Modeling the permeability of carbonate reservoir using type-2 fuzzy logic systems

Sunday Olusanya Olatunji a, Ali Selamat a,*, Abdulazeez Abdulraheem b
aIntelligent Software Engineering Laboratory, Faculty of Computer Science & Information Systems, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia
bCenter for Petroleum and Minerals, The Research Institute, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Box: 1105, Saudi Arabia

ARTICLE INFO

Article history:
Available online 24 November 2010

Keywords:
Permeability estimation
Type-2 fuzzy logic systems
Reservoir characterization
Support vector machines
Feedforward neural networks

ABSTRACT

In this work, the use of type-2 fuzzy logic systems as a novel approach for predicting permeability from well logs has been investigated and implemented. Type-2 fuzzy logic system is good in handling uncertainties, including uncertainties in measurements and data used to calibrate the parameters. In the formulation used, the value of a membership function corresponding to a particular permeability value is no longer a crisp value; rather, it is associated with a range of values that can be characterized by a function that reflects the level of uncertainty. In this way, the model will be able to adequately account for all forms of uncertainties associated with predicting permeability from well log data, where uncertainties are very high and the need for stable results are highly desirable. Comparative studies have been carried out to compare the performance of the proposed type-2 fuzzy logic system framework with those earlier used methods, using five different industrial reservoir data. Empirical results from simulation show that type-2 fuzzy logic approach outperformed others in general and particularly in the area of stability and ability to handle data in uncertain situations, which are common characteristics of well logs data. Another unique advantage of the newly proposed model is its ability to generate, in addition to the normal target forecast, prediction intervals as its by-products without extra computational cost. © 2010 Elsevier B.V. All rights reserved.

1. Introduction

Permeability is one of the most important reservoir properties, and its prediction has been one of the fundamental challenges to petroleum engineers and researchers [1]. Accurate knowledge of permeability property is required to determine the amount of oil or gas present in reservoirs, the amount that can be recovered, the flow rate of oil or gas, the forecast of future production, and the design of production facilities. The overall reservoir management and development requires accurate knowledge of permeability [1,2]. Permeability or flow capacity is the ability of porous rock to transmit fluid [3]. The fact that a rock is very porous does not necessarily translate to being very permeable. Permeability is the ease with which fluid is transmitted through a rock’s pore space. It is a measure of how interconnected the individual pore spaces are in a rock or sediment and it is a key parameter associated with the characterization of any hydrocarbon reservoir [4]. In fact, many petroleum engineering problems cannot be solved accurately without having an accurate permeability value [4]. During the past few decades, numerous efforts have been made to forecast permeability using well log data and available core data through laboratory measurements. In some oil fields, representative values for porosity and permeability obtained from different locations are generally assumed that a linear or non-linear relationship exists between permeability and other properties of the rock, but unfortunately the method has failed to solve permeability prediction problems [2,7].

The recent success recorded in the application of artificial neural networks (ANN) to solving various engineering problems has drawn researchers’ attentions to its potential viability in the petroleum industry. Thus, in attempt to resolve problems associated with the parametric approach, the standard ANNs have been used to provide better prediction models [8,9]. These works yielded significant prediction improvement in the oil and gas industries. However, the technique still suffers from several drawbacks. These shortcomings include the trial-and-error approach of ANN and the need to guess its architectural parameters in advance. Though, acceptable results may be obtained in some situations, it is obvious that potentially superior models may be overlooked unintentionally. Most importantly, ANN suffers from instability in its predictions and it is unable to model uncertainties that characterize well logs in particular.
Researchers have done their best to address and overcome these problems of ANN. As a result, several variants of ANN and other methods like support vector machines (SVM) and functional networks (FN) have been proposed and used [7,10], yet each has its limitations that still call for further research of this nature, particularly their inability to handle uncertainties and the need to ensure stability and consistency in permeability predictions.

It is an established fact that geosciences disciplines are not clear-cut and, most of the time, are associated with uncertainties [11], hence the need for fuzzy logic based systems, particularly the newly introduced type-2 fuzzy logic systems (type-2 FLS) that is able to adequately account for all forms of uncertainties [12]. For instance, prediction of core parameters from well log responses is difficult and is usually associated with uncertainties. Earlier methods try to minimize and ignore these uncertainties [11], while type-2 fuzzy logic derives useful information from the uncertainties and uses it as a good selection of parameters for increasing the accuracy of the predictions while also ensuring stability and consistency.

Recently, type-2 FLS have been proposed as a novel framework for both classification and prediction in order to handle all forms of uncertainties [12,13]. It is able to handle uncertainties that include those in measurements and data used to calibrate the parameters. It has been used in several fields and the results have been promising and very encouraging [14–16]. Therefore, there is a possibility that the type-2 FLS can handle uncertainty in reservoir well log data [11] as the type-2 fuzzy logic has been specifically invented to deal with all forms of uncertainties [12] that are inherent in our day to day natural encounters and mode of reasoning. Therefore in this paper, we propose a type-2 fuzzy logic based prediction model for estimating permeability from well log data.

The main objectives of this study are (i) to investigate the feasibility of type-2 FLS in forecasting permeability; (ii) to develop a new intelligence framework, based on type-2 FLS, for predicting permeability from well logs using real industrial well log data; (iii) to investigate how various earlier commonly used standard artificial neural network (ANN), support vector machines (SVM), and type-1 fuzzy logic system compared with the proposed method in predicting permeability of carbonate reservoirs from well logs; and (iv) to explore how type-2 FLS framework can generate permeability prediction intervals, effortlessly, without extra computational cost.

The rest of this paper is organized as follows: Section 2 presents a review of related researches and Section 3 presents the proposed model. Section 4 contains the empirical study, implementation process, comparative studies, results and discussions. The conclusion and future work recommendations are provided in Section 5.

2. Related research

Permeability is one of the most important reservoir properties, and its prediction has been one of the fundamental challenges to petroleum engineers and researchers [1]. In general, permeability determination has been carried out using either empirical, statistical, or the recently introduced “virtual measurement” methods. Respectively, the researchers have utilized the usage of empirically determined models, multiple variable regression, and predictive computational intelligence paradigms in predicting the oil well logs data sets.

2.1. Empirical and statistical methods

These methods respectively make use of empirically determined models and multiple variable regression. According to [2], the first equation relating measurable rock properties with permeability was proposed in 1927 by Kozeny and modified by Carman. Their formulations are valid only for packs of uniformly sized spheres. Another major drawback is that surface area can be determined only by core analysis, and only with special equipment. In 1941, Archie, in his classical paper [17], set the basis for quantitative log interpretation, though he did not provide a permeability formula. Analyzing the laboratory-determined resistivity of a large number of brine-saturated cores from various sand formations, Archie [17] has introduced the concept of “formation resistivity factor”. In 1949, Tixier [18] established a method for determining permeability from resistivity gradients using empirical relationships between resistivity and water saturation, water saturation and capillary pressure, and capillary pressure and permeability. The resistivity gradient is determined from a deep investigation tool, lateral or focused logs, and corrected for borehole effects. This method assumes that saturation exponent, \( n \), is equal to 2.0 at any level of water saturation. The model is physically limited in scope by the relative paucity of logs exhibiting valid oil water contacts and the necessity for estimating the hydrocarbon density as it exists in the reservoir. Also, the calculated permeability is an average for the zone corresponding to the resistivity gradient.

Following the work of Wyllie and Rose [19], Tixier [18] developed a simpler model that is used more often than his earlier equations. In 1950, Wyllie and Rose [19], in their thorough analysis of the theoretical basis of quantitative log interpretation, expanded the empirical relationship proposed by Tixier [18].

According to [20], Sheffield has proposed another correlation for permeability based on Kozeny’s equation as reported in [2] and following the establishment of a correlation coefficient for some well-known water-wet sands. Pirson [20] proposed another permeability formula in 1963. His empirical relationship was determined by multiple correlations from relatively few data. The formula is not valid for high gravity crudes (API > 40°, where API is referring to American Petroleum Industry) and for depths greater than 6500 ft. Timur [21] proposed a generalized equation in 1968 based on the work of Kozeny as reported in [2] and Wyllie and Rose [19]. His equation can be evaluated in terms of the statistically determined parameters. He applied a reduced major axis (RMA) method of analysis to data obtained by laboratory measurements conducted on 155 sandstone samples from three different oil fields from North America. Based on both the highest correlation coefficient and on the lowest standard deviation, Timur [21] developed his own formula for permeability based on selection from five alternative relationships. This model is applicable where condition of residual water saturation exists. Timur also assumed that a value of 1.5 for the cementation factor holds in all cases.

