Photoelectric factor prediction using automated learning and uncertainty quantification

Khalid Alsamadony1 · Ahmed Farid Ibrahim1,2 · Salaheldin Elkatatny1,2 · Abdulazeez Abdulraheem1,2

Received: 24 June 2022 / Accepted: 24 July 2023 / Published online: 14 August 2023
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

Abstract
The photoelectric factor (PEF) is an important well-logging tool to distinguish between different types of reservoir rocks because PEF measurement is sensitive to elements with high atomic numbers. Furthermore, the ratio of rock minerals could be determined by combining PEF log with other well logs. However, PEF logs could be missing in some cases such as in old well logs and wells drilled with barite-based mud. Therefore, developing models for estimating missing PEF logs is essential in those circumstances. In this work, we developed various machine learning models to predict PEF values using the following well logs as inputs: bulk density (RHOB), neutron porosity (NPHI), gamma ray (GR), compressional and shear velocity. The predictions of PEF values using adaptive-network-fuzzy inference system (ANFIS) and artificial neural network (ANN) models have errors of about 16% and 14% average absolute percentage error (AAPE) in the testing dataset, respectively. Thus, a different approach was proposed that is based on the concept of automated machine learning. It works by automatically searching for the optimal model type and optimizes its hyperparameters for the dataset under investigation. This approach selected a Gaussian process regression (GPR) model for the accurate estimation of PEF values. The developed GPR model decreases the AAPE of the predicted PEF values in the testing dataset to about 10% AAPE. This error could be further decreased to about 2% by modeling the potential noise in the measurements using the GPR model.

Keywords Machine learning algorithms · Well logging · Photoelectric factor · Automated learning · Uncertainty quantification · Gaussian process regression · Fuzzy logic · Artificial neural network (ANN)

1 Introduction
The photoelectric factor (PEF) log is a robust tool for identifying lithology. This is because PEF measurement is mainly affected by rocks with high atomic numbers while being insensitive to variations in porosity, fluid type, and saturation. Therefore, PEF log capability to identify the lithology is not diminished in gas-bearing zones, where it may be challenging to distinguish between lithologies using only formation density and neutron logs. One more advantage of the PEF log is that it can distinguish between clean (i.e., low shale content) sandstone and clean limestone, while the gamma ray log cannot clearly demystify the difference between them. PEF \( (P_e) \) is proportional to the atomic number as shown in the next empirical relationship (Eq. 1), where \( z \) is the atomic number [1–3].

\[
P_e = \left( \frac{Z}{10} \right)^{3.6}
\] (1)

PEF log may be missing in some cases because of instrument failure, old logs missing this tool, and economic constraints. Moreover, PEF log could be mismeasured and misinterpreted in wells drilled with barite and in wells that have heavy elements even in small quantity. Remeasuring PEF log is often impractical because of well casing or...
limited budget. Therefore, previous studies utilized machine learning techniques to predict PEF log due to its importance in improving formation evaluation [4, 5].

1.1 PEF logs as inputs for machine learning models

PEF log is an important input parameter for many machine learning models to classify lithology and predict other valuable parameters such as density, porosity, water saturation, sonic and geochemical logs. PEF was used for training classifiers (k-nearest neighbors (KNN), support vector machine (SVM), random forest (RF), decision tree (DT), XGBoost, LightGBM, artificial neural network (ANN) and rough set theory rules) to predict lithofacies and electrofacies [6–8]. Moreover, it is an input for several models (ANN, SVM, Super Learner, XGBoost, CatBoost, LightGBM and AdaBoost) to estimate water saturation [9, 10].

PEF log was utilized to generate sonic logs (compressional and shear velocity) using ANN, recurrent neural networks (RNNs), RF, ensemble method (combination of several models) and LightGBM, where data preprocessing and feature engineering had a significant impact on the accuracy [11, 12]. Another study proposed fast fuzzy modeling method (FFMM) to predict sonic and density logs using PEF log as an input, where the proposed method had similar accuracy to ANN and fuzzy logic (FL), but with less computational and storage cost [13].

