunNets) system associated with a proper topology was utilized to build the predictive model from well log data. A combination of the least-squares method, Lagrangian multipliers, and conjugate gradient descent technique for training different sets of linearly independent families (polynomials, exponential, Fourier, etc.) is applied to emulate a given training data set. In addition to grid partitioning, clustering of the input data was also done.

5.1. Handling core and wireline data: Cleaning data and preprocess implementations

The core data file is given in a txt file with extension *.las format* containing the depth and permeability information of the cores. We refine the data into two sets: one contains header text *depth* and *permeability* and the other contains the corresponding numeric values. The wireline data was of *.las format* as well with 11–13 predictors (input variables) apart from *depth*. Similarly, to acquire the data from the given wireline files, we divide the data into two sets containing header text and the numeric values; filter out the header information from the text data; arrange the numeric data according to the headers; segregate the relevant parameters along with its header information; read data from the given tops file (data containing the information regarding the layers and the number of core data points in that layer); and divide the data as per the tops. An intelligence cleaning strategy was created to extract and clean the noise/missing values in the provided input predictors within wireline data. We developed an intelligent data cleaning mechanism to remove the negative values (–9999.9999 for instance) within each of the provided predictors. This is to ensure that we do not lose too much data for developing the predictive permeability model. We interpolated the positive data for missing log values at the depths, where core data is available using both linear/nonlinear and *k*-nearest neighbor fitting mechanisms.

We utilized (i) an outlier identification process to extract the abnormal behavior record; (ii) principal component analysis to handle the collinearity among predictors; (iii) different clustering algorithms to investigate the data trend and accumulations; and (iv) determine and investigate the correlations between permeability and the provided predictor (logs) variables, and investigate the mutual correlations among all the provided predictor (logs) variables. Based on the correlation results shown in Table 1, out of these parameters only six predictors were chosen for predicting permeability based on numerous statistical and data mining predictive modeling schemes. For the sake of simplicity and space, we recorded the results of the most common predictive modeling

approaches, namely: statistical regression, multilayer feedforward neural networks, adaptive neuro-fuzzy logic inference system, and the new intelligence paradigm (functional networks intelligence system). The implementations were carried out through our developed intelligence predictive data mining intelligence system, utilizing a friendly graphical user interface (GUI) for: (i) input and clean data; (ii) data modeling; and (iii) data deployment.

A correlation analysis was carried out on the wireline data by correlating the core permeability (*k*) with individual wireline logs, namely, CT, DRHO, DT, DR, MSFL, NPFI, PHIT, RHOB, RT, SWT. As usual in the community of geoscientists and on the basis the correlation results, we take the logarithm (base 10) for CT, MSFL and permeability (*K* or core permeability values), since their range were found to be of several orders of magnitude. Taking logarithm gives better correlation between \log_{10} (MSFL) and \log_{10} (*K*) rather than with only MSFL. The correlation results are tabulated in Table 1. Significant improvement was observed in the correlation coefficients as shown in Table 1. Therefore, the wireline logs having correlation values greater than 0.5 were chosen as inputs for permeability prediction. The six wireline log input parameters chosen are DT, MSFL, NPFI, PHIT, RHOB, and SWT. \log_{10} was applied to MSFL and *k* throughout the modeling implementation process.

The provided data of each well was divided into two separate subsets, one for 'training' and the other for 'testing' using different criteria: (i) leave-one-out, (ii) ten-fold cross-validations, and (iii) random selection procedures for distributing the data points among training and testing sets (say for instance 70% of the data was used for training while the remaining 30% was used for testing).

We use the following abbreviations for the wireline log parameters for the sake of simplicity during the implementations, that is,

- N: NPFI or Neutron porosity log; P: PHIT or Total porosity log.
- M: MSFL or Micro-spherical formation resistivity log.
- R: RHOB or Bulk density log; S: SWT or Water saturation log.
- D or DT: Compressional velocity travel time for one foot of depth.
- PR: Combination of logs for P and R (as listed above).
- PM: Combination of logs for P, and M; PMR: Combination of logs for P, M, and R.
- PMRS: Combination of logs for P, M, R, and S; PMRSD: Combination of logs for P, M, R, S, and D.
- DMNPRS: Combination of logs for D, M, N, P, R, and S.

The implementation studies were conducted using three well log data sets, where we had only core data for two data sets to build the predictive data mining modeling approaches. Next, deploy and validate the built model on the new unseen wireline logs. We carry out comparative studies to check the efficiency of the new intelligence paradigm (FunNets) model versus the performance of statistical regression, multilayer feedforward neural networks, and adaptive fuzzy inference system.

We used different statistical quality measures to evaluate the built model and investigate its trend. We compare the trend of both predicted '*k*' and actual '*k*'; and compute R^2 , RMSE, standard deviations between both (k_a = actual permeability) and (k_p = predicted permeability). We detect and identify outliers (predicted values that lie far away from the actual data); a predicted point is considered to be an outlier if its value is out of range (maximum or minimum) of the actual data. Fig. 8a and b shows the actual permeability versus the predicted permeability using both functional networks with polynomial family of degree at most 2, ANFIS with subtractive clustering, and 10-fold cross validations using 'DMNPRS' logs. From the figures it is clear that the predictions of an entire well based on a functional networks intelligence system were found to be close to the actual core values.

Table 1
Correlation of both core permeability (*k*) and \log_{10} (*k*) with wireline parameters.