Coates and Dumanoir [5] proposed an improved empirical permeability technique. With the support of core and log studies, they adopted a common exponent for both the saturation exponent and cementation exponent. Coates and Dumanoir [5] also presented a methodology for testing if the formation is at irreducible water saturation. However, they note that if the reservoir is heterogeneous, it may fail that test and still be at irreducible water saturation. It must be noted at this point that the method of Coates and Dumanoir is the first that satisfies the condition of zero permeability at zero porosity [2].

Wendt et al. [6] established a general procedure for permeability prediction by multiple variable regression. They also pointed out the shortcomings of using this technique. When the regression method is used for prediction, the distribution of predicted values is narrower than that of the original data set [2]. The ability of a regression model to predict the permeability extremes is enhanced through a weighting scheme of the high and low values. However, the predictor can become unstable and also statistically biased [2].
Despite all these efforts, permeability estimation has not been effectively handled with the desired accuracy due to various shortcomings of these empirical and statistical methods, hence the need to explore virtual measurement techniques for better results.

2.2. Virtual measurement technique

A virtual measurement technique is referred to as the computational intelligence or neural network based methods. There have been several variants of these methods, the most popular of which is the classical artificial neural network paradigms. Attempts have been made to utilize artificial neural networks (ANNs) for identification of the relationship which may exist between well logs and core permeability. However, despite the wide range of applications and flexibility of ANNs, there is still no general framework or procedure through which the appropriate network for a specific task can be designed as stated by Mohsen et al. [22].

One of the most popular works on permeability has been done by Bruce et al. [23] where they presented a state-of-the-art review of the use of neural networks for predicting permeability from well logs. In this application, neural network was used as a non-linear regression tool to develop transformation between well logs and core permeability. Such a transformation can be used for estimating permeability in un-cored intervals and wells. In this work, the permeability profile was predicted by a Bayesian neural network. The network was using a training set with four well logs and core permeability. The network also provided a measure of confidence (the standard deviation of a Gaussian function), whereby the higher the standard deviation (σ), the lower the prediction reliability. This is very useful for understanding the risk of data extrapolation. The same tool can be applied to estimate porosity and fluid saturations. Another important application is the clustering of well logs for the recognition of lithofacies by Rogers et al. [24]. This provides useful information for improved petrophysical estimates and well correlation.

In [25] the authors used ANN and fuzzy logic to characterize naturally fractured reservoirs. Using these tools, they produced 2-D fracture intensity and fracture network maps in a large block of oil fields. The results showed that the proposed approach is a practical methodology to map the fracture network. The use of functional network was also reported in [7] for predicting permeability from well logs, where functional equations were made use of to get better predictions. Recently, application of fuzzy logic in petroleum engineering field has received considerable attention and has been successfully applied to address problems on various oil and gas reservoirs such as identification of lithofacies and prediction of permeability using wire line logs [26–28]. For predicting these properties generally Gaussian membership and fuzzy clustering algorithm are applied and better estimates have been reported compared to that of conventional techniques. In [26] the authors made use of fuzzy logic for the estimation of permeability from wireline logs in a Middle Eastern carbonate reservoir, and they reported better result compared to the ANN and regression methods.

Finol et al. [29] have proposed the use of fuzzy logic for the prediction of petro-physical rock parameters. He made use of the rule-based fuzzy model corresponding to the Takagi–Sugeno–Kang method of fuzzy reasoning proposed by Sugeno and Kang [43]. In the approach used, a fuzzy clustering algorithm is combined with the least-square approximation method to identify the structure and parameters of the fuzzy model from sets of numerical data. To verify the effectiveness of the proposed fuzzy modeling method, two case study examples have been developed using core and electrical log data from three oil wells in Ceuta Field, Lake Maracaibo Basin, Venezuela. The numerical results of the fuzzy modeling method have been compared with the results of a conventional linear regression model. It is reported that the fuzzy modeling approach is not only more accurate than the conventional regression approach but also provides some qualitative information about the underlying complexities of the porous system.

The use of hybrid approach which is a combination of neural network and fuzzy logic has also been reported in [30]. This hybrid 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.

Yuantu et al. [31] have introduced a new neural-fuzzy technique combined with genetic algorithms in the prediction of permeability in petroleum reservoirs. The methodology involved the use of neural networks to generate membership functions and to approximate permeability automatically from digitized data of well logs obtained from oil wells. The trained networks were used as fuzzy rules and hyper-surface membership functions. The results of these rules were then interpolated on the basis of membership grades and the parameters in the defuzzification operators which were optimized by genetic algorithms. The results showed that the integrated neural-fuzzy-genetic-algorithm (INFUGA) gave the smallest error on the unseen data when compared to similar algorithms.

Jong-Se [32] has suggested an intelligent technique using fuzzy logic and neural networks to determine reservoir properties from well logs using fuzzy curve analysis based on fuzzy logic for selecting the best related well logs with core porosity and permeability data. Neural network was then used as a nonlinear regression method to develop transformation between the selected well logs and core measurements. The results showed that the technique estimated the reservoir properties more accurately and reliably than conventional computing methods.

We have noted that all the reported cases of the use of fuzzy logic in modeling permeability are restricted to the classical fuzzy logic (also known as type-1 fuzzy logic). However, type-1 fuzzy logic systems have recently been found inadequate for handling all forms of uncertainties [12,13,33,34]. In response to this, type-2 fuzzy logic systems have been introduced as better computational intelligence approach for both prediction and classification to handle all forms of uncertainties [33]. The unique feature and advantage of type-2 fuzzy logic systems, coupled with its ability to generate prediction intervals without extra computational cost, formed part of what motivated this proposed work for permeability modeling.

3. Type-2 fuzzy logic systems framework

In this work, type-2 fuzzy logic systems (type-2 FLS) framework is investigated, implemented and utilized for predicting permeability from well log data based on five distinct real-industrial well log data obtained from middle-eastern petroleum reservoirs. As a unique characteristic of adaptive fuzzy systems, the goal is to completely specify the fuzzy logic systems using the training data. Type-2 adaptive fuzzy inference systems are adaptive networks that learn the membership functions and fuzzy rules, from data, in a fuzzy system based on type-2 fuzzy sets [12,35]. Type-2 fuzzy sets are the types of fuzzy sets whose grades of membership are also fuzzy, and they are intuitively interesting because grades of membership can never be obtained precisely in real-life situations [36]. Type-2 fuzzy sets can be used in situations where there is uncertainty about the membership grades themselves or other forms of uncertainty. For example, an uncertainty in the shape of the membership function or in some of its parameters, uncertainty about the consequent that is used in a rule, uncertainty about the measurements that activate the FLS, and uncertainty about the
data that are used to tune the parameters of a FLS. Consider the transition from ordinary sets to fuzzy sets; when we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is very fuzzy that we have difficulty in determining the membership grade as a crisp number in [0, 1], we use fuzzy sets of type-2. Thus, in general, “a fuzzy set is of type $n$, $n = 2, 3, \ldots$ if its membership function ranges over fuzzy sets of type $n - 1$” [37].

Generally type-2 fuzzy logic systems (type-2 FLS) are made up of five interconnected components that include fuzzifier, rules, inference engine, type-reducer and defuzzifier. The fuzzifier takes the well log input parameters values as inputs. The output from the fuzzifier is the fuzzified data, which will serve as input to the inference engine. The outputs from the inference engine are type-2 fuzzy sets which can be reduced to type-1 fuzzy set through the type reducer. The type reduced fuzzy sets produced in this model are interval sets that give the predicted permeability as a possible range of values. The defuzzifier then calculates the average of the generated interval sets to produce the predicted crisp permeability. The core components of the type-2 FLS framework proposed for permeability estimation are shown in Fig. 1.

### 3.1. Learning model in type-2 FLS

In this section, we present briefly the process for the automatic learning of the proposed system from the available input data together with other necessary details.

