Application of machine and deep learning techniques to estimate NMR-derived permeability from conventional well logs and artificial 2D feature maps

Milad Masroor1 · Mohammad Emami Niri1 · Amir Hossein Rajabi-Ghozloo1 · Mohammad Hassan Sharifinasab1 · Mozdeh Sajjadi1

Received: 17 January 2022 / Accepted: 19 March 2022 / Published online: 6 April 2022
© The Author(s) 2022

Abstract
Nuclear magnetic resonance (NMR) logs can provide information on some critical reservoir characteristics, such as permeability, which are rarely obtainable from conventional well logs. Nevertheless, high cost and operational constraints limit the wide application of NMR logging tools. In this study, a machine learning (ML)-based procedure is developed for fast and accurate estimation of NMR-derived permeability from conventional logs. Following a comprehensive preprocessing on the collected data, the procedure is trained and tested on a well log dataset, with selected conventional logs as inputs, and NMR-derived permeability as target, shallow and deep learning (DL) methods are applied to estimate permeability from selected conventional logs through artificial production of NMR-derived information from the input data. Three supervised ML algorithms are utilized and evaluated, including random forest (RF), group method of data handling (GMDH), and one-dimensional convolutional neural network (1D-CNN). Additionally, a modified two-dimensional CNN (named as Residual 2D-CNN) is developed which is fed by artificial 2D feature maps, generated from available conventional logs. The hyper-parameters of the ML and DL models are optimized using genetic algorithm (GA) to improve their performances. By comparing the output of each model with the permeability derived from NMR log, it is illustrated that nonlinear machine and deep learning techniques are helpful in estimation of NMR permeability. The obtained accuracy of RF, GMDH, 1D-CNN and Res 2D-CNN models, respectively, is 0.90, 0.90, 0.91 and 0.97 which indicate that Res 2D-CNN model is the most efficient method among the other applied techniques. This research also highlights the importance of using generated feature maps for training Res 2D-CNN model, and the essential effect of the applied modifications (i.e., implementing residual and deeper bottleneck architectures) on improving the accuracy of the predicted output and reducing the training time.

Keywords Permeability · Conventional well logs · NMR log · Machine learning · Deep learning · Artificial feature map · Residual -CNN

Introduction
Characterizing the transport and storage properties of a reservoir is a critical step in its reserve assessment and the formation evaluation. In particular, obtaining accurate description of the permeability distribution is important in design of the development plans and efficient management of the production processes from oil and gas fields. Beside its vital importance to oil and gas industry, permeability estimation is crucial for applications concerning fluid flow in subsurface, such as CO2 storage (Zhang et al. 2017), water supply management (Kang et al. 2017), and geothermal systems’ development (Siler et al. 2019).
Laboratory core measurements and well pressure transient testing (including pressure buildup (Horner 1951), repeat formation testing (Jensen and Mayson 1985) and drill-stem testing (Van Poollen 1961)) are common sources for reservoir permeability estimation. Both approaches are time-consuming and costly and thus can only provide information on reservoir permeability in few drilled wells (Mohaghegh et al. 1995). As an alternative approach, geophysical well logging can be used for obtaining the permeability of porous medium. The conventional well logs have shown poor performance in providing direct and accurate measurement of permeability, as they are traditionally used to measure porosity, and the corresponding permeability values can only be predicted by empirical or semi-empirical correlations (Kozney 1927; Carman 1937; Archie 1942; Biot and Willis 1957; Crain 1986; Amaeful et al. 1993). In the past decades, an advanced well logging technique referred to as nuclear magnetic resonance (NMR) has become a popular source of information for more accurate estimation of permeability (Maximiano and Carrasquilla 2011). Unlike conventional logs which are highly sensitive to mineralogy and respond to both the solid and the fluid components of the medium, NMR logging measurements correspond to the volume, composition, viscosity, and distribution of the fluids in porous media. Therefore, NMR logs are associated with hydraulic conductivity of the rocks with more accuracy than the other logs (Maximiano and Carrasquilla 2011). Some obstacles restrict wide application of NMR logging tools: It is more expensive to be acquired, time-consuming to be analyzed, and cannot be run in cased-hole wells. These obstacles limit the availability of NMR data. To take advantage of the NMR logs in estimation of reservoir characteristics such as permeability, a key approach has been the synthetic generation of NMR logs (or the NMR-derived outputs) from the more widely accessible conventional logs.