Well#	Well-1	Well-2	Well#	Well-1	Well-2
Kcore	1	1	\log_{10} (kcore)	1	1
phi-core	0.53	0.36	phicore	0.89	0.86
CT	0.14	0.32	CT	0.27	0.57
\log_{10} (CT)	0.16	0.25	\log_{10} (CT)	0.19	0.4
DRHO	–0.32	–0.17	DRHO	–0.44	–0.33
DT	0.46	0.38	DT	0.76	0.83
GR	–0.06	–0.12	GR	–0.09	–0.08
MSFL	–0.23	–0.35	MSFL	–0.5	–0.76
\log_{10} (MSFL)	–0.4	–0.48	\log_{10} (MSFL)	–0.74	–0.8
NPFI	0.48	0.34	NPFI	0.82	0.83
PHIT	0.49	0.38	PHIT	0.83	0.85
RHOB	–0.47	–0.39	RHOB	–0.8	–0.84
RT	–0.13	–0.03	RT	–0.13	–0.04
SWT	–0.17	–0.14	SWT	–0.52	–0.53

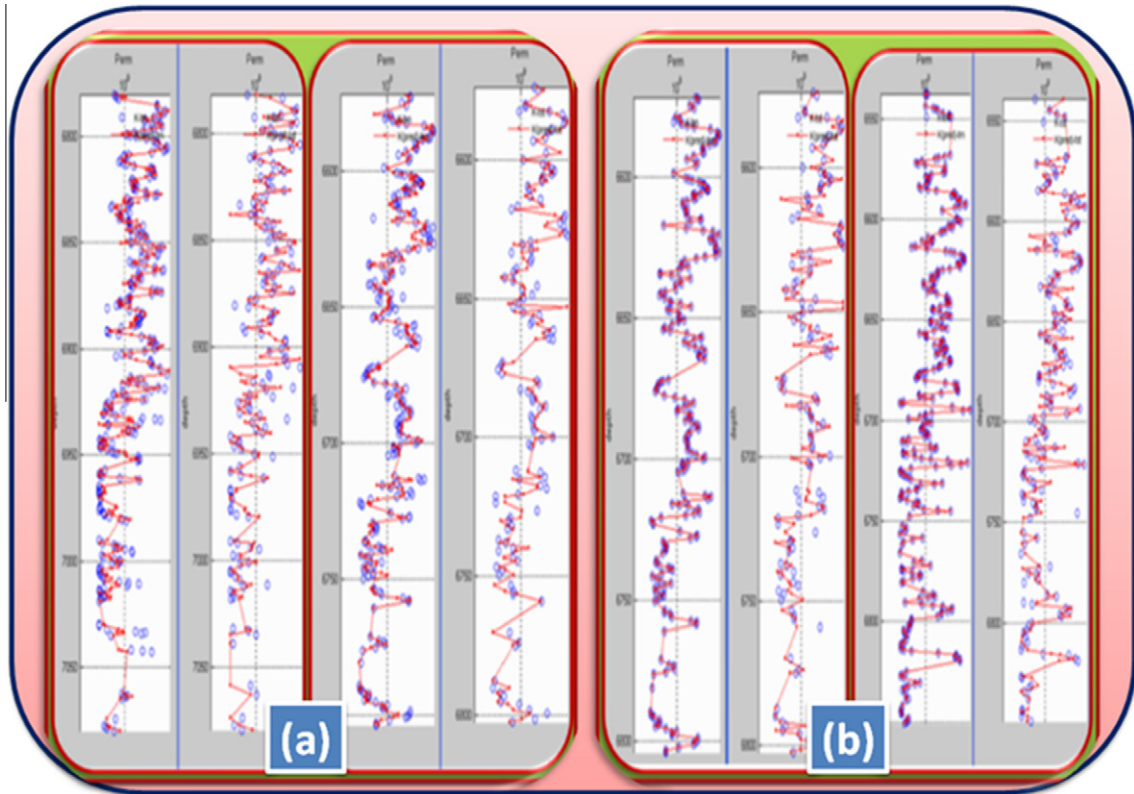


Fig. 8. 'kcore' and 'kpred' for Well-1 and Well-2 based on (a) neuro-fuzzy logic system with subtractive clustering and (b) functional networks with the second degree polynomial family using 'DMNPRS' logs with 10-fold selection criterion.

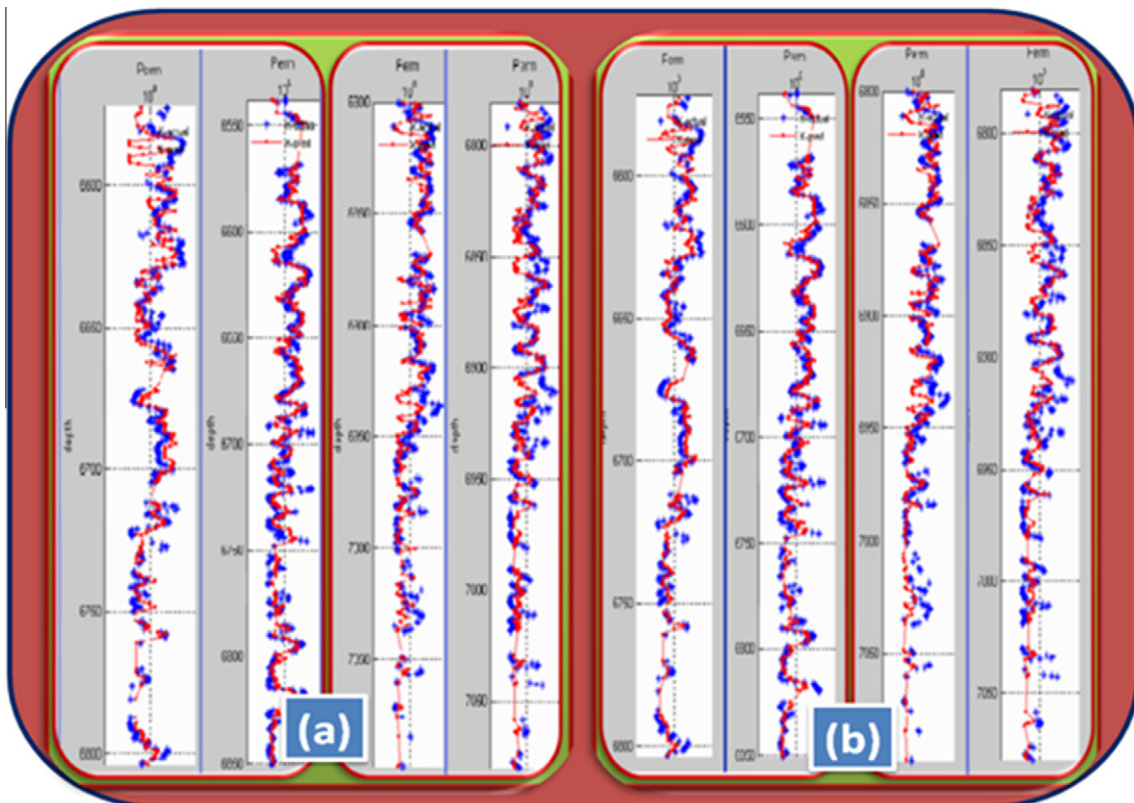


Fig. 9. 'kcore' and 'kpred' for Well-1 and Well-2 based on functional networks with the second degree polynomial family using (a) 'PHIT' log and (b) 'RHOB' log with 10-fold selection criterion.