#### 3.1.1. Inferencing in the proposed type-2 FLS

Fuzzy inference engine combines the fired fuzzy rules and maps inputs into type-2 output fuzzy sets. Generally a type-2 FLS is a fuzzy logic system in which at least one of the fuzzy sets used in the antecedent and/or consequent parts and each rule inference output is a type-2 fuzzy set. Consider a type-2 Mamdani FLS [12] having $n$ inputs $x_1 \in X_1, \ldots, x_n \in X_n$ and one output $y \in Y$. The rule base contains $l$ type-2 fuzzy rules expressed in the following form:

$$R^i : IF X_1 = F^i_1 and X_2 = F^i_2 and \ldots and x_n = F^i_n THEN y = G^i$$

where $i = 1, 2, \ldots, l$, $F^i_j$ and $G^i$ are type-2 fuzzy set.

This rule represents a type-2 fuzzy relation between the input space $X = X_1 \times X_2 \times \cdots \times X_n$ and the output space $Y$ of the system. We denote the membership function of this type-2 relation as:

$$\mu_{F^i_1 \times \cdots \times F^i_n} \rightarrow G^i(x, y)$$

where $F^i_1 \times \cdots \times F^i_n$ denotes the Cartesian product of $F^i_1, F^i_2, \ldots, F^i_n$ and $x = [x_1, x_2, \ldots, x_n]$.

The antecedents in the fuzzy rules are connected using the meet operation, the firing strength of the input fuzzy sets is combined with output fuzzy sets using the extended sup-star composition, and the multiple rules are combined using the join operation [12]. However, the computing load involved in deriving the system output from a general type-2 FLS model is high in practice, and the general practice is to use the interval type-2 FLS in which the fuzzy sets $F^i_1$ and $G^i$ are interval fuzzy sets through which the computing of type-2 FLS can be greatly simplified. The membership grades of interval fuzzy sets can be fully characterized by their lower and upper membership grades of the footprint of uncertainty (FOU) separately [12].

Without the loss of generality, let $\mu_{F^i_1}(x) = [\mu_{F^i_{1L}}(x), \mu_{F^i_{1U}}(x)]$ and $\mu_{G^i}(y) = [\mu_{G^i_{1L}}(y), \mu_{G^i_{1U}}(y)]$ for each sample $(x, y)$. The firing strength of interval type-2 FLS $\mu_{F^i}(x) = \cap_{j=1}^n \mu_{F^j}(x)$ is an interval [12] i.e.

$$\mu_{F^i}(x) = [\mu^i_{F^i_{1L}}, \mu^i_{F^i_{1U}}](X)$$

in the proposed interval type-2 FLS, the meet operation under product t-norm is used, so that the firing strength is an interval type-1 set [12] as shown below:

$$f^i(X) = \left[\mu^i_{F^i_{1L}}(X), \mu^i_{F^i_{1U}}(X)\right] = \left[\bar{f}^i, f^i\right]$$

where $\bar{f}^i(X)$ and $f^i(X)$ can be re-written as follows with * representing the t-norm product operation:

$$\bar{f}^i(X) = \mu^i_{F^i_{1L}}(x_1)^* \ldots \mu^i_{F^i_{1L}}(x_n)$$

$$f^i(X) = \mu_{G^i_{1L}}(x_1)^* \ldots \mu_{G^i_{1U}}(x_n)$$

#### 3.1.2. Type reduction

The results from the inference engine are type-2 fuzzy sets. There is need to reduce the type-2 fuzzy sets to type-1 fuzzy sets in order to give room for defuzzification so that the final crisp outputs can be generated. Center-of-sets (COS) type-reducer algorithm developed by Mendel and Karnik [12,38] has been used in this study because it provide reasonable computational complexity compared to others like the expensive centroid type reducer. COS type reducer use two steps in reducing the type-2 fuzzy sets as follows: (i) calculating the centroids of type-2 fuzzy rule consequences and (ii) calculating the reduced fuzzy sets. These stages are described in the following two subsections.

#### 3.1.2.1. Computing the centroids of type-2 fuzzy rule consequences

Suppose that the output of an interval type-2 FLS is represented by type-2 fuzzy sets $G^t$, where $t = 1, \ldots, T$, $T$ is the number of output fuzzy sets. In this first stage, the centroids of all the $T$ output fuzzy sets are calculated and they will be used in calculating the reduced sets in the next stage. The centroid of the $i$th output fuzzy set $y^i$ is a type-1 interval set which can be expressed in the following equation [12,38]:

$$y^i = [y^i_L, y^i_U] = \frac{1}{\sum_{i=1}^T \sum_{j=1}^T y^j_i y^j_j} \sum_{j=1}^T y^j_j$$

where $y^i_L$ and $y^i_U$ are the leftmost and rightmost point of $y^i$ respectively.

---

**Fig. 1.** Schematic diagram of type-2 FLS based framework for predicting the permeability of carbonate reservoirs from well logs.
3.1.2.2. Computing the reduced fuzzy sets. To calculate the type-reduced set, it is sufficient to compute its upper and lower bounds of the reduced set \( y_l \) and \( y_u \), which can be expressed as follows:

\[
y_l = \frac{\sum_{i=1}^{M} f_i^l y_i^l}{\sum_{i=1}^{M} f_i^l}, \quad y_u = \frac{\sum_{i=1}^{M} f_i^u y_i^u}{\sum_{i=1}^{M} f_i^u}
\]

where \( f_i^l \) and \( y_i^l \) are the firing strength and the centroid of the output fuzzy set of \( i \) th rule \((i = 1, \ldots, M)\) associated with \( y_l \) respectively. Similarly, \( f_i^u \) and \( y_i^u \) are the firing strength and the centroid of the output fuzzy set of \( i \) th rule \((i = 1, \ldots, M)\) associated with \( y_u \) respectively.

3.1.3. Defuzzification

We defuzzify the type-reduced set to get a crisp output from the type-2 FLS. The final output of type-2 FLS is thus set to the average of \( y_l \) and \( y_u \) as shown below:

\[
y(x) = \frac{y_l + y_u}{2},
\]

where \( y(x) \) is the final crisp output.

3.1.4. The steepest descent approach for training FLS

The purpose of the training algorithm is to minimize the error function for \( E \) training epochs.

\[
E(t) = \frac{(f(x^{(i)}) - y^{(i)})^2}{y^{(i)}}, \quad t = 1, \ldots, N
\]

Consider a FLS with Gaussian membership functions, center of sums type-reducer, average defuzzification, max-product composition, and product implication; it could be expressed by the equation:

\[
y(x^{(i)}) = \sum_{i=1}^{M} f_i^l \frac{1}{\sqrt{2\pi} \sigma_i^l} \exp\left[-\frac{(x^{(i)} - \mu_i^l)^2}{2\sigma_i^l}ight]
\]

\[
\sum_{i=1}^{M} f_i^u \frac{1}{\sqrt{2\pi} \sigma_i^u} \exp\left[-\frac{(x^{(i)} - \mu_i^u)^2}{2\sigma_i^u}\right]
\]

\[
i = 1, \ldots, N
\]

where \( M \) is number of rules, \( p \) is number of antecedents and \( N \) is number of data points, \( m_i^l \) and \( \sigma_i^l \) are the mean and standard deviation for the membership function respectively.

Given an input–output training pair \((x^{(i)}, y^{(i)})\) also known as data point, we wish to design a fuzzy logic system (FLS) so that the error function is minimized. The steepest descent approach [12] can be applied to obtain the following recursions to update all the design parameters of this FLS in order to minimize the error function.

\[
m_{i,l}(t+1) = m_{i,l}(t) - \alpha_m [f_i(x(i)) - y(i)] y_i^l \frac{x_i^{(i)} - m_{i,l}(t)}{\sigma_{i,l}^l(t)} \phi_i(x^{(i)})
\]

\[
y_i^l(i+1) = y_i^l(i) - \alpha_{y_i} [f_i(x(i)) - y(i)] \phi_i(x^{(i)})
\]

\[
\sigma_{i,l}^l(i+1) = \sigma_{i,l}^l(i) - \alpha_{\sigma_{i,l}} [f_i(x(i)) - y(i)] [y_i^l(i) - f_i(x^{(i)})] \frac{x_i^{(i)} - m_{i,l}(i)}{\sigma_{i,l}^l(i)} \phi_i(x^{(i)})
\]

Now, the back propagation algorithm can be applied as follows:

**Algorithm 1** (Back propagation algorithm for FLS).

1. Initialize the parameters of all the membership functions for all the rules, \( m_i(0), \sigma_i(0) \) and \( \theta_i(0) \)
2. Set an end criterion to achieve convergence.