Machine learning (ML) and artificial intelligence (AI) can provide cost-effective, fast and often physically reasonable solutions to geoscience and engineering problems. Development of more complicated forms of these techniques along with the use of artificial neural networks (ANN) concepts has resulted in formation of more robust outlooks for automated analyses and has improved the performance of automated reservoir characterization, especially in permeability prediction (Richardson 2000; Ogilvie et al. 2002; Kadkhodaie Ilkhchi et al. 2006; Rezaee et al. 2007; Kadkhodaie Ilkhchi et al. 2009; Labani et al. 2010; Haghshenas et al. 2020; Amiri et al. 2021; Gohari et al. 2021). In this context, Mohaghegh et al. (2000) proposed an ANN model to generate synthetic NMR log from density, spontaneous potential (SP), induction and gamma ray (GR) logs. Al-Anazi and Gates (2012) studied the application of support vector regression (SVR) method for estimation of porosity and permeability from a set of conventional well logs. Support vector machine (SVM), fuzzy logic (FL), and back propagation neural network (BP-NN) have been used by Rafik Baouche et al. (2017) for prediction of permeability in a sandstone reservoir. A combination of NMR and conventional log data was used, and satisfying prediction results were reported. FL and BP-NN regression models showed better performance during validation stage. Also, Al Khalifah et al. (2020) presented a case study on a tight carbonate reservoir comparing the efficiency of ANN, genetic algorithm (GA), and several theoretical/empirical ‘benchmark’ models for permeability prediction. They showed that ANN and GA have better performance and accuracy compared to the ‘benchmark’ models. To set the weights and biases of ML methods meta-heuristic optimization algorithms such as GA, particle swarm optimization (PSO), and imperialist competitive algorithm (ICP) have been recently implemented. For example, GA-optimized ANN and FL techniques have been used for synthetic development of NMR parameters from conventional wireline logs by Briones and Carrasquilla (2013). Ahmadi and Chen (2019) carried out a comparative study on the application of conventional ANN, Fuzzy Decision Tree (FDT), and Least-Square SVM (LSSVM), optimizing them by PSO, ICP, GA, and a combination of these three algorithms, for permeability prediction. In their study, LSSVM model optimized by a hybrid PSO and GA (HGAPSO-LSSVM) algorithm showed the most accurate prediction.

Although ML methods have shown acceptable performance in prediction of permeability, the recent advances in digital rock imaging and progresses in deep learning and image-based AI methods have driven the studies toward new approaches. Convolutional neural networks (CNNs) are a subset of deep learning methods, which effectively deal with both numerical (Tsantekidis et al. 2017; Malek et al. 2018) and image-based datasets (Valueva et al. 2020). Due to the CNN models’ capability in efficient feature extraction, and prevention of information loss and over-fitting, they have been used in many classification problems, and recently in regression problems (Alqahthani et al. 2018; Liu et al. 2019; Misbahuddin 2020; Chawshin et al. 2021; Kwon et al. 2021; Razak et al. 2021). Two forms of CNNs have been utilized for permeability prediction. The first form is compatible with 1D numerical dataset (i.e., wireline logs). For instance, Salehi et al. (2020) applied multi-layer perceptron (MLP), 1D-CNN, and recurrent neural network (RNN) methods to construct a regression model for estimation of permeability in coal from mud logging data and showed that 1D-CNN model provided accurate results. The other proposed form of CNN is 2D-CNN which is trainable with images and 2D datasets. Zhong et al. (2019) trained a 2D-CNN regression model for estimating permeability in an oil field. They converted geophysical well logs to geological feature images.
and used these images as inputs of a plain (or sequential) 2D-CNN.

In this paper, ML/DL methods of (1) random forest (RF), (2) group method of data handling (GMDH), and (3) 1D-CNN are used to predict NMR-derived permeability based on conventional wireline logs. In addition, a modified version of 2D-CNN (named as Residual 2D-CNN), which is trained by artificially produced 2D feature maps, is also implemented. The 1D and 2D-CNN models have been chosen due to their specific capability in efficient feature extraction and prevention of information loss during feature engineering step. The 2D-CNN method is modified by applying residual and deeper bottleneck architectures. These modifications improved the speed and accuracy of the original 2D-CNN model and utilizing a more comprehensive dataset, helped to achieve a more reliable prediction in our study.

**Machine and deep learning methods**

In this section, the employed machine and deep learning methods for permeability estimation from conventional wireline logs are briefly described.

**Random forest (RF)**

RF is a supervised ML algorithm applicable for solving both classification and regression problems. The RF method is, in fact, an ensemble learning technique which means that it combines different classifiers to solve complex problems (Jaiswal and Samikannu 2017; Sagi and Rokach 2018). RFs work by building plenty of decision trees during the training stage. The decision trees generate a ‘forest’ that is trainable through bagging or bootstrap aggregating algorithms (Breiman 2001). The output of RF is either a class chosen by the majority of the trees for a classification task, or an average of the estimation proposed by individual trees for a regression problem. Therefore, it is expected that a higher number of trees leads to a more precise outcome. The most important hyper-parameters of a RF model include the number of trees, the maximum depth of the trees and the minimum number of samples required to split the internal node: though RF can provide reliable predictions with no need for hyper-parameter tuning (Breiman 2001).

**Group method of data handling (GMDH)**

GMDH is a self-organized AI algorithm with the capability to efficiently solve complex and/or nonlinear problems (Hwang 2006; Amanifard et al. 2008; Ebtehaj et al. 2015). The method benefits from an inductive procedure, meaning that it finds the optimum solution by sorting-out the feasible variants. Ivakhnenko (1971), first, used the GMDH algorithm for modeling of a complicated system with multiple-inputs and a single-output. The GMDH algorithm automatically optimizes the NN parameters including the number of layers, the number of neurons within each of the hidden layers, input variables, and the model structure. Another obvious advantage of GMDH is its strong mathematical support, since a nonlinear function known as the Volterra series is used for relating the input and output variables, expressed as:

\[
\hat{y} = a_0 + \sum_{i=1}^{m} a_i x_i + \sum_{i=1}^{m} \sum_{j=1}^{m} a_{ij} x_i x_j + \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} a_{ijk} x_i x_j x_k + \ldots
\]

where \(m\) is the number of the function components, and \(a_i\) are the unknown coefficients of Volterra series. The aim of a GMDH algorithm is to determine \(a_i\). This is often carried out by using regression methods for each pairs of \(x_i\) and \(x_j\) input variables (Farlow 2020). The general form of Eq. (1) might be simplified as Eq. (2) by applying a quadratic polynomial system:

\[
\hat{y} = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + a_4 x_j^2 + a_5 x_i x_j
\]