PEF log is an important input in predicting geochemical logs (Al, Ca, Fe, Na, and Si) using the following methods: SVM, ANN, RF, AdaBoost, and XGBoost [14]. Moreover, it is one of the inputs for long short-term memory (LSTM) model to output neutron porosity and resistivity logs [11].

1.2 Prediction of PEF logs

Tatsipie and Sheng trained LSTM model to predict PEF log using the following inputs: gamma ray, bulk density, depth, and well coordinates. PEF log was predicted with high errors relative to other predicted logs in this study, which justified that the trained LSTM model could not capture the complex relation between PEF, gamma ray, and bulk density [11].

Rostamian et al. predicted PEF, bulk density, and neutron porosity logs using several machine learning methods: multilinear regression, deep neural network (DNN), DT, KNN, RF, and XGBoost. These models receive corrected gamma ray (CGR), caliper (CALI), deep lateral log (LLD), shallow lateral log (LLS), micro-spherical focused log (MSFL), and sonic transit time, where all the well logs were normalized (Z-score transformation). XGBoost, RF, and KNN generated the best results, in addition, their hyperparameters and selected inputs were optimized by a genetic algorithm. This optimization slightly improved the accuracy and allowed reducing the required number of inputs, while predicting the missing logs with acceptable accuracy [15].

1.3 Novelty

The objective of this study is to develop machine learning models for estimating PEF values for a heterogeneous dataset that was collected from a field in the Middle East. In our dataset, the PEF values range from (2.1) to (13.4), while they range from from (1) to (7.8), (2.5) to (5.5), and (1) to (4.9) in the datasets of the previous studies [11, 15, 16]. Moreover, the previous research papers rely only on the manual selection of machine learning models and do not quantify uncertainty in predictions of well logs [11, 15, 16].

The previous studies have predicted PEF values with relatively high errors. Rostamian et al. [15] predicted PEF values that have about 10 times more error than the prediction of neutron porosity (NPHI) according to the AAPE. Tatsipie and Sheng concluded that the predicted PEF log has the worst accuracy compared to the other predicted well logs [11]. Our results, despite being based on a different dataset, confirm that predicting PEF logs accurately could be challenging; hence, this seems related to the prediction of PEF logs regardless of the dataset. For this reason, a method based on automated machine learning is proposed for better accuracy and efficient implementation of machine learning tools by removing the time-consuming manual steps. Additionally, we utilize uncertainty quantification to enhance the reliability of machine learning methods by modeling the potential noise that is usually in the measurements of actual cases at varying levels.

2 Methodology

In this work, we developed three machine learning models to predict PEF log, which are adaptive neuro-fuzzy inference system (ANFIS), artificial neural network (ANN) and Gaussian process regression (GPR). The inputs of these models are 5 logs: bulk density (RHOB), neutron porosity (NPHI), gamma ray (GR), compressional and shear velocity. The accuracy of the trained models is evaluated using the common metrics: average (mean) absolute percentage error (AAPE) (Eq. 2) and Pearson correlation coefficient (R) (Eq. 3) between the predicted and the actual PEF values. For the uncertainty quantification (UQ) model, AAPE refers to the percentage of the predicted PEF values that are outside the confidence interval (CI).
\[ AAPE = \frac{100}{n} \times \sum_{i=1}^{n} \left| \frac{\text{actual PEF}_i - \text{predicted PEF}_i}{\text{actual PEF}_i} \right| \]  

\[ R = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \]

where \( n \) is the number of datapoints, \( x_i \) is the actual PEF value, \( \bar{x} \) is the average of the actual PEF values, \( y_i \) is the predicted PEF value, and \( \bar{y} \) is the average of the predicted PEF values.

### 2.1 Data description

The dataset consists of 9399 datapoints from a field in the Middle East such that 4922 datapoints were used for training the models, 895 datapoints for the initial testing, and 2603 datapoints for the final validation of the models.

Statistical analysis of the dataset is presented in Table 1, where the heterogeneity of the dataset can be observed. Specifically, the minimum and the maximum values of GR (3, 381) and NPHI (0, 0.44) indicate that the dataset contains different types of zones with distinct properties. The kurtosis values of GR (9.75) and NPHI (4.67) show that they have many extreme values. R-values depict that RHOB is the most linearly correlated with PEF (−0.37), that is followed by compressional velocity (0.18). R value measures the linear correlation between two variables, where value of one indicates a perfect proportional linear correlation, value of negative one indicates a perfect inversely proportional linear correlation, and zero indicates that there is no linear correlation between the two variables.