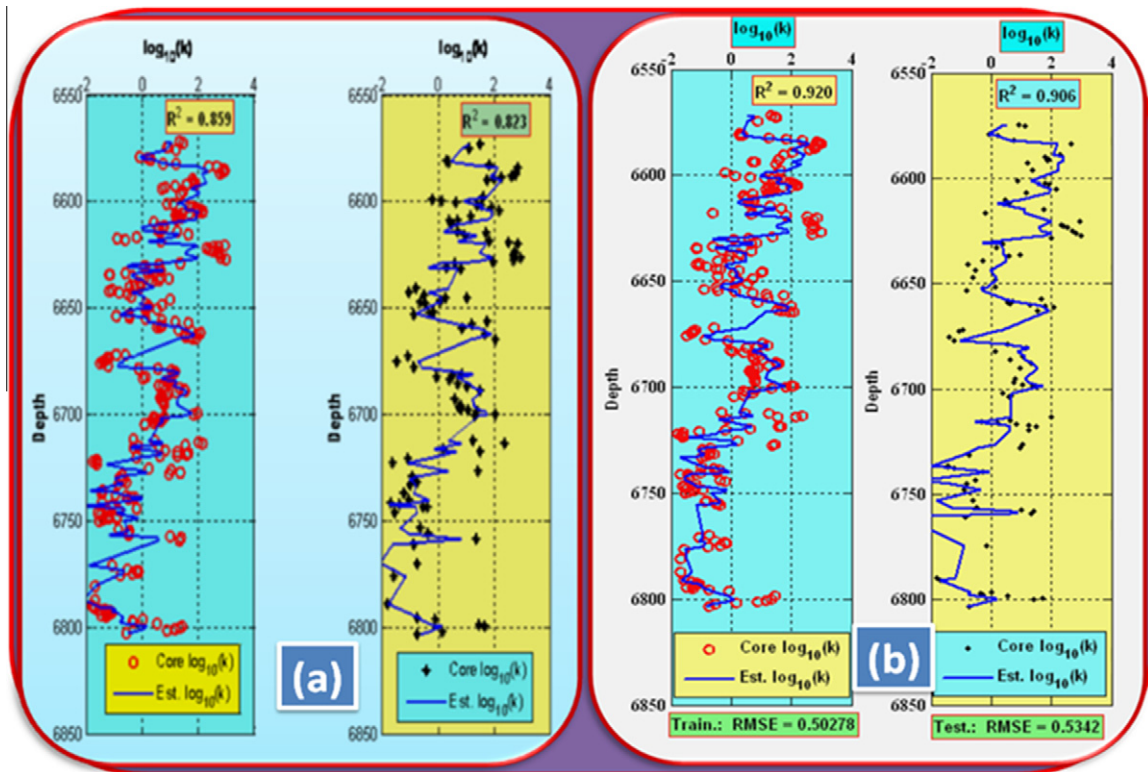


Fig. 10. Prediction of permeability based on (a) statistical regression and (b) functional networks with the second degree polynomial family with 10-fold cross validations using the specified six input variables.

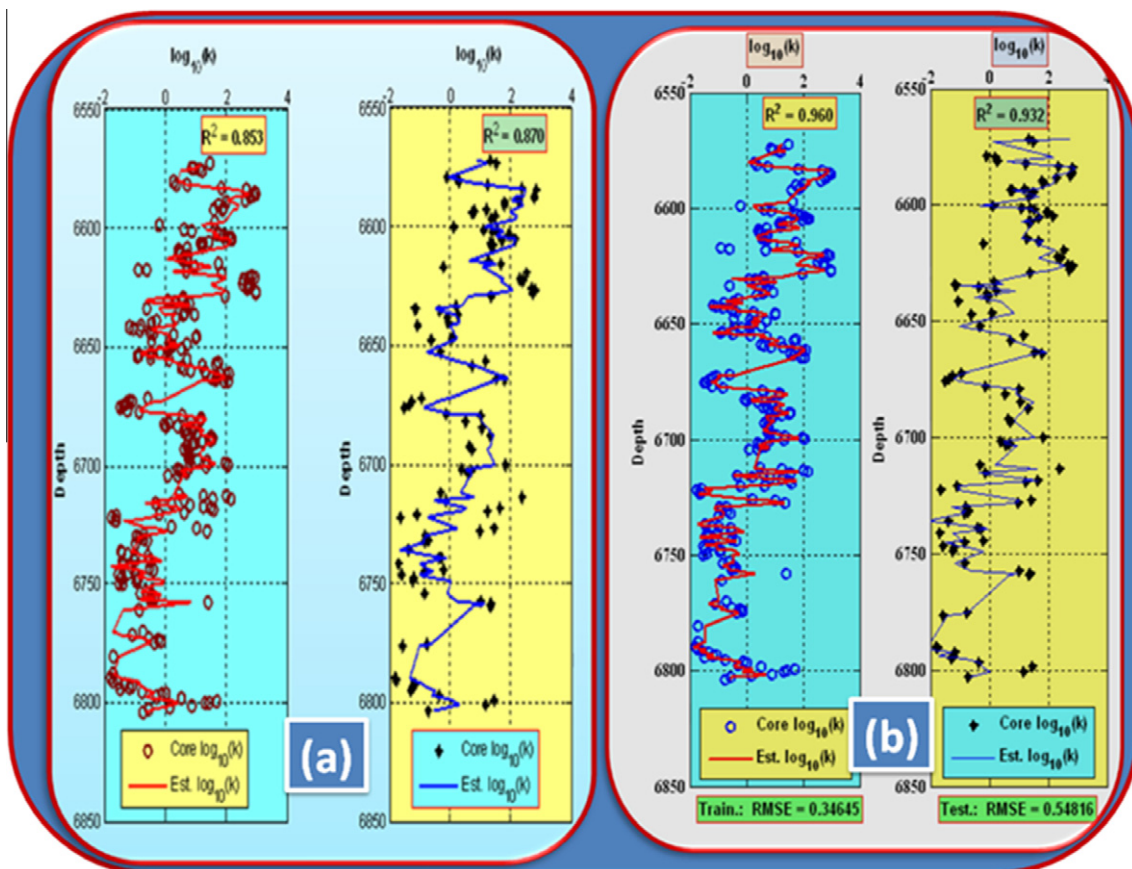


Fig. 11. Prediction of permeability based on (a) statistical regression and (b) functional networks with the second degree polynomial family with 10-fold cross validations using the specified all input variables.