By considering the principles of the least-squares error, \(\hat{y}\) can be written as follows:

\[
E = \frac{\sum_{i=1}^{M} (y_i - \hat{y}_i)^2}{M} \rightarrow \text{Min}
\]

\[
y_i = f(x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in}) \quad (i = 1, 2, 3, \ldots, M)
\]

where \(M\) is the number of data pairs, \(y_i\) is the actual output, and \(\hat{y}_i\) is the predicted output.

**Convolutional neural network (CNN)**

CNNs are the most popular class of deep learning structures, and their widespread application is mostly related to image processing tasks such as remote sensing (Cheng et al. 2016; Ghamisi et al. 2016), biomedical imaging (Srinivas et al. 2016; Cui et al. 2016), and biometrics (Nogueira et al. 2016; Rikhtegar et al. 2016). CNN architecture is composed of three main layer types: (1) convolutional layer, (2) subsampling (pooling or downsampling) layer, and (3) fully connected (FC) layer.

**Convolutional layer**

The basic building block of the CNN is the convolutional layer, which consists of three components, input data, kernel, and feature map. The input data have three dimensions of height, width, and channel. The second component, kernel, moves across the input image receptive fields, checking the
presence of a certain feature. This process is named convolution. By moving kernel with specified stride on image pixels, the dot products are calculated between the kernel and the input image. Finally, a feature map is created from the series of dot products (Fig. 1). After each convolution operation, a rectified linear unit (ReLU) transformation is applied to the output feature map, to introduce nonlinearity to the model (Venkatesan and Li 2017).

Subsampling layer

The subsampling layer aims to decrease the number of parameters and reduce dimensionality of the input. In this layer, kernel sweeps through the entire input without weights, then applies an aggregation function to the receptive field values, and populates the output array (Ciresan et al. 2011). Max pooling and average pooling are two of the main techniques used in the subsampling procedure (Ciregan et al. 2012).

Fully connected layer

FC layer connects each node in the output layer to a node in the previous layer, a functionality that does not directly occur in partially connected convolutional and subsampling layers. In this study, the CNN structure is used in two forms of 1D-CNN and a modified version of 2D-CNN (named as Res 2D-CNN).

1D-Convolutional neural network (1D-CNN)

In addition to the image-based datasets, CNN architecture has shown an acceptable performance in dealing with numerical or structural datasets (such as wireline logs), and 1D-CNN structure is the best choice for this purpose (Zhang et al. 2022). 1D-CNN, used in this study, has a similar basic architecture to that of the original CNN. Feeding 1D input data to the CNN model needs using 1D filters on convolutional layers, and the forward and backward propagation equations are required to be accordingly modified during the training stage. Consider a set of training samples resembled by $X = [x_1, x_2, x_3, \ldots, x_N]^T$, where $N$ is the number of the training samples, and each vector $x_i$ represents a feature in the space of $d$-dimensional measurements. Correspondingly, the matrix of real-output target can be considered as $Y = [y_1, y_2, y_3, \ldots, y_N]^T$. A 1D-CNN is composed of $L$ layers, each layer constituting of $m^l$ feature signals and carrying out both convolution and subsampling operations. The subsampling factor ($SS$) is assumed to be equal to 2 ($SS = 2$) (Malek et al. 2018). A general architecture of 1D-CNN model, used in this study, is illustrated in Fig. 2.

Residual 2D-convolutional neural network (Res 2D-CNN)

2D-CNN architecture is highly compatible with the image-based datasets. In this study, in order to enhance the model's performance with regard to our specific problem (permeability prediction), three modifications are applied to the general structure of 2D-CNNs, which are described hereafter. The first modification is the elimination of the subsampling layer, aiming at the prevention of information loss. The second modification is adding the residual architecture. Residual structure was originally introduced by He et al. (2016) in order to enhance the accuracy and reliability of CNN models. The residual structure increases the depth of the model without introducing extra parameters and computational complexity to the training process. In networks without residual architecture, the weights on each CNN layer have no reference to the weights of the previous layers which results in higher training times. The modified structure brings in the residual shortcut connections to help the training process of very deep CNN models. Figure 3 shows a singular residual block, illustrating the structure of the residual unit and the identity shortcut. The output of previous layers, $x$, is fed into the convolutional block as input. The output would be transformed $x(i.e,F(x))$. Applying residual unit adds the original data, $x$, to the transformed output. As a result, the output of residual convolutional block would be $F(x) + x$. This network modification leads to a more efficient optimization of activation functions ($F$) by conserving the references to the input, $x$, and involving linear patterns along with nonlinear patterns through feature extraction stage. The
third modification is utilizing deeper bottleneck architecture. For this purpose, we used a stack of 3 convolutional layers instead of 1 (Fig. 4). Using bottleneck structure increases the depth of the network and decreases the number of trainable parameters which leads to a lower training time. In this study, adding bottleneck architecture decreased the number of trainable parameters from 2,025,781 to 557,749 which means 72% fewer parameters. Figure 5 illustrates a schematic overview of the developed Res 2D-CNN architecture based on the discussed modifications.