Before the training process, the inputs and the PEF values were normalized to be between zero and one using their minimum and maximum values (Eq.4). However, the predicted PEF values were restored to the original scale (denormalized), so we can evaluate the accuracy of the models using AAPE and R. The dataset was checked for erroneous values such as zeros, negatives, and very extreme unreasonable values. Moreover, we attempted to remove the outliers using common statistical rules such as 1.5 IQR and 3 sigma (standard deviation). However, the accuracy of ANN model did not improve, but it slightly decreased. This probably due to the heterogeneous nature of our dataset, where these extreme values represent some layers. Thus, removing them hinders the predictive capability of the artificial intelligence models. Additionally, numerous feature engineering and data transformation methods were applied, but they were not effective at reducing the error, as they only reduced AAPE by about 1% in the best case. As a consequence, a different approach was implemented utilizing automated learning and uncertainty quantification.

Table 1 Statistical description of the dataset

| Statistical analysis | RHOB  | NPHI  | GR    | Compressional | Shear | PEF   |
|----------------------|-------|-------|-------|---------------|-------|-------|
| Min                  | 2.18  | −0.01 | 3.17  | 40.31         | 60.82 | 2.14  |
| Max                  | 3.00  | 0.44  | 381.90| 83.04         | 147.35| 13.39 |
| Mean                 | 2.71  | 0.08  | 55.80 | 53.34         | 93.50 | 5.77  |
| Mode                 | 2.85  | 0.02  | 19.97 | 43.69         | 82.84 | 4.21  |
| Median               | 2.59  | 0.22  | 192.53| 61.67         | 104.08| 7.76  |
| Standard deviation   | 0.17  | 0.06  | 49.14 | 8.86          | 11.68 | 2.05  |
| Coef. of variation   | 0.06  | 0.80  | 0.88  | 0.17          | 0.12  | 0.36  |
| Skewness             | −0.43 | 1.21  | 2.29  | 0.75          | 1.23  | 0.83  |
| Kurtosis             | 1.98  | 4.67  | 9.75  | 2.49          | 4.46  | 3.21  |
| R (PEF)              | −0.37 | 0.10  | 0.03  | 0.18          | 0.13  | 1.00  |

### 2.2 Models description

In this section, we briefly introduce the three machine learning methods that have been used to generate the results in our work, which are ANFIS, ANN and GPR. Moreover, we cite some references that explain the theory of these methods in detail and their different applications.

Adaptive-network-based fuzzy inference system (ANFIS) can be used as a supervised learning method since it is capable of modeling nonlinear functions. Its prediction (inference) is based on Fuzzy rules (if–then rules) to map inputs of the system to outputs; thus, it can resemble human knowledge and reasoning. These rules (if–then) are implemented using membership functions. This is the main difference between ANFIS and ANNs [17]. Several studies used ANFIS to predict rock properties and well logs [18–20].

ANNs have been widely used for classification and regression tasks. The basic component of ANN is the neuron, which forms the hidden layers. The hidden layers map the input layer to the output layer, and each hidden
layer is followed by a transfer function to learn nonlinear features. The neurons are associated with weights that are updated during the training process according to the backpropagated errors and the training algorithm, e.g., gradient descent [21]. Several studies used ANN to predict rock properties and well logs [18–20, 22, 23].

The general concept of GPR assumes that datapoints with similar input values are expected to have similar output values. This concept is applied using covariance (kernel) functions that model these relations of similarity [24]. Due to this, kriging that is the well-known method used in geo-statistics to interpolate the missing values based on the established spatial covariance or variogram is considered a GPR method. GPR is less commonly used than ANN to predict rock properties and well logs [23, 25, 26].