3. Repeat
   a. For all data points \((x^{(i)}, y^{(i)}) i = 1, \ldots, N\)
      a) Propagate the next data point through the FLS.
      b) Compute error.
      c) Update the parameters of the membership functions using Eqs. (10)–(12).
   b. End for (*end for each input–output pair*)
   c. Compute the root mean square relative error (RMSRE) as in Eq. (13).
   d. Test the end criterion. If satisfied break.

Until (*end for each epoch*)

\[
RMSRE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left[ f_i(x^{(i)}) - y^{(i)} \right]^2}
\]

3.2. Implementation of the proposed type-2 FLS

In implementing the proposed type-2 fuzzy logic system (FLS), the following procedures were used. (i) Initializing the type-2 FLS, (ii) training the type-2 FLS, and (iii) testing the type-2 FLS.

3.2.1. Initializing the type-2 FLS

To carry out initialization, the components of type-2 FLS must be defined from the perspective of permeability modeling and then using the available well log datasets to initialize the framework. In a fuzzy system of this kind there must be the antecedents and the consequents: (1) internal attributes are the antecedents, which in this case are the well log variables; while (2) external attribute is the consequent which is the permeability value to be predicted in this case.

To initialize the framework, we made use of the training data extracted from the available database using the stratifying sampling approach. In the proposed model, the antecedent and consequent membership functions are made to be type-2 Gaussian with uncertain mean \( m \) and the input membership functions are made to be type-2 Gaussian with uncertain standard deviation \( \sigma \) as shown below.

\[
\mu_A(x) = \exp \left[-\frac{1}{2} \frac{(x-m)^2}{\sigma} \right] m \in [m_1, m_2]
\]

For each value of \( m \), there will be a different membership curve as depicted in Fig. 2.

\[
\sigma_A(x) = \exp \left[-\frac{1}{2} \frac{(x-m)^2}{\sigma} \right] \sigma \in [\sigma_1, \sigma_2]
\]

Also, for each value of \( \sigma \), there will be a different membership curve as shown in Fig. 3.
The uniform shadings for the footprint of uncertainties (FOU), in the two figures above, stand for interval sets of the secondary membership function and represent the entire interval type-2 fuzzy set \( \tilde{m}(x, u) \).

Sample rule for the proposed type-2 FLS looks like:

\[
R_i: \text{IF } x_1 \text{ is } F_{i1} \text{ and } x_2 \text{ is } F_{i2} \ldots \text{and } x_p \text{ is } F_{ip} \text{ THEN } y_i \text{ is } G^i
\]

From this rule, for the consequent part, \( R^i \) represent the \( i \)th type-2 fuzzy rule for the \( i \)th sample, \( F_i \) is a fuzzy set whose membership function is centered at the 1st attribute of the \( i \)th sample. For the consequent part, \( G^i \) is a fuzzy set whose membership function is centered at target output \( y_i \) of \( i \)th sample. For a further detail explanation of various ways to initialize and train type-2 FLS, see [12].

3.2.2. Training of the framework using adaptive type-2 fuzzy learning process

Having initialized type-2 FLS, training is then carried out using certain percentage of the available well log dataset. The training procedure used in this work follow strictly type-2 fuzzy logic standard, details of which can be found in [12,34,38].

The adaptive network for the proposed type-2 fuzzy inference system is shown in Fig. 4. The parameters to be learnt in the adaptive network shown in Fig. 4 include: the first layer parameters for the membership grades of the antecedent type-2 fuzzy sets, and the parameters for the consequent type-2 fuzzy sets in the 3rd layer. As it is the case with the classical adaptive networks, the final numeric output from the depicted network is compared with the target output and the error then back propagated to adjust the nodes’ parameters.

For a brief illustration, the method for two inputs is considered but this is extendable. GR and RT are vectors representing well log inputs (which represent gamma rays and resistivity respectively) that might have different representations for varied type-2 fuzzy sets while \( Perm \) denote the final permeability output from the system.

The grades in type-2 fuzzy sets are fuzzy numbers in \([0, 1]\), rather than a membership grade being crisp in \([0, 1]\), where the size of the set indicates the uncertainty attached to the number. Now consider the following four fuzzy rules from a two input, one output model:

Rule 1 : IF GR is \( A_1 \) and RT is \( B_1 \) THEN \( Z \) is \( C_1 \)
Rule 2 : IF GR is \( A_1 \) and RT is \( B_2 \) THEN \( Z \) is \( C_2 \)
Rule 3 : IF GR is \( A_2 \) and RT is \( B_1 \) THEN \( Z \) is \( C_3 \)
Rule 4 : IF GR is \( A_2 \) and RT is \( B_2 \) THEN \( Z \) is \( C_4 \)

where \( A_1, A_2, B_1, B_2, C_1, C_2, C_3, \) and \( C_4 \), are type-2 fuzzy sets.

The type-2 fuzzy system encoded in the rules above is depicted in the network of Fig. 4. There are two node types in the network representations: square and circle nodes. The square node stands for an adaptive node that contains parameters that can be modified while a circle stands for a fixed node that just performs a function.

The procedure consists of a forward pass and a backward pass with a learning algorithm that makes use of a combination of least square error method and gradient descent to determine the values of the parameters that are in the adaptive nodes. The modified forward pass is presented in the following section, interested readers can see [12,35,39] for further details.
3.2.3. The adaptive type-2 fuzzy logic learning procedures for the proposed permeability model

1st layer: All nodes here are adaptive nodes. Each of the nodes here contains three linguistic grades of membership of the type-2 fuzzy set that will match the well log inputs supplied for training. In this work, three linguistic grades that include low, medium and high are considered.

\[ O_{i,j}(x) = \frac{1}{1 + \exp(-a_{ij}(x - c_{i,j}))} \quad \text{for} \quad a_{ij} < 0, \quad 0 \leq x \leq 1 \]  

\[ O_{i,h}(x) = \frac{1}{1 + \exp(-a_{i,h}(x - c_{i,h}))} \quad \text{for} \quad a_{i,h} > 0, \quad 0 \leq x \leq 1 \]  

\[ O_{i,m} = \frac{1}{1 + |x - c_{i,m}/a_{i,m}|^{2k_{w,ir}}}, \quad 0 \leq x \leq 1 \]  

2nd layer: Here the nodes are non-adaptive in nature (i.e. fixed), which means that there is no parameters’ learning that will take place in these nodes. The AND operations in the rules are carried out in this layer. Each node has two linguistic grades as inputs coming from the output of the 1st layer. Assuming that the grade from \( A_i \) is given as \( f_i \) and that of \( B_i \) is given as \( g_i \), then applying Zadeh’s definition in [37], we have:

\[ O_{Z,j} = \sum_{j} (f_i(u_j) \land g_i(w_k))/(u_j \land w_k), \quad i = 1, 2 \]

The resultant output from this 2nd layer is a type-1 fuzzy set which is also a membership grade in a type-2 fuzzy set.

3rd layer: In this layer, the nodes are adaptive, which means parameters’ learning take place here. The nodes here are used for learning the type-2 fuzzy set denoted as \( C_i \). Just like the case in the 1st layer, each node here has three linguistic grades low, medium and high, which are represented thus:

\[ O_{C_{i,j}}(x) = \frac{1}{1 + \exp(-a_{i,j}(x - c_{i,j}))} \quad \text{for} \quad a_{i,j} < 0, \quad 0 \leq x \leq 1 \]  

\[ O_{C_{i,h}}(x) = \frac{1}{1 + \exp(-a_{i,h}(x - c_{i,h}))} \quad \text{for} \quad a_{i,h} > 0, \quad 0 \leq x \leq 1 \]  

\[ O_{C_{i,m}} = \frac{1}{1 + |x - c_{i,m}/a_{i,m}|^{2k_{w,ir}}}, \quad 0 \leq x \leq 1 \]

Using Zadeh’s extension principle [37,40], for \( O_{Z,j} \Rightarrow C_i \) then the membership grade \( O_{C_{3,j}} \) is given as:

\[ O_{C_{3,j}} = \sum_{i,j} a_{ij} \land \beta_j (1 - u_j) \land (v_i \land w_j) \]  

where \( O_{Z,j} \) and \( O_{C_{3,j}}(\varepsilon = 1, m or h) \) have been discretised to

\[ O_{Z,j} = \sum_{i} a_i/v_i \]  

\[ O_{C_{3,j}} = \sum_{j} \beta_j /w_j \]  

The final output in this 3rd layer is a type-2 fuzzy set.