**Methodology**

As mentioned earlier, in this study, ML/DL-based procedures are developed for estimating NMR-derived permeability from conventional wireline logs. The successive steps of the procedures are illustrated in Fig. 6. The workflow can be explained under the following two main stages:

- **Stage 1**: Dataset gathering, preprocessing (including image generation from numerical well log data)
- **Stage 2**: Model training and evaluation
Res 2D-CNN is shown in Fig. 7. It can be clearly observed that the correlation between RHOB and log(K_{NMR}) is much higher than that between RHOB and K_{NMR}.

**Feature selection**

Feature selection is a process that tries to reduce the number of input variables of predictive models in order to minimize the computational cost and improve the model performance. In this study, two kinds of feature selection schemes are implemented: (1) linear relationship investigation and (2) nonlinear relationship investigation.

The linear relationship investigation method checks out the linear relationship between each input variable and the target feature, using correlation coefficients. It helps to select the input variables with stronger linear correlation with the target feature. Figure 8 is the feature correlation heat map that shows correlation coefficients between NMR-derived permeability and the available conventional logs. In our case, the correlation between permeability and CALI, DCAL, and DRHO logs is too weak. So, these three logs (CALI, DCAL, DRHO) are removed, and the other seven logs (LLD, LLS, NPHI, PEF, RHOB, CGR, SGR) are selected as input variables.

Nonlinear relationship investigation (NRI) is another method for choosing the best input features for training the model. One of the subsets of NRI is forward selection which starts training the ML algorithm by individually adding each input feature to the model and then checking the algorithm’s functionality. The input variable with the best model outcomes is then kept, and this process is repeated until selecting the best features. The results of applying the NRI method on our dataset were consistent with the outcomes of the linear relationship investigation method. Therefore, the set of conventional well logs of (LLD, LLS, NPHI, PEF, RHOB, CGR, SGR) is selected as the best input features for training the networks.

**Normalization**

Normalization is generally applied when the features have different data ranges, while they all need to be in a unique range. For example, comparing the RHOB and LLD log data (shown in Table 1), the range of the data in RHOB log is between 2 and 3, whereas the range of LLD log’s data is between 0.65 and 236. Obviously, the effect of LLD log would be more prominent than the RHOB log, while theoretically, it may not be true. There are several techniques for normalizing the scales without distorting the differences in the ranges of the values such as linear scaling, clipping, log scaling, and Z-score. In this study, the linear scaling technique is used for this purpose (Eq. (5)). The summary statistics of the obtained results are reported in Table 2.

---

**Stage 1**

**Dataset gathering and preprocessing**

A set of log measurements from three wells (w_1, w_2, and w_3) of a carbonate reservoir in southwest Iran is used as input data. The log data include neutron porosity (NPHI), bulk density (RHOB), photo-electric factor (PEF), spectral gamma ray (SGR and CGR), deep lateral-log (LLD), shallow lateral-log (LLS), caliper (CALI), differential caliper (DCAL), and delta density (DRHO). The permeability derived from NMR logs is considered as the target data. The logs from two wells are used for training the algorithm, and the third well’s data are used for verification of the model. The field data collection is often weakly controlled, which may result in out-of-range values, missing values and incorrect data combinations. Consequently, the raw dataset is required to be preprocessed before feeding into ML/DL algorithms. In this study, after missing data detection and outlier removal, the following preprocessing steps are applied to the acquired numerical well log dataset:

1. Transformation
2. Feature Selection
3. Normalization

**Transformation**

Transformations are required when meaningful relationship between input and target features is not observable. As an example, in our case study, the cross-plot of RHOB versus the target variable (NMR-derived permeability (K_{NMR})) before and after applying logarithmic transformation on
Image generation

The preprocessed well log data, as a numerical dataset, can be used for RF, GMDH, and 1D-CNN models. But, as explained before, the developed Res 2D-CNN model needs 2D images as inputs. For this purpose, the approach proposed by Zhong et al. (2019) has been followed. Although the procedure is purely computational, the production of 2D images based on the numerical logs may be vindicated by acknowledging that each conventional well log represents a geological/physical phenomenon/property, interacting with or affected by the others. Combining different well logs in the form of an image leads to defining an artificially produced property that may not have a geological/physical interpretation (thus called an artificial 2D feature map) but contains all the logs’ information and their possible interactions.

To generate the artificial 2D feature maps, at every single depth, decimal number of each feature is first converted to a binary string. In our dataset, all values are smaller than 1000, and then, a string with 32-bit length is suitable to represent all of the dataset values. Seven well logs, corresponding to seven petrophysical variables, are converted to seven 32-bit binary strings. Output of this process is a $7 \times 32$ matrix for each depth point. Next, all of the matrices are transformed into black/white 2D images by filling the squares with the

\[ x' = \frac{(x - x_{\text{min}})}{(x_{\text{max}} - x_{\text{min}})} \]  

(5)
value “1” by color number 255, and the squares with the value “0” by color number 0. These images are stacked, and then, they are ready to feed into the training process of the Res 2D-CNN model. The schematic diagram of this data transformation process is summarized in Fig. 9. Obviously, each one of the produced artificial 2D feature maps corresponds to the permeability value of a certain depth point.