5.2. Deployment and validation on different wells

We next consider the deployment field to investigate the capabilities of the functional networks intelligence system in predicting the permeability of new unseen wells based on the developed predictive models. Here, the entire well was considered for training. Testing was done using a new unseen well log within the same or different reservoir. The training well was selected as that having the maximum number of core data points, (Well No.1 in the present study) and the log combination 'DMNPRS' selected for the present study included six different well log inputs 'DT', 'MSFL (log10)', 'NPHI', 'PHIT', 'RHOB' and 'SWT'. These log combinations were chosen based on their correlation coefficients performance.

We built a predictive model using multiple wireline wells and tested it on a new unseen well. In addition, the number of input logs was reduced from six input variables, (DMNPRS) to four input variables; (NPRS) and FunNets were developed with two combined wells as a single input during the training, and then tested it on unseen new well data. In the beginning the effects of individual logs 'N', 'R' and 'S' on the predictions were investigated, followed by for the combination 'NRS'. Reduction in the errors, outliers and also in the abnormal behavior of the predicted trend generated was observed with these combinations as input during the model development.

Observing that the predictor variable "SWT" is too dependent on both restivity and porosity, we decide to exclude it from the

chosen list. In addition, as both "NPHI" and "PHIT" are highly correlated, to avoid the ill-conditioning (collinearity problem), we applied principal component analysis to exclude one of them. We excluded 'RHOB' when we have a model that have "DT" or "PHIT" as shown in Fig. 9a and b. We have investigated the functional networks intelligence predictive model versus feedforward neural networks, ANFIS, and statistical regression using different sets of combination of significant inputs, such as, "MSFL, DT, and PHIT (MDP)"; "MSFL, DT, and NPHI (MDN)"; and "MSFL and RHOB (MR)". The performances were improved, but not as accurate as with six or the entire set of inputs. We display some results in Fig. 9a and b. From the figures, it is clear that the individual wireline log input 'RHOB' gives better predictions than other input combinations. From these results it can be concluded that, we can estimate the permeability of any unknown new well based on the developed functional networks predictive models using a combination of data from other known wells as shown in Fig. 15.

6. Results and discussion

As stated in the previous section, the performance of functional networks intelligence system, and the other statics and data mining approaches were investigated based on two wells (Well- W_1 and Well- W_2). These wells were drilled in a Middle-Eastern Carbonate reservoir. Wireline logs from these wells included CT (electrical conductivity), DRHO (density), DT (sonic travel time), MSFL

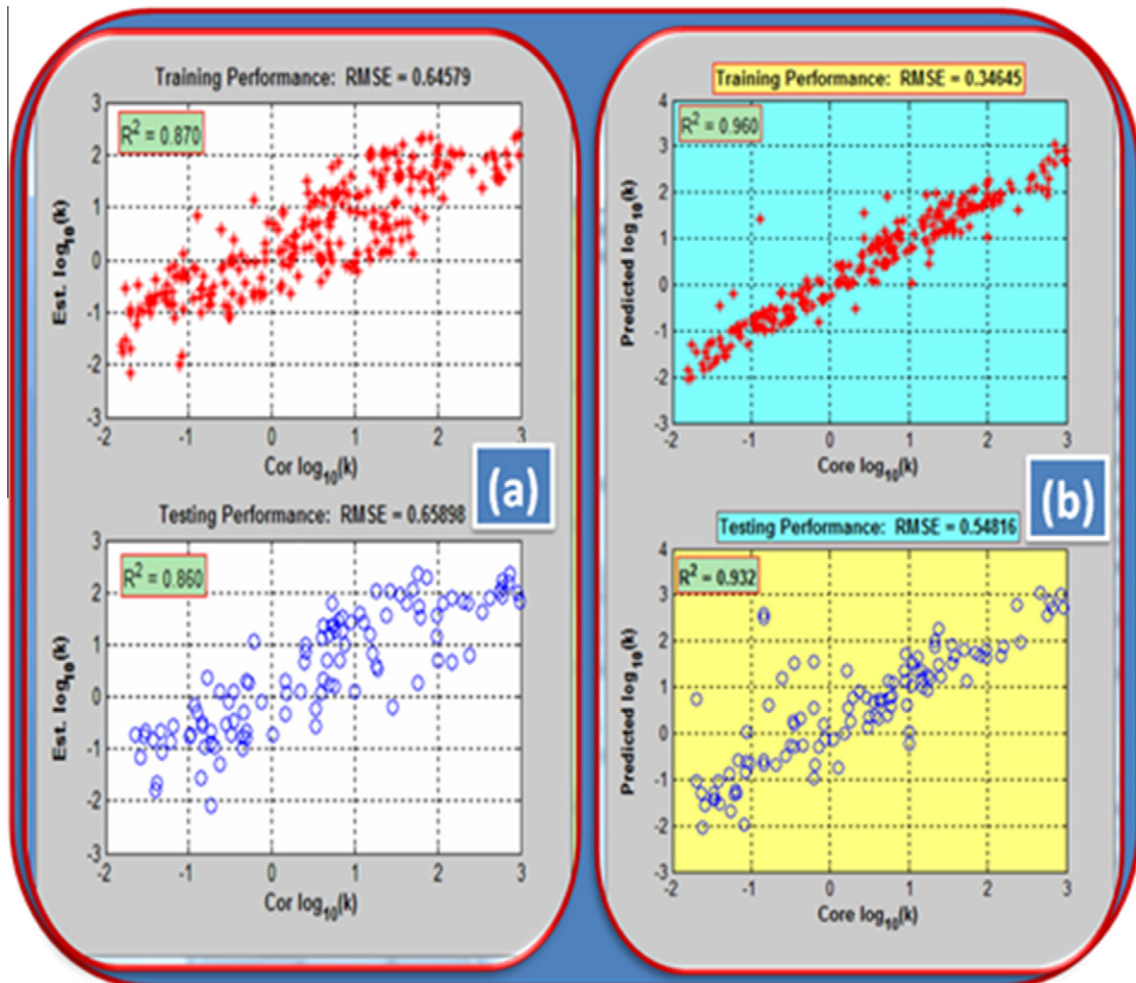


Fig. 12. Actual versus estimated permeability: (a) Statistical regression and (b) functional networks (second degree polynomial families) with 10-cross validation using all input variables.