4th layer: There is only a single fixed node in this layer. This fixed node does the composition of the entire outputs of the rules. This involves carrying out “join” operation (union) on the membership grades of the \( C_i \). Therefore,

\[ O_4 = \cup_j O_{C_{3,j}} \]  

where for two type-2 fuzzy sets \( \mu_A(x) = \sum f_i(u_i)/u_i \) and \( \mu_B(x) = \sum g_i(v_j)/v_j \), the join is defined by

\[ \mu_{A \land B}(x) = \mu_A(x) \cup \mu_B(x) = \sum_{i,j} f_i(u_i) \lor g_i(v_j) \]  

where the functions \( f \) and \( g \) are membership functions of fuzzy grades, \( \cup \) denotes join and \( \lor \) denotes meet.

The final resultant output from this 4th layer is also a type-2 fuzzy set.

5th layer: The only node in this layer is fixed, and its function is to carry out the type reduction and defuzzification operations. See [12,41] for details on different types of reduction process. This is where the crisp permeability values will be produce after the type reduction and defuzzification processes. The generated crisp permeability value can then be compared with the target value to determine the error that will be ‘back propagated’ through the network for parameters modification.

3.2.4. Testing the type-2 FLS

In any developed model, testing or validation is a crucial task required to demonstrate whether the new framework actually perform to expectations or not. In the present work, the validation of the type-2 FLS framework is carried out using part of the acquired well log databases, which was divided using the stratifying sampling approach. The validation facilitates easy determination and comparison of performance of the new model viz-a-viz other existing methods.

3.3. Generated type-2 FLS

Given below in Fig. 5a–g are samples of the generated type-2 fuzzy logic systems in this paper, specifically the membership functions for each of the independent variables and the dependent variable. The membership function shown represents the six predictor well logs variables that include sonic travel time (DT), Micro spherically Focused Log (MSFL), Neutron porosity (NPHI), total porosity (PHIT), bulk density (RHOB), and water saturation (SWT), and then the permeability (PERM), which is the predicted variable.

3.4. Support vector machines (SVM)

Support vector machines combine generalization control with a technique to address the curse of dimensionality. The formulation results in a global quadratic optimisation problem with box constraints, which is readily solved by interior point methods. The kernel mapping provides a unifying framework for most of the commonly employed model architectures, enabling comparisons to be performed. SVM which was primarily developed for classification problems has also been recently extended to regression problems. In classification problems, generalization control is obtained by maximising the margin, which corresponds to minimizing the weight vector in a canonical framework. The solution is obtained as a set of support vectors that can be sparse. These lie on the boundary and as such summarize the information required to separate the data. Fig. 6 shows how a margin is created between two sets of data in a classification problem.

3.4.1. Support vector regression (SVR)

Unlike classification problems where the outputs are either 1 and 0 or 1 and −1, the outputs in the regression problems are real numbers. This makes it a bit difficult to model this type of information which has infinite possibilities. In the case of regression, a margin of tolerance \( \varepsilon \) is set in approximation to the SVM which would have already being inferred from the problem.
Fig. 5. (a) Membership functions for the sonic travel time (DT); (b) membership functions for the Micro spherically Focused Log (MSFL); (c) membership functions for the Neutron porosity (NPHI); (d) membership functions for the total porosity (PHIT); (e) membership functions for bulk density (RHOB); (f) membership functions for the water saturation (SWT); (g) membership functions for the permeability (PERM).
Mathematically, since the main idea is to optimize the margin then the quadratic optimization problem becomes
\[
\min_W \frac{1}{2} W^T W \quad \text{s.t.} \quad \left\{ \begin{array}{l}
y_j - (W^T \cdot \phi(X) + b) \leq \varepsilon + \xi_j \\
(W^T \cdot \phi(X) + b) - y_j \leq \varepsilon - \xi_j \quad \xi_j, \xi_i \geq 0, i = 1, \ldots, I
\end{array} \right.
\]

where \( \phi(X) \) is the kernel function, \( W \) is the margin and the pair \((x_i, y_i)\) is the training set. Then we add a bound in order to set the tolerance on errors number that can be committed:
\[
\min_{W} \frac{1}{2} W^T W + C \sum_{i=1}^{I} (\xi_i + \xi_i^*)
\]

This principle is similar to SVM for classification. Once it is trained, SVR will generate predictions using the following equation:
\[
f(X) = \sum_{i=1}^{I} \theta_i \phi(X, X_i) + b
\]

For the kernel, possible options are functions such as: Gaussian, polynomial, radial basis and sigmoid.

**Kernel function**: The kernel function is responsible for transforming the data set into hyperplane. The variables of the kernel must be computed accurately since they determine the structure of high-dimensional feature space which governs the complexity of the final solution. The most commonly used kernel functions in the literature are: polynomial, linear, Gaussian, and sigmoid.

**Regularization parameters (C)**: This determines the trade-off cost between minimizing the training error and minimizing the model’s complexity.

**The tube size of the \( \varepsilon \)-insensitive loss function (\( \varepsilon \))**: This is equivalent to the approximation accuracy placed on the training data.

### 3.5. Learning algorithm for artificial neural networks (ANN)

We consider briefly the common learning algorithm for the mostly used ANN, called the backpropagation (BP) algorithm which was published in the mid 1980s for multilayer perceptrons. Hornik et al. [44] suggested that if a sufficient number of hidden units are available then an MLP with one hidden layer having a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer can approximate any function to any degree of accuracy.

Backpropagation is a systematic method for training multilayer neural networks due to its strong mathematical foundation. The steps to implement the backpropagation algorithm are given as follows:

**Step 1**: The error signal at the output of neuron \( j \) at iteration \( n \) (i.e. presentation of the \( n \)th training pattern) is defined by
\[
e_k(n) = d_j(n) - y_j(n)
\]
where \( d_j(n) \) refers to the desired response for neuron \( j \) and \( y_j(n) \) is the function signal appearing at the output of neuron \( j \) and \( e_j(n) \) refers to the error signal at the output of neuron \( j \). The instantaneous value of the sum of squared errors is obtained by summing square error over all neurons in the output layer; which is written as:
\[
\xi(n) = \sum_{j=1}^{N} e_j^2(n)
\]

**Step 2**: The net internal activity level \( v_j(n) \) produced at the input of the nonlinearity associated with neuron \( j \) is therefore
\[
v_j(n) = \sum_{i=0}^{P} w_{ji}(n) y_i(n)
\]

where \( p \) is the total number of inputs (excluding the threshold) applied to neuron \( j \) and \( w_{ji}(n) \) denote the synaptic weight connecting the output of neuron \( i \) to the input of neuron \( j \) at iteration \( n \). Hence the output of neuron \( j \) at iteration \( n \) is
\[
y_j(n) = \psi_j(v_j(n))
\]

**Step 3**: The instantaneous gradient which is proportional to the weight correction term is given as:
\[
\frac{\partial \xi(n)}{\partial w_{ji}(n)} = -e_j(n) \psi_j'(v_j(n)) y_j(n)
\]

**Step 4**: The correction \( \Delta w_{ji}(n) \) applied to \( w_{ji}(n) \) is defined by the delta rule
\[
\Delta w_{ji}(n) = \eta \delta_j(n)
\]

**Step 5**: When neuron \( j \) is located in a hidden layer of the network, the local gradient is redefined as
\[
\delta_j(n) = e_j(n) \psi_j'(v_j(n))
\]

where the \( \delta_k \) requires the knowledge of the error signals \( e_k \) for all those neurons that lie in the layer to the immediate right of hidden neuron \( j \). The \( w_{kj}(n) \), consists of the synaptic weights associated with these connections. We are now ready to put forward the weight correction update for the back-propagation algorithm, which is defined by the delta rule:
\[
\Delta w_{ji}(n) = \eta \delta_j y_j
\]

The network performance is checked by monitoring the average squared error. The average square error is obtained by summing \( \xi(n) \) over all \( n \) and then normalizing with respect to \( N \) (number of
training patterns)

$$\xi_{av} = \frac{1}{N} \sum_{n=1}^{N} \xi(n)$$  (45)

The process is repeated a number of times for each pattern in the training set until the total output squared error converges to a minimum or until some limit is reached in the number of training iterations.