### Stage 2

At the end of stage 1, two types of datasets are generated: (1) numerical well log data for training the RF, GMDH, and 1D-CNN models, and (2) 2D feature maps for training the Res 2D-CNN model. For both datasets, the training set is split into three groups of samples: (1) the training samples (70% of the whole training dataset), (2) the validation samples (15% of the whole training dataset), and (3) the

---

**Table 1** Summary statistics of the dataset (before normalization)

| Statistics | SGR | RHOB | LLS | LLD | PEF | CGR | NPHI | log (\(K_{NMR}\)) |
|------------|-----|------|-----|-----|-----|-----|------|------------------|
| Mean       | 32.170101 | 2.511958 | 1.197345 | 27.060725 | 4.775131 | 15.519107 | 0.118727 | −0.771730 |
| Std        | 8.502929 | 0.098168 | 0.356510 | 31.003053 | 0.242283 | 4.488663 | 0.064807 | 1.427728 |
| Min        | 14.433237 | 2.158199 | 0.090633 | 0.659990 | 2.671211 | 5.632280 | −0.000867 | −5.000000 |
| Max        | 105.88634 | 2.774054 | 3.230126 | 236.48841 | 5.800828 | 54.628308 | 0.426417 | 1.912711 |

**Table 2** Summary statistics of the dataset (after normalization)

| Statistics | SGR\_NORM | RHOB\_NORM | LLS\_NORM | LLD\_NORM | PEF\_NORM | CGR\_NORM | NPHI\_NORM | log (\(K_{NMR}\)\_NORM) |
|------------|-----------|------------|------------|------------|-----------|-----------|------------|------------------|
| Mean       | 0.227081  | 0.554408   | 0.008605   | 0.019814   | 0.644429  | 0.192263  | 0.226629   | 0.479162 |
| Std        | 0.089154  | 0.109319   | 0.002655   | 0.022908   | 0.074211  | 0.080182  | 0.116079   | 0.161795 |
| Min        | 0.041109  | 0.160467   | 0.000362   | 0.000306   | 0.000003  | 0.015651  | 0.012417   | 0.000001 |
| Max        | 1.000000  | 0.846274   | 0.023745   | 0.174561   | 0.958597  | 0.890884  | 0.777749   | 0.783372 |

---

**Fig. 7** The cross-plot of input feature (RHOB) and target variable (\(K_{NMR}\)). (a) no transformation is applied to \(K_{NMR}\); (b) logarithmic transformation is applied to \(K_{NMR}\).

**Fig. 8** Statistical characteristics of the dataset. Feature correlation heat map shows correlation coefficients between every two features.
testing samples (15% of the whole training dataset). The first group is fed into the network in the training process, and the second one is utilized to determine the generalization of the network. Then, the network is adjusted according to the errors that are measured by subtracting the outputs from the targets of the training samples. Also, the validation samples provide an independent measurement of the network performance during and after the training process.

Each ML/DL model has its own specific hyper-parameters significantly influencing the performance of the employed method. It is not always straightforward to choose the correct value for these hyper-parameters. Therefore, in this study, GA algorithm is utilized for hyper-parameter tuning purpose by using the testing samples, i.e., image data for Res 2D-CNN model and numerical well log data for the other three models. Afterward, ML/DL models are trained with related training and validation samples, and in the final step, the model performance is evaluated using the blind dataset.

The correlation coefficient ($R$), mean square error (MSE), and root-mean square error (RMSE) are used to evaluate the performance of the trained networks. $R$ is defined as follows:

$$ R = \frac{\sum_{i=1}^{m} (y_i - \bar{y}_i)(\hat{y}_i - \bar{\hat{y}}_i)}{\sqrt{\sum_{i=1}^{m} (y_i - \bar{y}_i)^2 \sum_{i=1}^{m} (\hat{y}_i - \bar{\hat{y}}_i)^2}} $$

(6)

where $y_i$ is the true NMR-derived permeability, $\hat{y}_i$ is the predicted permeability by a ML/DL model, $\bar{y}_i$ is the average of NMR-derived permeability values. As mentioned earlier, a logarithmic transformation is used for permeability. The range of $R$ values is between 0 and 1. $R = 1$ indicates that regression predictions perfectly fit the actual data. MSE is the average of the square of the differences between the predicted and true permeability values:

$$ \text{MSE} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 $$

(7)

MSE is always larger than or equal to 0. An MSE of 0 means that the model predicts permeability with perfect accuracy, which is practically impossible. Finally, RMSEA is defined as:

$$ \text{RMSEA} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2} $$

(8)