3 Results

3.1 ANFIS results

To develop the ANFIS model, we used subtractive clustering (Sugeno fuzzy) algorithm that searches for data clusters, from which membership functions and rules are constructed [27, 28]. The main hyperparameter of the subtractive clustering method is the radius of the clusters. During the training phase, the ANFIS model was optimized by varying the radius values to identify the optimum value that results in the highest R and lowest AAPE for the testing dataset. The optimized ANFIS model has a radius of 0.139 and was trained for 200 epochs. The accuracy of the developed ANFIS model in predicting the PEF values is 19.19% AAPE in the training data and 16.04% AAPE in the testing data. Moreover, the correlation coefficient (R) between the predicted and the actual PEF values is 0.715 in the training data and 0.77 in the testing data. Figure 1 presents the cross-plots between the predicted and the actual PEF values. We observe that the ANFIS model underfits the training data because it has higher errors in predicting the PEF values in the training data than in the testing data.

3.2 ANN results

The accuracy of ANN models is affected by their hyperparameters such as the number of neurons and layers, the transfer (activation) function and the training algorithm that minimizes the loss function by updating the weights of the network. The studied search space of the ANN hyperparameters is the following: the number of hidden layers (1–3), the number of neurons (5–35), the transfer functions (hyperbolic tangent sigmoid, log sigmoid, hard limit, linear, radial basis, and soft max), and the training algorithms (Levenberg–Marquardt, BFGS quasi-Newton, Bayesian, and conjugate gradients).

The developed ANN model was optimized by selecting the optimum combination of those hyperparameters to generate the best accuracy for the testing dataset. The best hyperparameters are 2 hidden layers with 16 and 20 neurons, respectively, log sigmoid transfer function, and Bayesian regularization backpropagation. The accuracy of the ANN model in predicting the PEF values is (AAPE = 13.32% and R = 0.856) in the training data, and (AAPE = 14.09% and R = 0.815) in the testing data. Figure 2 shows the cross-plots between the actual PEF and the predicted PEF values. This shallow ANN model slightly underfits the training data. Although the accuracy of the ANN model is better than the ANFIS model, it is still not perfectly accurate.

We developed another ANN model that has only one hidden layer, so it can be converted to an equation (white box model). The best hyperparameters of this ANN model are 30 neurons, log sigmoid transfer function, and Bayesian regularization backpropagation. The accuracy of the shallow ANN model in predicting the PEF values is (AAPE = 16.22% and R = 0.801) in the training data, and (AAPE = 14.44% and R = 0.809) in the testing data. This shallow ANN model slightly underfits the training data. The code to use the equation of the ANN model is available in the supplementary material.

3.3 GPR results

Automated machine learning searches for the optimal algorithm (model) for the dataset of interest, by applying different types of methods, while optimizing their hyperparameters for the best results [29]. Thus, it increases the efficiency of applying artificial intelligence tools through eliminating the time wasted on manual steps.

The automated learning code compares different models including multi-linear regression, Gaussian kernel regression, decision tree, GPR, ensemble learning (Least-squares gradient boosting and Bootstrap aggregation (random forest)), ANNs and support vector machine (SVM). The code utilizes Bayesian optimization to select the optimal hyperparameters for the models and find the most accurate model for the prediction task [30]. The results show that the best model is a GPR model. One additional unique advantage of GPR is that it can incorporate uncertainty, hence accounting for the potential noise in the PEF measurements. This advantage is extremely critical in zones where the PEF values cannot be predicted accurately because it reduces the probability of misinterpreting the predicted PEF log by geoscientists and engineers.

The optimum GPR model has these hyperparameters: sigma = 0.032, kernel scale = 0.602, constant basis
function, and exponential kernel function. It is evident that the GPR model outperforms the ANFIS and the ANN models as summarized in Table 2. The accuracy of the GPR model in predicting the PEF values is (AAPE = 0.01\% and $R = 0.999$) in the training data, and (AAPE = 10.32\% and $R = 0.889$) in the testing data. Figure 3 is the cross-plots between the actual and the predicted PEF values. Furthermore, the error of the GPR model in the testing dataset could be further decreased to about (4.6–1.9\%) using (95–99\%) confidence interval to model the noise of the dataset.