4. Empirical study, discussion and comparative studies

In order to carry out empirical study, real-industrial databases from five different Middle Eastern distinct reservoir wells were acquired. These wells are code-named in Table 1 together with their corresponding number of data points. Each of the databases is having well log inputs parameters that include sonic travel time (DT), Micro spherically Focused Log (MSFL), Neutron porosity (NPHI), total porosity (PHIT), bulk density (RHOB), and water saturation (SWT). The descriptive statistics of the wells are given in Table 2. From Tables 2 and 3, it could be deduced that well-1 to well 3 are better correlated viz-a-viz their core parameters whereas well-4 and 5 are less correlated in similar regards, with well-5 being the least correlated data. These can also be visualized in the pattern of final model results for these wells.

For the better understanding of the detail statistical information about the dataset use, we have decided to present some important statistical analysis and also carry out the estimation of the correlation coefficient between permeability and each of the well log input variables. Also, cross plots have been produced to support and enhance the statistical analysis. The results of the statistical analysis are all contained in Tables 2 and 3 and Fig. 3a–e.

### Table 2

<table>
<thead>
<tr>
<th>Code-name for wells</th>
<th>Number of cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well-1</td>
<td>477</td>
</tr>
<tr>
<td>Well-2</td>
<td>431</td>
</tr>
<tr>
<td>Well-3</td>
<td>356</td>
</tr>
<tr>
<td>Well-4</td>
<td>387</td>
</tr>
<tr>
<td>Well-5</td>
<td>203</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Code-name for wells</th>
<th>Number of cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well-1</td>
<td>477</td>
</tr>
<tr>
<td>Well-2</td>
<td>431</td>
</tr>
<tr>
<td>Well-3</td>
<td>356</td>
</tr>
<tr>
<td>Well-4</td>
<td>387</td>
</tr>
<tr>
<td>Well-5</td>
<td>203</td>
</tr>
</tbody>
</table>

Acronym used in the tables: sonic travel time (DT), Micro spherically Focused Log (MSFL), Neutron porosity (NPHI), total porosity (PHIT), bulk density (RHOB), and water saturation (SWT).

### Table 2

<table>
<thead>
<tr>
<th>Code name for wells</th>
<th>Number of cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well-1</td>
<td>477</td>
</tr>
<tr>
<td>Well-2</td>
<td>431</td>
</tr>
<tr>
<td>Well-3</td>
<td>356</td>
</tr>
<tr>
<td>Well-4</td>
<td>387</td>
</tr>
<tr>
<td>Well-5</td>
<td>203</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Code name for wells</th>
<th>Number of cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well-1</td>
<td>477</td>
</tr>
<tr>
<td>Well-2</td>
<td>431</td>
</tr>
<tr>
<td>Well-3</td>
<td>356</td>
</tr>
<tr>
<td>Well-4</td>
<td>387</td>
</tr>
<tr>
<td>Well-5</td>
<td>203</td>
</tr>
</tbody>
</table>

Acronym used in the table: sonic travel time (DT), Micro spherically Focused Log (MSFL), Neutron porosity (NPHI), total porosity (PHIT), bulk density (RHOB), and water saturation (SWT).
Correlation coefficient is used to determine whether two data sets are related, and if so, how strongly. The correlation coefficient ranges from +1, indicating a perfect positive linear relationship, to −1, indicating a perfectly negative linear relationship. The benefit of using correlation coefficient to measure the relationship between two variables as opposed to using covariance is that the unit of measurement does not matter.

The correlation coefficient has been used to reveal the intrinsic characteristic and pattern of the well log dataset used in this work. It indicates the nature of the relationship between each of the independent variable (well log input parameter) and the actual permeability. Table 3 contains the computed correlation coefficient for each of the six well log input variables vis-a-vis the target variable, which is permeability.

The higher the correlation coefficient the better, because it indicates how far each of the independent variables and the dependent variable relates and affects one another. In this sense, we can see that correlation report for well-1, well-2 and well-3 are better compared to the remaining two wells, with well-5 having the poorest correlation report. These account for the huge disparity in the performance of the presented models for the first four wells compared to the remaining ones.

Specifically we can notice a big difference between the proposed type-2 FLS and the other models for well-5. This arises as a result of the lowest correlation coefficient between permeability and the well log input variables, the maximum being 0.5437 unlike other wells where we have higher correlation coefficient. This confirms again the superiority of the proposed

![](https://via.placeholder.com/150)

**Fig. 7.** (a) Crossplots between core permeability and log inputs for well-1; (b) crossplots between core permeability and log inputs for well-2; (c) crossplots between core permeability and log inputs for well-3; (d) crossplots between core permeability and log inputs for well-4; (e) crossplots between core permeability and log inputs for well-5.
type-2 fuzzy logic system in such situation where there are uncertainties/irregularities in the dataset presented.

Furthermore, in order to make the underline nature of the well log dataset clearer, we have produced the crossplots between core permeability and each of the well log variables for all the five wells that we made use of in this work, as shown in Fig. 7a–e. The cross plot is just a diagramatical view of the correlation among the two variables plotted, which is the permeability versus each of the well log input variables.

4.1. Implementation process

In order to evaluate the performance of each modeling scheme investigated, each of the acquired databases is divided, using the stratified sampling approach, into training set and testing set. 80% of the data was used for training and building each model (internal validation) and the remaining 20% reserved for testing the model. For the testing and evaluation of the newly developed type-2 FLS framework, and to carry out effective comparative studies viz-a-viz type-1 FLS, artificial neural network (ANN) and support vector machines (SVM), the most common statistical quality measures that are utilized in data mining and petroleum engineering journals were employed in this study. The training sets were used to build the models while the testing sets were utilized in evaluating the predictive capability of the models. As for the implementation, we did not use any ready-made software, the entire coding has been done using MATLAB though some MATLAB inbuilt functions,
most especially in the case of ANN, and few others made available online, particularly those of SVM, have been called and used in some cases. Also part of the type-2 fuzzy logic functions made available in [12] was also made use of.

In the case of the type-2 FLS based model, the implementation process proceeded by supplying the system with the available input data sets, one sample at a time, and the rules and membership functions is automatically leaned from the available input data. Gaussian membership function has been used based on two different learning criteria that include least squares and back-propagation. The same combination was utilized in training FLS membership function parameters. Further details on initializing, training and validating type-2 FLS have been presented in Section 3, and additional details could be found in [12,34,38]. The outputs from the type-2 based model are reduced type-2 fuzzy sets, which generate prediction intervals for the permeability without extra computational cost. The type-2 reduced set is finally defuzzified to produce the final crisp output representing the predicted permeability. The type-1 FLS have also been implemented following similar procedures except that the fuzzy sets used are those of type-1.

As for the artificial neural network implementation, a single hidden layer feedforward neural network based on back propagation (BP) learning algorithm with both linear and sigmoid activation functions was utilized. As usual, the initial weights were produced randomly with the learning epoch set to 1000 or 0.001 goal error and 0.01 learning rate. Regarding support vector machines implementation, the support vector regression (SVR) was used and an optimization was carried out to arrive at the best parameters that include the regularization factor $C = 450$, $\varepsilon = 0.2$ while the kernel option is set to be polynomial.

4.1.1. Optimal parameters search procedure for SVM and ANN

The optimal parameters search procedure for SVM and ANN are described as follows:

4.1.1.1. Optimal parameters search procedure for SVM. The parameters associated with the SVM were optimized through a test-set-cross-validation on the available data set. The details of the test-set-cross-validation for optimizing the SVM parameters go thus: for each run of generated training and testing set, the values of RMSE and correlation coefficient were monitored for a group of parameters $C$ (bound on the Lagrangian multiplier) and $\lambda$ (conditioning parameter for QP methods). Searching through all possible values of the parameters in a given range will identify the best performance measures and the corresponding values of the parameters for the fixed set of features. In our experiment, this process was repeated for every SVM kernel option available, each time with an incremental step of parameters. The optimal values of the parameters and the kernel option associated with the best performance measure were identified. A summary of the procedure is as follows:

Step 1: Choose the initial kernel option from the list of available kernel options.