(Micro spherically Focused Log), *NPHI* (Neutron porosity), *PHIT* (total porosity), *RHOB* (bulk density), *RT* (Resistivity), and *SWT* (water saturation). Laboratory based permeability values obtained from rock cores were used for both training and testing of the models. The total numbers of available core data points for Well-W₁ were 471 and for Well-W₂ were 431.

For sake of simplicity, the criteria used for evaluating the predicted values include the agreement between the predicted and the actual core permeability; the correlation between the predicted and the actual core permeability; and the root mean square of error (RMSE) between the actual and the predicted value.

The plots within Fig. 10a shows the scatter plot of statistical regression model results for the training and testing data, while the plots within Fig. 10b provides the corresponding functional networks model results. It can be noticed from these two plots that the functional networks prediction curve seems to pass through most of the target data points compared to its statistical regression counterpart, demonstrating the ability of functional networks with polynomial family of degree at most 2 with 10-fold cross validation to predict relatively more accurately. This is true for both training and testing as is evident from these plots.

Similar variations but with increased input variables are shown in Fig. 11a and b for statistical regression techniques and functional networks, respectively. During the training phase, both of these approaches predicted the permeability values reasonably close to the actual. But for the testing phase, the prediction trend shown by functional networks was more in agreement with the target data

compared to that of statistical regression. Also with an increasing number of input parameters, the prediction using functional networks seems to improve significantly during the testing phase as shown in Fig. 11b, while no such improvement is noticed with the statistical regression approach, Fig. 11a. It is clear that the number of inputs has a much more significant effect on the performance of functional networks than on that of statistical regression.

Fig. 12a and b show the cross plots between predicted and core permeability for statistical regression and functional networks during training and testing processes using all input data from all the available logs. It is clear from the cross plots that the functional networks provide a slope close to 1 (straight line) compared to that of statistical regression. Furthermore, increasing the number of input variables helps the functional networks to give more accurate prediction both during the training and the testing phase.

The overall performance of these two approaches was evaluated based upon the root mean square error (RMSE) and correlation factor (R^2). These values are listed in Table 2 for both training and testing using functional networks with polynomial families of degree at most 2 (basis), adaptive neuro-fuzzy logic inference systems with subjective clustering, multilayer feedforward neural networks with sigmoid activation function and three hidden layers, and statistical regression based on 10-fold cross validation for well W-1 data.

From the table it is clear that higher R^2 and lower RMSE values were obtained under all conditions with functional networks intelligence system technique. Furthermore, higher correlation and

Table 2
The correlation coefficient and RMSE values for the two models.

Set of input data	Training		Testing		Training		Testing	
	R^2 (%)	RMSE	R^2 (%)	RMSE	R^2 (%)	RMSE	R^2 (%)	RMSE
<i>Functional networks</i>					<i>Adaptive neuro-fuzzy logic inference</i>			
<i>Functional networks</i>					<i>Adaptive neuro-fuzzy logic inference</i>			
Six Variables	92	0.5	90.6	0.53	91	0.55	89	0.57
Entire wireline log	96	0.35	93.2	0.54	95	0.45	92	0.59
<i>Neural networks</i>					<i>Nonlinear regression</i>			
Six Variables	90	0.6	88	0.6	86	0.64	82	0.65
Entire wireline log	93	0.5	91	0.62	87	0.65	85	0.66

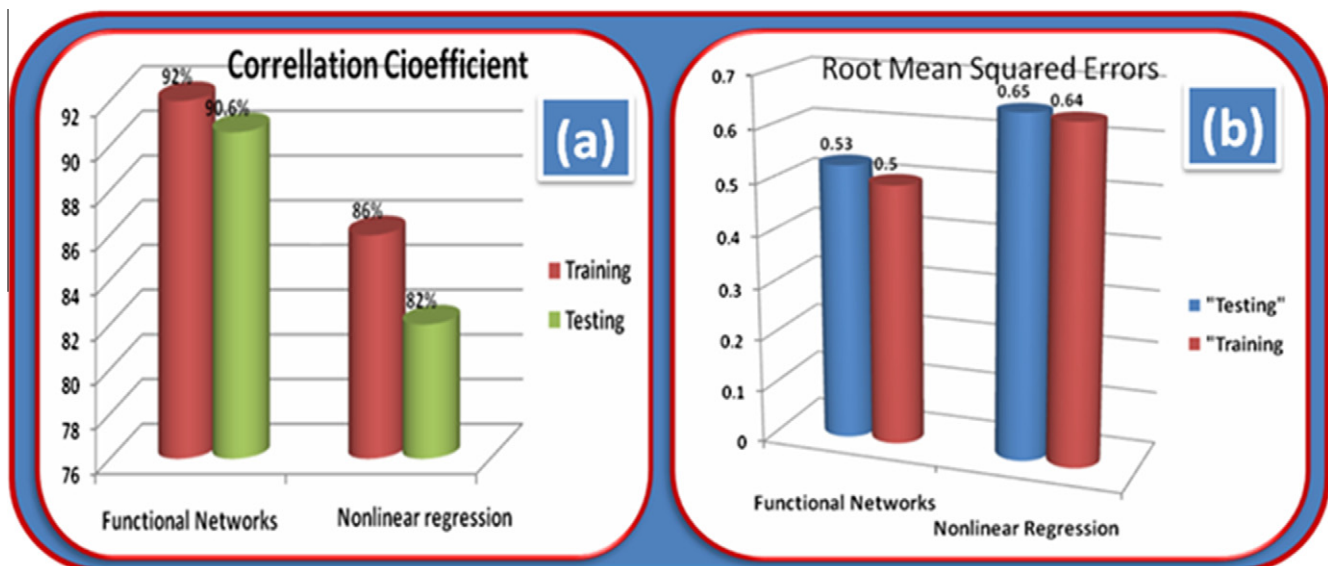


Fig. 13. (a) Correlation coefficients and (b) root mean-squared errors: Training and testing for both functional networks and statistical regression with 10-fold cross validation selection criterion.

lower error values were obtained when the input variable of functional networks predictive model were increased during the training and testing phase. This reason for functional networks to perform better with more input data is due to its basic architecture. This basic architecture can deal with multiargument functions, unlike statistical regression, which deals only with a single argument. Moreover, this kind of multiargument performance is like a clustering scheme which minimizes the computational time and provides better prediction. Graphs in Fig. 13a and b shows the R^2 and RMSE values obtained during training and testing of the data of well W-1 using functional networks and the other statistics and data mining predictive modeling schemes, respectively.