**Result**

In this section, the final result of each model on the train, test, and blind datasets is presented. The whole dataset is divided into two parts: Wells $w_a$ and $w_b$ are used as training dataset, and well $w_c$ is used as the blind dataset. Three of the methods (RF, GMDH, and 1D-CNN) are trained by numerical well logs, and the fourth one (Res 2D-CNN) is trained by artificial 2D feature maps. The hyper-parameters of all four employed ML and DL models are tuned using genetic algorithm; the tuned values are reported in Tables 3, 4, 5 and 6. In addition, the performance of RF, GMDH, 1D-CNN, and Res 2D-CNN models on the blind dataset are shown in
Figs. 10, 11, 12 and 13, respectively. The obtained results are discussed in the following, separately:

**RF results**

The performance of each model is evaluated by comparing the algorithm’s predicted permeability against the targeted training dataset, test dataset and blind dataset. The performance evaluation parameters of $R$, MSE, and RMSE are obtained for each evaluation to represent the quality of the prediction. For the training dataset, the values of $R$, MSE, and RMSE are 0.9804, 0.0016, and 0.04009, respectively, while for testing dataset, $R$, MSE, and RMSE are 0.9201, 0.0059, and 0.0773. So while RF can learn to predict the permeability based on conventional logs with a relatively small error, its performance on the test dataset is not as accurate. Figure 10 shows the final results of the trained RF network tested on the blind dataset (blind well). Figure 10a shows the blind well permeability profile. The black line indicates the ML predicted permeability, and the red circles show NMR-derived permeability values. In Fig. 10b, the cross-plot shows the estimated versus the original values of permeability with $R = 0.9053$. The correlation coefficient for the blind well is even smaller than the predictions made on the test and training data sets. The errors between the target values and the predicted outputs shown in Fig. 10c reveal a larger MSE of 0.0111 and RMSE of 0.1053 for the blind dataset as well. The errors are following a log-normal distribution in Fig. 10d. The relatively poor performance of RF on the blind dataset (e.g., underestimation of high permeability values) may be due to over-fitting which is one of the shortcomings of RF methods.

**GMDH network results**

The GMDH network in comparison with RF method results in slightly better predictions on the training and the test datasets. However, for blind dataset, its performance is almost the same as RF. Due to a deeper structure, GMDH model does not have the over-fitting problem.

**1D-CNN results**

The performance evaluation parameters ($R$, MSE and RMSE) of 1D-CNN are between those of RF and GMDH models for training and testing stages; also, its performance on the blind dataset is better than the other two methods. This performance is because of efficient feature extraction.

---

**Table 3** RF regressor parameters

| The number of trees in the forest (n_estimators) | The maximum depth of the tree (max_depth) | The minimum number of samples required to split an internal node (min_sample_split) | The function to measure quality of a split (criterion) |
|-------------------------------------------------|------------------------------------------|---------------------------------------------------------------------------------|------------------------------------------------------|
| 247                                             | 25                                       | 10                                                                              | mse                                                  |

Table 4  GMDH network parameters

| Reference function (ref_function_types) | Criterion for selecting best neurons (criterion_type) | The threshold to stop train (stop_train_epsilon_condition) | Regularization value used in neuron fit by ridge regression | Number of parallel processes to model (n_jobs) |
|----------------------------------------|------------------------------------------------------|--------------------------------------------------------|-------------------------------------------------------------|-----------------------------------------------|
| Quadratic (full polynomial of the 2-nd degree) | Validate                                              | 0.0001                                                  | 0.5                                                          | max                                           |

Table 5  1D-CNN parameters

| Optimization function (optimizer) | Learning rate | Activation functions | Number of samples for gradient update (batch_size) | Number of epochs to the train the model | Number of hidden layers (Conv 1D) | Number of neurons (Conv 1D) | Number of neurons (Dense) |
|-----------------------------------|---------------|----------------------|---------------------------------------------------|----------------------------------------|--------------------------------|---------------------------|--------------------------|
| RMSprop                           | 0.00133       | ReLU                 | 16                                                | 50                                     | 1                              | 71                        | 90                       |

Table 6  Res 2D-CNN parameters

| Optimization function (optimizer) | Learning rate | Activation functions | Number of samples for gradient update (batch_size) | Number of epochs to the train the model | Number of hidden layers (Conv 2D) | Number of neurons (Conv 2D) | Number of neurons (Dense) |
|-----------------------------------|---------------|----------------------|---------------------------------------------------|----------------------------------------|--------------------------------|---------------------------|--------------------------|
| Adam                              | 0.00027       | ReLU                 | 8                                                 | 500                                    | 9                              | 128 & 32                  | 90                       |
Fig. 10 NMR-derived permeability prediction performance of RF regressor on blind dataset.  

- **a** True (Targets) versus predicted (Outputs) permeability values.  
- **b** Cross-plot showing the true versus estimated permeability values.  
- **c** Relative deviation of RF regressor for permeability prediction versus relevant true permeability data samples.  
- **d** Histogram of the error between the true and estimated permeability values.

Fig. 11 NMR-derived permeability prediction performance of GMDH network on blind dataset.  

- **a** True (Targets) versus predicted (Outputs) permeability values.  
- **b** Cross-plot showing the true versus estimated permeability values.  
- **c** Relative deviation of GMDH for permeability prediction versus relevant true permeability data samples.  
- **d** Histogram of the error between the true and estimated permeability values.