Step 2: Identify the best values of the parameters $C$ and $\varepsilon$ through a test-set-cross-validation and store the corresponding performance measures.

Step 3: If there is no kernel option left, then go to Step 4. Otherwise, add the next kernel option and go to Step 2.

Step 4: Identify the best performance measure and its associated kernel option and the parameters values.

Step 5: Use the optimized kernel option and the parameters values to train the final SVM.

Step 6: Calculate the performance measures for both the training and testing sets using the system obtained in the previous Step 5.

This can be presented in mathematical form as follows:

Let the set $A$ contains all the possible kernel options, the element of $A$ is of the shape $A(j)$, where $j$ is the kernel function number, $i$ is the index for selected value of $C$ and $k$ is the index for selected value of $\lambda$. $nf$ is the total number of kernel functions available, $nc$ is the maximum value of $C$ assumed and $nl$ is the maximum value of $\lambda$ assumed. Also $pm$ represents performance measure taken, $ix$ represents index for best kernel function, $jx$
represents index for best value of $C$, and $kx$ represents index for best value of $\lambda$. The algorithm then goes thus:

Algorithm 2 (Optimal parameters search procedure for SVM).

```
Initialization: $jv = 0$, $nx = 0$, $ix = 0$, $kx = 0$
for $i = 1 \rightarrow nf$
  for $j = 1 \rightarrow nc$
    for $k = 1 \rightarrow nL$
      if $pm = f(A(i,j,k))$ (Performance measure for the present parameters combination)
        if $p_{m}$ is better than $vx$ then $vx = pm$
        $ix = i$, $jx = j$, $kx = k$ (storing the index of the better parameter)
      end
    end
  end
end
```

4.1.1.2. Optimal parameters search procedure for ANN. As for the case of ANN, an optimized-parameter search method similar to that of SVM was also used to arrive at the final ANN structure. This procedure is contained below.

The parameters associated with the ANN were optimized through a test-set-cross-validation on the available data set. The details of the test-set-cross-validation for optimizing the ANN parameters goes thus: for each run of generated training and testing set, the values of RMSE and correlation coefficient were monitored for a group of parameters that include activation function type and number of hidden layers. Searching through all possible values of the parameters in a given range will identify the best performance measures and the corresponding values of the parameters for the fixed set of features. To get the best parameters, this process was repeated for every activation function available, each time with an incremental step of parameters (like number of hidden layers). The optimal values of the parameters and the activation function associated with the best performance measure were identified. A summary of the procedure is as follows:

Step 1: Choose the initial “activation function” option from the list of available options.
Step 2: Identify the best values of the number of hidden layers and hidden neuron through a test-set-cross-validation and store the corresponding performance measures.
Step 3: If there is no activation function option left, then go to Step 4. Otherwise, add the next activation function option and go to Step 2.
Step 4: Identify the best performance measure and its associated parameters values.
Step 5: Use the optimized activation function option and the parameters values to train the final ANN.
Step 6: Calculate the performance measures for both the training and testing sets using the system obtained in the previous Step 5.

This is presented in mathematical form as follows:

Let the set $A$ contains all the possible activation functions options, the element of $A$ is of the shape $A(i,j)$, where $i$ is the activation function number, $j$ is the selected number of layers, $nf$ is the total number of activation functions available, and $nL$ is the maximum number of hidden layer assumed. Also $pm$ represents performance measure taken, $ix$ represents index for best activation function, $jx$ represents index for best number of layer. The algorithm then goes thus:

Algorithm 3 (Optimal parameters search procedure for ANN).

```
Initialization: $jv = 0$, $nx = 0$, $ix = 0$
for $i = 1 \rightarrow nf$
  for $j = 1 \rightarrow nL$
    $pm = f(A(i,j))$ (Performance measure for the present parameters combination)
    if $pm$ is better than $vx$ then $vx = pm$
    $ix = i$, $jx = j$, $kx = k$ (storing the index of the better parameter)
end
```

4.2. Results and discussion

The results of comparisons, between the newly proposed type-2 FLS and the three other methods namely artificial neural network (ANN) support vector machines (SVM) and type-1 fuzzy logic system, using the external validation checks (testing results) were summarized in Tables 4–8. Ideally, the best forecasting scheme should have the lowest root mean squared error (RMSE), the lowest average absolute percent relative error ($E_a$) and the highest correlation coefficient ($R^2$). It could be easily observed that type-2 FLS based intelligence modeling scheme outperforms others throughout the reported results.

Results summarized in Tables 4–8 are further represented in scatter plots of Fig. 8 in order to facilitate making ‘at a glance decision’ on which of the methods presented is better in performance.

From the tables and the scatter plots presented, it could be easily observed, for instance in the case of well-1 with 477 dataset; type-2 fuzzy based model has the highest correlation coefficient, $R^2 = 0.9543$, which represent 7.4% improvement over that of support vector machines (SVM) and 1.9% improvement over that of artificial neural network (ANN). In terms of root mean squared error (RMSE), type-2 FLS had 42% improvement over SVM and 25% over ANN, while in terms of absolute percent relative error ($E_a$), type-2 FLS had 70.1% improvement over SVM and 40.4% over ANN. Other reported results also follow similar trends with type-2 FLS taking the lead always.

### Table 4

<table>
<thead>
<tr>
<th>Prediction methods</th>
<th>$R^2$</th>
<th>RMSE</th>
<th>$E_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-2 fuzzy model</td>
<td>0.9543</td>
<td>0.2735</td>
<td>70.3851</td>
</tr>
<tr>
<td>Type-1 fuzzy model</td>
<td>0.9359</td>
<td>0.3753</td>
<td>195.5063</td>
</tr>
<tr>
<td>ANN model</td>
<td>0.9362</td>
<td>0.3652</td>
<td>118.1036</td>
</tr>
<tr>
<td>SVM model</td>
<td>0.8886</td>
<td>0.4717</td>
<td>240.5559</td>
</tr>
</tbody>
</table>

### Table 5

<table>
<thead>
<tr>
<th>Prediction methods</th>
<th>$R^2$</th>
<th>RMSE</th>
<th>$E_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-2 fuzzy model</td>
<td>0.9398</td>
<td>0.3631</td>
<td>239.0630</td>
</tr>
<tr>
<td>Type-1 fuzzy model</td>
<td>0.9326</td>
<td>0.3454</td>
<td>225.3241</td>
</tr>
<tr>
<td>ANN model</td>
<td>0.9175</td>
<td>0.4230</td>
<td>255.8370</td>
</tr>
<tr>
<td>SVM model</td>
<td>0.8735</td>
<td>0.5066</td>
<td>284.0859</td>
</tr>
</tbody>
</table>

### Table 6

<table>
<thead>
<tr>
<th>Prediction methods</th>
<th>$R^2$</th>
<th>RMSE</th>
<th>$E_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-2 fuzzy model</td>
<td>0.9210</td>
<td>0.4188</td>
<td>59.4337</td>
</tr>
<tr>
<td>Type-1 fuzzy model</td>
<td>0.9146</td>
<td>0.4375</td>
<td>64.6357</td>
</tr>
<tr>
<td>ANN model</td>
<td>0.8201</td>
<td>0.6197</td>
<td>72.7040</td>
</tr>
<tr>
<td>SVM model</td>
<td>0.8741</td>
<td>0.5194</td>
<td>84.8492</td>
</tr>
</tbody>
</table>
Table 7
Testing results for well-4 with 387 dataset. $R^2$, correlation coefficient; SD, standard deviation; $E_a$, average absolute percent relative error; RMSE, root mean squared.