To check the performance of the functional networks intelligence system paradigm, an input dataset containing the same input variables on which the functional networks calibration predictive model has been built on, was chosen from another well. We combined the wireline logs and core data of both wells solving the problem within the differences between the depth scale in both core and wireline logs using the internal interpolations with 0.5 cm or 0.5 f nearest neighbors. Therefore, we have a unique depth scale within both core and wireline data diameter core plugs and then we have a good match between the entire core and wir-

eline data before, we build the functional networks model. In addition, we concatenated numerous of different layers of wireline logs together to have more uncertain regions with complexities associated with desired permeability prediction in carbonate reservoirs combined together. We accomplished the required assessment for pre-process (more quantitative assessment) to have a stable and unique input wireline logs and then build the data mining predictive model based on functional networks with polynomial family of degree at most 2 as a basis and apply 10-fold cross validation criterion.

Fig. 14 shows the 'kcore' and 'kpred' for Well-1 and Well-2 individually based on combined Well-1 and Well-2 using 'NRS' logs. Moreover, we investigated the developed model on new unseen wells to see how much the performance of this model is reliable and proper for real-time and daily life use.

We concluded that the predicted permeability falls and agreed with the expectations of petrophysicist, geoscientist, and reservoir engineers prior expert knowledge that they had using state of equations as it is shown in Fig. 15. In addition, results obtained through the developed data mining predictive model based on functional networks is reliable and lay within the acceptable range of permeability values. Because of the strength of the functional

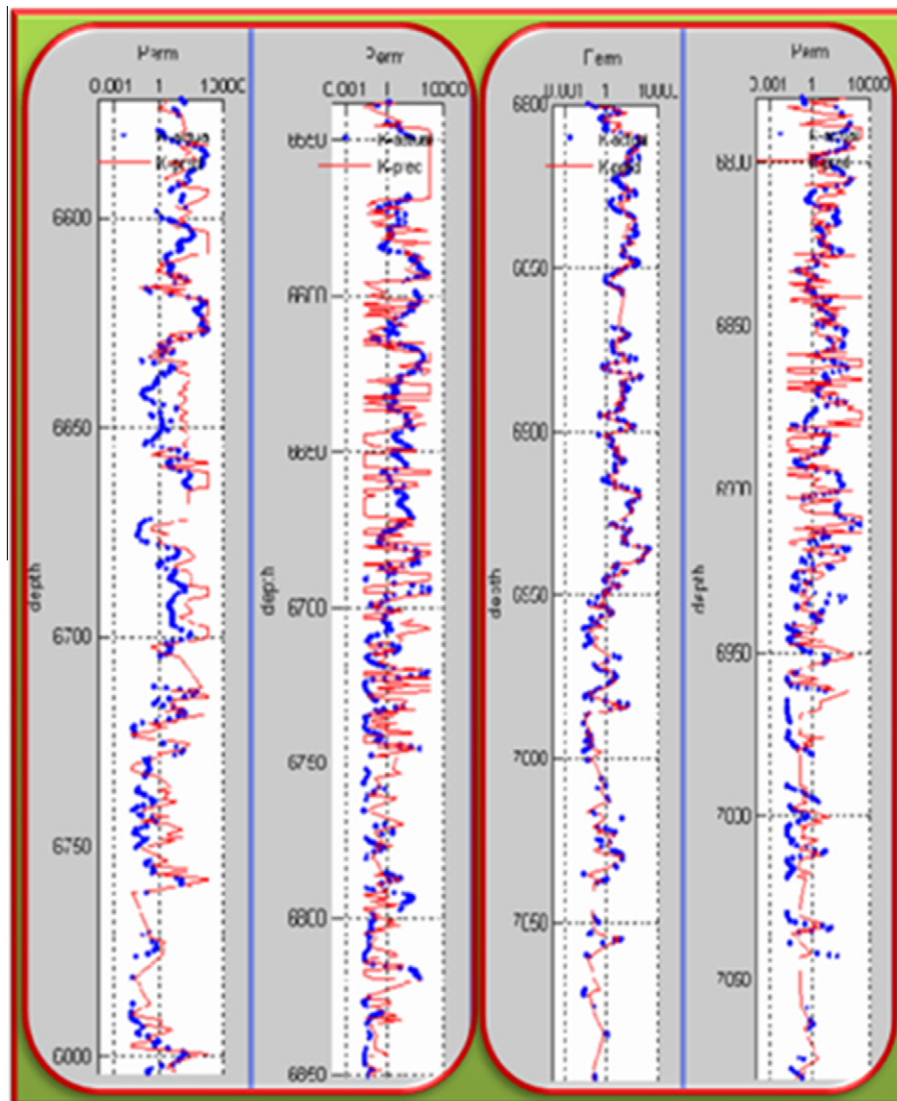


Fig. 14. 'kcore' and 'kpred' for Well-1 and Well-2 individually based on combined Well-1 and Well-2 using 'NRS' logs based on Functional networks and 10-fold selection criterion.