Springer
Fig. 12 NMR-derived permeability prediction performance of 1D-CNN on blind dataset. a True (Targets) versus predicted (Outputs) permeability values. b Cross-plot showing the true versus estimated permeability values. c Relative deviation of 1D-CNN for permeability prediction versus relevant true permeability data samples. d Histogram of the error between the true and estimated permeability values

Fig. 13 NMR-derived permeability prediction performance of Res 2D-CNN on blind dataset. a True (Targets) versus predicted (Outputs) permeability values. b Cross-plot showing the true versus estimated permeability values. c Relative deviation of Res 2D-CNN for permeability prediction versus the true permeability data samples. d Histogram of the error between the true and estimated permeability values
of 1D-CNN and its advanced deep learning structure. Also, the obtained result reveals that 1D-CNN does not suffer from over-fitting problem.

**Res 2D-CNN results**

The performance of Res 2D-CNN models on the blind data-set is shown in Fig. 13. As it can be clearly seen, the Res 2D-CNN model provides a strongly accurate permeability prediction with a very high $R$ value and very low MSE and RMSE values.

**Comparison of the models**

The applied ML/DL models on training and testing datasets are compared in Fig. 14 in terms of the performance evaluation parameters. In addition, the summary of the performance evaluation parameters from the described approaches mentioned above is reported in Fig. 15. The results indicate that the nonlinear ML/DL techniques are efficient in the prediction of NMR-derived permeability. It is clear that among the methods with numerical input, 1D-CNN has a better performance with higher $R$ value than GMDH, and RF methods, and lower MSE and RMSE. Res 2D-CNN achieves the best performance (closest $R$-value to 1 and MSE and RMSE values closest to 0) in permeability prediction with artificial 2D feature maps as input features. Therefore, for this case, deep learning methods (1D-CNN and Res 2D-CNN) are preferred over the other applied methods (RF and GMDH).

**Discussion**

Examination of the employed ML/DL algorithms shows that 1D-CNN and Residual 2D-CNN methods perform better in predicting NMR-derived permeability from conventional wireline logs. In general, CNN models benefit from effective feature extraction without loss of information and high capability in pattern recognition and image processing. In the case of Residual 2D-CNN, by converting the well logs to images, in the format of artificial 2D feature maps, the input variables (well log data) can communicate during convolution calculation stage. Such a communication does not happen in the traditional ML algorithms as the neurons in each hidden layer are separated. Beside the inherent superiority of 2D-CNN method owing to its deeper structure and pattern recognition potential, the applied modifications have improved its performance and training speed. The residual and the deeper bottleneck architectures increase the depth of the Res 2D-CNN model and decrease the number of trainable parameters. Considering all the output results, Res 2D-CNN has shown satisfactory stability and capability of generalization.

Compared to the shallow networks, DL methods can extract more complex patterns in the input data, but the larger number of parameters involved in deeper networks.
necessitates a larger number of data samples for training the algorithm and optimal determination of these parameters. This requirement may cause difficulty in the course of dataset selection and the preparation process. For example, similar patterns in the input data should be avoided as they may orient the model toward specific trends. Besides, the training dataset is supposed to have almost all possible patterns in approximately equal proportions to maintain the comprehensiveness of the model. Establishing this balance in a larger number of samples in the training dataset is more difficult. The issue of limited data availability in this work has been addressed by selecting the best sample points with the highest variability in measured features and integrating the deeper bottleneck architecture into the DL structure to reduce the required number of trainable parameters.

While the main objectives of this research are well-achieved, as the future work, the following concepts may be investigated for possible improvements:

- **Generating physically understandable feature images:** Although DL models could recognize the artificial 2D image patterns, it might be beneficial to generate input images that are visually understandable by expert engineers so that the machine’s performance can be judged by professionals’ intuition.
- **Using a multiple-input deep neural network:** Multiple-input deep neural networks are able to handle and analyze two types of data as inputs so geologic 2D feature images and numerical well log dataset can be used simultaneously.

### Conclusion

A comprehensive methodology based on ML/DL algorithms is presented to use the widely available conventional wireline logs for estimation of the reservoir permeability. The presented algorithms help to improve the accuracy and the performance of the permeability estimation process. The permeability values are derived from NMR logs and depending on the employed algorithm, the conventional logs or the artificially produced feature maps are used to develop the related predictive models. Once trained and tested, the predictive model can be used for the other wells in the field with no NMR log or core measurements.

Based on the performance of the trained models, shallow and deep learning techniques have proven to be capable of predicting NMR-derived permeability. Between RF, GMDH, 1D-CNN, and Res 2D-CNN, the latter showed the best performance in correlating NMR-derived permeability with the synthetized feature maps. Converting the numerical well log data to artificial 2D feature maps proved to be an efficient method to be employed in CNN models for predicting permeability. Also, adding residual
and deeper bottleneck structures helped 2D-CNN model to achieve better permeability predictions at higher speeds.