<table>
<thead>
<tr>
<th>Prediction methods</th>
<th>$R^2$</th>
<th>RMSE</th>
<th>$E_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-2 fuzzy model</td>
<td>0.89875</td>
<td>0.436357</td>
<td>83.66692</td>
</tr>
<tr>
<td>Type-1 fuzzy model</td>
<td>0.85409</td>
<td>0.54004</td>
<td>153.339</td>
</tr>
<tr>
<td>ANN model</td>
<td>0.763981</td>
<td>0.680512</td>
<td>258.9238</td>
</tr>
<tr>
<td>SVM model</td>
<td>0.814307</td>
<td>0.598978</td>
<td>233.9014</td>
</tr>
</tbody>
</table>

Table 8
Testing results for well-5 with 203 dataset. $R^2$, correlation coefficient; SD, standard deviation; $E_a$, average absolute percent relative error; RMSE, root mean squared.

<table>
<thead>
<tr>
<th>Prediction methods</th>
<th>$R^2$</th>
<th>RMSE</th>
<th>$E_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type-2 fuzzy model</td>
<td>0.919375</td>
<td>0.414592</td>
<td>84.70893</td>
</tr>
<tr>
<td>Type-1 fuzzy model</td>
<td>0.8230</td>
<td>0.6153</td>
<td>89.0989</td>
</tr>
<tr>
<td>ANN model</td>
<td>0.67565</td>
<td>1.116008</td>
<td>158.4053</td>
</tr>
<tr>
<td>SVM model</td>
<td>0.6253</td>
<td>0.842976</td>
<td>101.6796</td>
</tr>
</tbody>
</table>

For the case involving well-2 with 431 data points, type-2 fuzzy FLS had 7.6% improvement over SVM and 2.4% over ANN in terms of correlation coefficients. In terms of root mean squared error (RMSE), type-2 FLS had 28.3% improvement over SVM and 14.2% over ANN, while in terms of absolute percent relative error ($E_a$), type-2 FLS had 15.9% improvement over SVM and 6.6% over ANN. Similarly, in the case of well-3 with 356 datasets, type-2 fuzzy FLS had 12.3% improvement over ANN and 5.4% over SVM in terms of correlation coefficients. In terms of root mean squared error (RMSE), type-2 FLS had 32.4% improvement over ANN and 19.4% over SVM, while in terms of absolute percent relative error ($E_a$), type-2 FLS had 30% improvement over ANN and 18.3% over ANN.

Furthermore, results for well-4 follow usual trend with type-2 FLS taking the lead. In terms of correlation coefficients, type-2 fuzzy FLS had 17.6% improvement over ANN and 10.4% over SVM. In terms of root mean squared error (RMSE), type-2 FLS had 35.9% improvement over ANN and 14.2% over SVM, while in terms of absolute percent relative error ($E_a$), type-2 FLS had 67.7% improvement over ANN and 64.2% over SVM. Similarly, in the case of well-5, type-2 fuzzy FLS had 36.1% improvement over ANN and 47% over SVM in terms of correlation coefficients. In terms of root mean squared error (RMSE), type-2 FLS had 62.9% improvement over ANN and 50.1% over SVM, while in terms of absolute percent relative error ($E_a$), type-2 FLS had 46.5% improvement over ANN and 16.7% over SVM.

As for the performance of type-2 FLS compared to type-1 FLS, we can notice a steady and consistent superiority of type-2 FLS over those of type-1 FLS. For instance in cases of well-1 to well-5, type-2 FLS had improvement of 1.96%, 0.8%, 0.7%, 5.2% and 11.7% over type-1 FLS respectively. Similar improvement trends are also obtained for other quality measures use as can be easily seen in the tables provided and the scatter plots given earlier. This has further confirmed the earlier claims in earlier publications about the superiority of type-2 FLS over type-1 FLS.

Furthermore, it could be easily noted, that there is huge disparity in the performance of type-2 FLS model compared to others for some wells, particularly for the case of well-5. This is by no accident; rather it is a function of the level of uncertainties or irregularities in the dataset, which are special situations where the power of type-2 FLS can easily be felt. We can easily observe here that even type-2 FLS model has gone as far as having 11.7% improvement over that of type-1 FLS. With this reality, we can surely expect that other models like ANN and SVM cannot survive such uncertainties onslaught as demonstrated by their very low performance compared to the proposed model. As for the case of well-1 and well-2 where ANN has a close performance to type-2 FLS, this actually depends on the nature of the data set. If the level of uncertainties or irregularities is very low or absent, then there is the tendency for individual model to perform according to their normal ability. This is the case regarding these two wells. But on encountering data with high uncertainties then each model uncertainty handling capability will surely become evident. This is the case in the other wells where the level of uncertainties/irregularities is very high, and hence other models performance became very low. Details statistical analysis to expose the nature of the dataset have been presented in Section 4 with adequate explanations supported with tables and figures.

From the overall reported experimental results, it could be easily noted that type-2 FLS performed outstandingly better in all fronts. This is evident as its quality measure values are consistently better than others, with consistent and stable performance. This indicates that type-2 FLS is able to consistently deal with the nature of reservoir well log data due to its ability to cater for all forms of uncertainties.

4.3. Permeability prediction intervals’ estimation using type-2 fuzzy logic system

One important additional benefit of type-2 FLS worth mentioning at this point is that, compared with earlier used models, the
proposed type-2 FLS will generate, in addition to the permeability predictions, the prediction intervals effortlessly as its by-product and without any extra computational cost. It achieved this through its type-reduction process that generates the intervals as the by-product. See [12,41,42] for further details. A sample permeability prediction intervals generated in this work is shown in Fig. 6. This, indeed, is another great contribution of this work. The importance of interval based modeling cannot be overemphasized. Nguyen et al. [42] have argued that the use of intervals is necessary to describe an expert's degree of belief, and many people believe that assigning an exact number to an expert's opinion is too restrictive whereas the assignment of an interval of values is more realistic. Conventional regression methods use real numbers as input and produce real numbers as results without any indication of the accuracy. Interval based computation, as in interval type-2 fuzzy logic system, uses interval elements throughout the computation and produces intervals as output with the guarantee that the true results are contained in them. This is an advantage over the other methods that are currently in use (Fig. 9).

5. Conclusion and recommendation

In this work, five distinct industrial dataset collected from Middle Eastern reservoir wells were used in investigating the feasibility, performance, and accuracy of the proposed type-2 FLS based modeling scheme as a new framework for predicting the permeability of carbonate reservoirs from well log. The following conclusions and recommendations could be drawn based on previous analysis, discussions, deep investigation, experiments, and comparative studies in this work. A new modeling scheme based on the type-2 FLS has been investigated, developed and implemented, as predictive solution that takes care of all forms of uncertainties, for predicting permeability from well logs. Validation of the framework has been carried out using real industrial well log databases. In-depth comparative studies have been carried out between this new framework and the standard neural networks and support vector machines. Also the proposed method has been compared with the type-1 fuzzy logic system to indicate the improvement provided by the new model, and the results confirmed the superiority of type-2 FLS over type-1 FLS with improvement of up to 11.7% in terms of correlation coefficient. Empirical results from simulations show that the proposed model outperformed all the other compared models in all fronts. Specifically, the proposed model achieve up to 36.1% improvement over other models in terms of correlation coefficient, up to 62.9% and 67.7% in terms of root mean squared error (RMSE) and absolute percent relative error (E_p) respectively. Given any new data, the proposed type-2 FLS will be able to handle uncertainties that might be present and perform the required prediction effectively with stable and consistent results. As an additional contribution from this work, it can be said here that, this work has presented a type-2 FLS based model that will generate not only the target permeability forecast but also prediction intervals without any extra computational cost. This is really an interesting contribution because it has been argued in [42] that the use of intervals is necessary to describe an expert’s degree of belief and that many people believe that assigning an exact number to an expert’s opinion is too restrictive whereas the assignment of an interval of values is more realistic. Thus, the importance of interval based computation and prediction cannot be overemphasized. Motivated by the success of this work, it is suggested as a form of future work recommendation, that other models like that of PVT properties, porosity, history matching, lithofacies and other reservoir engineering properties could be built using this framework.

Acknowledgements

The authors would like to thank the reviewers for providing valuable comments that have improved the manuscript. Furthermore, the University Teknologi Malaysia (UTM) and Ministry of Higher Education (MOHE), Malaysia under FRGS/1/105/UTM/02/ 10 and King Fahd University of Petroleum and Minerals (KFUPM) Saudi Arabia are hereby acknowledged for some of the facilities utilized during the course of this research work. The support from KACST project 08-OIL82-4 is gratefully acknowledged.

References