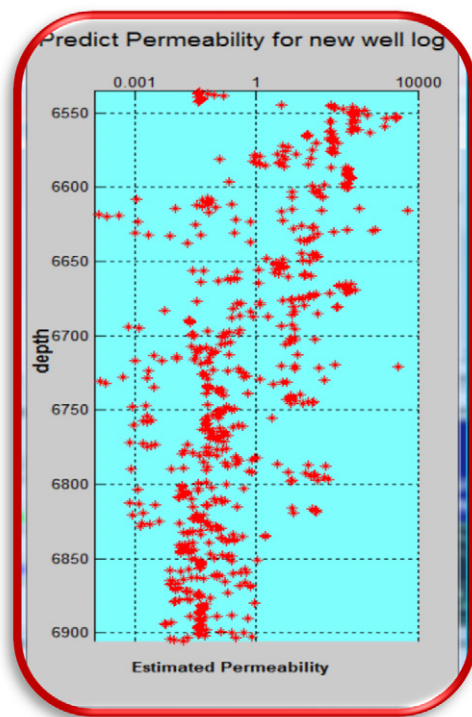


Fig. 15. Permeability prediction for a new well (Well-W2) based on the developed functional networks with polynomial of degree at most 2 and 10-cross validation using well log data.

networks intelligence system, it is expected that the proposed approach will perform equal to, if not better than, the other state of equations and statistical methods.

7. Conclusion and recommendation

We have utilized numerous real-life databases of well logs to investigate the capabilities of functional networks intelligence predictive models to forecast permeability from well logs. Results have shown that the performance of the new functional networks intelligence paradigm with polynomial basis outperforms the most common existing statistical and data mining approaches with stable and reliable output permeability, especially in uncertain regions. Based on the above study, the following conclusions can be drawn.

- The proposed computational intelligence scheme helps to overcome the limitations of the common soft-computing techniques, such as neural networks and statistical regression.
- The developed functional networks predictive model has been applied with real-time new unseen wells. The obtained results show that the performance of this model is reliable and proper for real-time and for regular use. Furthermore, because of the strength of the functional networks intelligence system, it is expected that the proposed approach will perform equal to, if not better than, the other analytical and/or statistical methods.
- The performance of functional networks with only second order degree basis polynomial families provides a reliable indicator to use it in different oil and gas industry applications such as multiphase flow regimes, history matching problems and risk analysis.

The work can be extended to include more complex non-linear basis and ensemble learning mechanism, which may lead to better accuracy and more reliable real-time results in new reservoirs.

Acknowledgements

The authors wish to acknowledge the support of Research Institute at King Fahd University of Petroleum and Minerals, Saudi Arabia with Mansoura University, Egypt for the facilities utilized to perform the present work and for their support. They are also grateful to MEDai Inc. an Elsevier Company R&D team in Orlando, Florida, USA, for the facilities utilized to perform the present work and for their support.

References

- Abdulraheem, A., El-Sebakhy, E., Ahmad, M., Vantala, A., Korvin, G., & Raharja, I. (2007). The capability of neuro-fuzzy systems in predicting permeability and porosity from well-log. In *The Proceedings the society of petroleum engineering (SPE)*. Bahrain, SPE paper 105350.
- Al-Bokhitani, S. (2007). Functional networks as a novel approach for predicting reservoir fluids PVT properties. Master thesis, Dhahran 31261, Kingdom of Saudi Arabia: Computer Science and Engineering. King Fahd University of Petroleum and Minerals. <<http://www.gradworks.umi.com/14/46/1446315.html>>.
- Amabeoku, M., Lin, O., Khalifa, C., Cole, A. A., Dahan, J., Jarlow, M. J., et al. (2005). Use of fuzzy-logic permeability models to facilitate 3D geo-cellular modeling and reservoir simulation: Impact on business. In *International petroleum technology conference*, Doha, Qatar.
- Arbogast, S., & Franklin, H. (1999). Artificial neural networks and high speed resistivity modeling software-speed reservoir characterization. *Petroleum Engineering International*, 72(5), 57–61.
- Badarinadh, V., Suryanarayana, K., Youssef, Z., Sahouh, K., & Valle, A. (2002). Log-derived permeability in a heterogeneous carbonate reservoir of middle east, Abu Dhabi, using artificial neural network. In *SPE 74345, international petroleum conference and exhibition in Mexico*, Villahermosa, Mexico.
- Balan, B., Mohaghegh, S., & Ameri, S. (1995). State-of-the-art in permeability determination from well log data: Part I – A comparative study, model development. In: *SPE eastern regional conference and exhibition*, West Virginia.
- Bhatt, A., & Helle, H. B. (2002). Committee neural networks for porosity and permeability prediction from well logs. *Geophysical Prospecting*, 50, 645–660.
- Bruce, A. G., Wong, P. M., Zhang, Y., Salisch, H. A., Fung, C. C., & Gedeon, T. D. (2000). A state-of-the-art review of neural networks for permeability prediction. *Australian Petroleum Production and Exploration Association (APPEA) Journal*.
- Bruen, M., & Yang, J. (2005). Functional networks in real-time flood forecasting—a novel application. *Advances in Water Resources*, 28(9), 899–909.
- Castillo, E. (1998). Functional networks. *Neural Process. Letters*, 7(3), 151–159.
- Castillo, E., Cobo, A., Gutiérrez, J. M., & Hadi, A. S. (1999). A general framework for functional networks. *Networks*, 35, 70–82.
- Castillo, E., Gutiérrez, J. M., Hadi, A. S., & Lacruz, B. (2001). Some applications of functional networks in statistics and engineering. *Technometrics*, 43, 10–24.
- Castillo, E., Hadi, A. S., Lacruz, B., & Pruned, E. R. (2008). Semi-parametric nonlinear regression and transformation using functional networks. *Journal of Computational Statistics and Data Analysis*, 52(5 and 10), 2129–2157.
- Chen, C. H., & Lin, Z. S. (2006). A committee machine with empirical formulas for permeability prediction. *Journal of Computers & Geosciences*, 32, 485–496.
- Cross, N., Goodall, I., Hollis, C., Burchette, T., Al-Ajmi, H., Gorman-Johnson, I., et al. (2010). Reservoir description a mid-Cretaceous siliciclastic –carbonate ramp reservoir. *Maudud Formation in the Raudhatain and Sabiriyah Fields*, North Kuwait, *GeoArabi*, 15, 17–50.
- Cuddy, S. (1997). The application of mathematics of fuzzy logic to petrophysics, In *SPWLA annual logging symposium*.
- Cuddy, S. (1998). Litho-facies and permeability prediction from electrical logs using fuzzy logic. In *Proceedings the society of petroleum engineering (SPE)*, Abu-Dhabi international petroleum exhibition and conference, (ADIPEC), Abu Dhabi, UAE, SPE paper 49470.
- Duda, R. O., Hart, P. E., & Stock, D. G. (2001). *Pattern classification* (2nd ed.). NY: John Wiley and Sons.
- El-Sebakhy, A. E. (2004). Functional networks training algorithm for statistical pattern recognition. In *IEEE symposium on computers and communications, the 9th international symposium on computers and communications* (vol. 1, pp. 92–97).
- El-Sebakhy, A. E. (2009). Data mining in forecasting PVT correlations of crude oil systems based on Type1 fuzzy logic inference systems. *Journal of Computers & Geosciences*, 35, 1817–1826.
- El-Sebakhy, A. E. (2009). Software reliability identification using functional networks: A comparative study. *Journal of Expert Systems with Applications*, 36, 4013–4020.
- El-Sebakhy, A. E. (2010). Functional networks as a novel data mining paradigm in forecasting software development efforts. *Expert Systems with Application Journal*. doi:10.1016/j.eswa.2010.08.005.
- El-Sebakhy, A. E., Asparouhov, O., Abdulraheem, A., Al-Majed, A., Wu, Donghui, Latinski, K. (2010). On utilizing functional networks computational intelligence in forecasting rock mechanical parameters for hydrocarbon reservoirs:

- Methodology and comparative studies. In *Innovative geosciences solutions: Society of petroleum engineering (SPE 739016): The 9th middle east geosciences international conference (GEO 2010)*. March 7–10, Manama, Bahrain. <<http://www.searchanddiscovery.com/abstracts/>>.
- El-Sebakhy, A. E., Hadi, S. A., & Faisal, A. K. (2007). Iterative least squares functional networks classifier. *IEEE Transactions Neural Networks*, 18(3), 844–850.
- Fahlman, S.E. (1988). *An empirical study of learning speed in back propagation networks*. Technical report, CMU-CS-88-162, Carnegie-Mellon University.
- Fangming, L., Zhongyuan, T., Aming, J., & Xiaoxia, W. (2009). Prediction of reservoir depletion degree and production GOR using logging-while-drilling data. *Journal of Petroleum Exploration and Development*, 36(5), 617–622.
- Hambalek, N., & Reinaldo, G. (2003). Fuzzy logic applied to litho-facies and permeability forecasting. In *Proceedings the society of petroleum engineering (SPE). Latin America and Caribbean petroleum engineering conference*, Spain, Trinidad, West Indies, SPE paper 81078.
- Haykin, S. (1994). *Neural networks: A comprehensive foundation*. NY: Macmillan College Publishing Company.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Processing of the National Academy of Sciences of the USA*, 79, 2554–2558.
- Hush, D. R., & Horne, B. G. (1993). Progress in supervised neural networks: What's new since Lippmann? *IEEE Signal Processing Magazine*, 10(1), 8–39.
- Jacob, R. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1, 295–307.
- Jeirani, Z., & Mohebbi, A. (2006). Estimating the initial pressure, permeability and skin factor of oil reservoirs using artificial neural networks. *Journal of Petroleum Science and Engineering*, 50, 11–20.
- Kadkhodaie-Ilkhchi, A., Rezaee, M. R., & Moallemi, S. A. (2006). A fuzzy logic approach for the estimation of permeability and rock types from conventional well log data: An example from the Kangan reservoir in Iran Offshore Gas Field, Iran. *Journal of Geophysical Engineering*, 3, 356–369.
- Kamali, M. R., & Mirshady, A. A. (2004). Total organic carbon content determined from well logs using $\Delta \log R$ and neuro fuzzy techniques. *Journal of Petroleum Science and Engineering*, 45, 141–148.
- LeCun, Y., Botou, L., Jackel, L., Drucker, H., Cortes, C., Denker, J., et al. (1995). Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Networks*, 4, 261–276.
- Lim, J. S. (2005). Reservoir properties determination using fuzzy logic and neural networks from well data in offshore Korea. *Journal of Petroleum Science and Engineering*, 49, 182–192.
- Liu, C., Nakashimam, K., Sakom, H., & Fujisawam, H. (2003). Handwritten digit recognition: Bench-marking of state-of-the-art techniques. *Pattern Recognition*, 36, 2271–2285.
- Lucia, F. J. (2008). *Carbonate reservoir characterization* (2nd ed.). New York: Springer.
- Lucia, F. J., Ruppel, S. C., Jennings, J. W., & Laubach, S. E. (2001). The importance of layering in reservoir characterization: South Wasson Clear Fork reservoir, Gaines County, Texas (abs.). In S. M. Ingram & J. J. Viveiros (Eds.), *The Permian Basin: Microns to satellites: looking for oil and gas at all scales* (pp. 131). West Texas Geological Society Publication 01–110.
- Mizutani, S., & Jang (1997). *Neuro-fuzzy and soft computing*. Prentice Hall.
- Nikraves, M., & Aminzadeh, F. (2000). Mining and fusion of petroleum data with fuzzy logic and neural network agents. *Journal of Petroleum Science and Engineering*, 29(3–4), 221–238.
- Nikraves, M., & Aminzadeh, F. (2003). Soft computing for intelligent reservoir characterization and modeling. *Development in Petroleum Sciences*, 51, 3–32.
- Riedmiller, R. H., & Braun (1993). A direct adaptive method for faster backpropagation learning: The prop algorithm. In *Processing of the IEEE international conference on neural networks* (pp. 586–591).
- Sears, S. O., & Lucia, F. J. (2006). Reservoir characterization of northern Michigan Niagaran pinnacle reefs (abs.). In *Eastern section annual meeting, American association of petroleum geologists, program with abstracts* (p. 30).
- Simon, H. (1994). *Neural networks – A comprehensive foundation*. NJ, USA: Macmillan Publishing Englewood Cliffs.
- Tamhane, D., Wong, P. M., Aminzadeh, F., & Nikraves, M. (2000). Softcomputing for intelligent reservoir characterization. In *SPE59395, Asia Pacific conference on integrated modeling for asset management, Yokohama, Japan*.
- White, H., & Stinchcombe, M. (1989). Multilayer feed-forward networks are universal approximates. *Neural Networks*, 2, 359–366.