**Funding** No funding is applied in this research.

**Declarations**

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

**References**

Ahmadi MA, Chen Z (2019) Comparison of machine learning methods for estimating permeability and porosity of oil reservoirs via petrophysical logs. Petroleum 5(3):271–284. https://doi.org/10.1016/j.petlm.2018.06.012

Al Khalifa H, Glover PWJ, Lorinzi C (2020) Permeability prediction and diagenesis in tight carbonates using machine learning techniques. Mar Pet Geol 112:104096. https://doi.org/10.1016/j.marpetgeo.2019.104096

Al-Anazi AF, Gates ID (2012) Support vector regression to predict porosity and permeability: effect of sample size. Comput Geosci 39:64–76. https://doi.org/10.1016/j.cageo.2011.06.011

Alqahtani N, Armstrong RT, Mostaghimi P (2018) Deep learning convolutional neural networks to predict porous media properties. In: SPE Asia Pacific oil and gas conference and exhibition. OnePetro. https://doi.org/10.2118/191906-MS

Amaefule JO, Altunbay M, Tiab D, Kersey DG, Keelan, DK, (1993) Enhanced reservoir description: using core and log data to identify hydraulic (flow) units and predict permeability in uncored intervals/wells. In: SPE annual technical conference and exhibition. OnePetro. https://doi.org/10.2118/26436-MS

Amanifard N, Nariman-Zadeh N, Farahani MH, Khalkhali A (2008) Modelling of multiple short-length-scale stal cells in an axial compressor using evolved GMDH neural networks. Energy Convers Manage 49(10):2588–2594. https://doi.org/10.1016/j.enconman.2008.05.025

Amiri R, Haddadpour H, Emami Niri M (2021) Investigating the capability of data-driven proxy models as solution for reservoir geological uncertainty quantification. J Petrol Sci Eng 205:108860. https://doi.org/10.1016/j.petrol.2021.108860

Archie GE (1942) The electrical resistivity log as an aid in determining some reservoir characteristics. Trans AIME 146(01):54–62. https://doi.org/10.2118/942054-PA

Baoouche R, Alfa T, Baddari K (2017) Intelligent methods for predicting nuclear magnetic resonance of porosity and permeability by conventional well-logs: a case study of Saharan field. Arab J Geosci 10(24):545. https://doi.org/10.1007/s12517-017-3344-y

Biot MA, Willis DG (1957) The elastic coefficients of the theory of consolidation, https://doi.org/10.1115/1.4011606

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Briones V, Carrasquilla A (2013) Simulating porosity and permeability of NMR log in carbonate reservoirs of campos basin southeast Brazil using conventional logs and artificial intelligence techniques. In: SEG technical program expanded abstracts 2013. Society of Exploration Geophysicists, pp 2522–2527. https://doi.org/10.1190/segam2013-0409.1

Carman PC (1937) Fluid flow through granular beds. Trans Inst Chem Eng 15:150–166

Chawshin K, Berg CF, Varagadolo D, Lopez O (2021) A deep-learning approach for lithological classification using 3D whole core CT-scan images. In: SPWLA 62nd annual logging symposium. OnePetro. https://doi.org/10.30632/SPWLA-2021-0029

Cheng G, Zhou P, Han J (2016) Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans Geosci Remote Sens 54(12):7405–7415. https://doi.org/10.1109/TGRS.2016.2601622

Ciregan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 3642–3649. https://doi.org/10.1109/CVPR.2012.6248110

Ciresan DC, Meier U, Masci J, Gambardella LM, Schmidhuber J (2011) Flexible, high performance convolutional neural networks for image classification. In: Twenty-second international joint conference on artificial intelligence. https://doi.org/10.5599/978-1-57735-516-8%2FIJCAI11-210

Crain ER (1986) Log analysis handbook. PennWell Books, Tulsa, USA

Cui Z, Yang J, Qiao Y (2016) Brain MRI segmentation with patch-based CNN approach. In: 2016 35th Chinese control conference (CCC). IEEE, pp 7026–7031. https://doi.org/10.1109/ChiCC.2016.7554465

Ebetjej I, Bonakdari H, Zaji AH, Azimi H, Khoshbin F (2015) GMDH-type neural network approach for modeling the discharge coefficient of rectangular sharp-crested side weirs. Eng Sci Technol Int J 18(4):746–757. https://doi.org/10.1016/j.jestch.2015.04.012

Farlow SJ (2020) Self-organizing methods in modeling: GMDH type algorithms. CRC Press, Florida

Ghamisi P, Chen Y, Zha XX (2016) A self-improving convolution neural network for the classification of hyperspectral data. IEEE Geosci Remote Sens Lett 13(10):1537–1541. https://doi.org/10.1109/LGRS.2016.2595108

Gohari MSJ, Emami Niri M, Ghasi-Freez J (2021) Improving permeability estimation of carbonate rocks using extracted pore network parameters: a gas field case study. Acta Geophys 69(2):509–527. https://doi.org/10.1007/s11600-021-00563-z

Haghshenas Y, Emami Niri M, Amini S, Amiri R (2020) Developing grid-based smart proxy model to evaluate various water flooding injection scenarios. Pet Sci Technol 38(17):870–881. https://doi.org/10.1080/10916466.2020.1796703

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778. https://doi.org/10.1109/CVPR.2016.90

Horner DR (1951) Pressure build-up in wells. In: 3rd world petroleum congress. OnePetro.

Hwang HS (2006) Fuzzy GMDH-type neural network model and its application to forecasting of mobile communication. Comput Ind Eng 50(4):450–457. https://doi.org/10.1016/j.cie.2005.08.005

Ikhlchi AK, Rezaee M, Moallemi SA (2006) A fuzzy logic approach for estimation of permeability and rock type from conventional well log data: an example from the Kangan reservoir in the Iran