We observe that the GPR model overfits the training data, which is common to some GPR methods [31]. However, this probably does not reduce its accuracy as can be observed by comparing the GPR model accuracy against the accuracy of the ANN and the ANFIS models using the testing data and the final validation data (next section). In

Table 2 Compares the accuracy of the developed models

| Method | $R$ (train) | AAPE (train), \% | $R$ (test) | AAPE (test), \% |
|--------|-------------|-----------------|------------|-----------------|
| ANFIS  | 0.715       | 19.19           | 0.77       | 16.04           |
| ANN    | 0.856       | 13.32           | 0.815      | 14.09           |
| GPR    | 0.999       | 0.01            | 0.889      | 10.23           |

![Fig. 1](image1.png) Cross-plots between the actual and the predicted PEF values from the ANFIS model

![Fig. 2](image2.png) Cross-plots between the actual and the predicted PEF values from the ANN model
general, weak learners tend to underfit the data (e.g., in our case the ANFIS model), while strong learners tend to overfit the data.

4 Validation

In this section, the best model (GPR + UQ) is further tested on larger unseen dataset for final validation. This dataset consists of 2603 datapoints. The cross-plot between the predicted and the actual PEF values in the validation data is shown in Fig. 4. The GPR predicts the PEF values with the highest accuracy (AAPE = 9.5% and $R = 0.894$) in the validation data (Table 3), where estimating the uncertainty using 95% confidence (prediction) interval decreases the error to (4.8%). The importance of uncertainty quantification is noticeable in zones where the actual PEF values deviate from the predictions, e.g., depth index from 1080 to 1090 (Fig. 5). The uncertainty model could also consider the potential noise in the inputs that can be observed from the depth index (1035–1040) in Fig. 5, where the PEF values are similar but have different confidence estimations because of having different inputs.

In Fig. 5, the UQ error of 99% confidence interval is lower since it includes more points within its prediction interval, e.g., the datapoint at depth index 1089 is outside the 90% prediction interval. The average confidence interval width of the 99% plot is the highest (5), which may limit the functionality of the PEF log in identifying lithology. Thus, despite that the error of the uncertainty could be further decreased to about 2% using a 99% confidence interval, a lower confidence interval (95% or 90%) may be selected as they generate acceptable results with a more reasonable prediction width (3.8 and 3.2, respectively). However, in other cases, a different confidence interval (lower or higher) may be selected depending on the

Table 3 Compares the accuracy of the developed models using the final validation dataset (2603 datapoints)

| Method | $R$ (validation) | AAPE (validation), % |
|--------|------------------|----------------------|
| ANFIS  | 0.761            | 15.51                |
| ANN    | 0.810            | 13.32                |
| GPR    | 0.894            | 9.55                 |

Fig. 3 Cross-plots between the actual and the predicted PEF values from the GPR model

Fig. 4 Cross-plot between the actual and the predicted PEF values from the GPR model (final validation data)
knowledge about the predicted parameter, the acceptable range of error and the width of confidence interval.

5 Discussion

The main characteristics of GPR can be summarized as multivariate normal (Gaussian) distributions (joint normal distribution), kernels (covariance functions), and posterior (conditional) distribution. The multivariate normal distribution is considered to be the prior information about dataset distribution without any training. In contrast to univariate normal distribution which is defined by only one variance assuming the mean value is zero, multivariate normal distribution requires a covariance matrix to be defined. The covariance matrix is represented by a kernel that has hyperparameters that are learned during training, i.e., the kernel updates the initially assumed multivariate normal distribution with training data. Consequently, given the updated multivariate normal distribution and inputs of a testing datapoint, the posterior predictive distribution can be defined. After neglecting the mean (without explicit basis) and the noise variance, the GPR expected output (prediction) and variance can be simplified for one test datapoint to Eqs. (5 and 6). Thus, in GPR, each prediction can be viewed as a prediction of a normal probability density function (PDF) that is defined by the expected prediction value and its variance. [24, 32–35]

\[
\text{Expected output} = K_T^T K^{-1} Y_t \\
\text{Variance of the output} = K_{xx} - K_x^T K^{-1} K_x
\]

In Eqs. (5 and 6), \( Y_t \) is a vector that has previous observations (training targets), \( K^{-1} \) is the inverted covariance matrix that represents the training inputs. \( K_x \) is a vector that has covariance values between inputs of a testing datapoint and the training data. \( K_{xx} \) is the variance from the diagonal of the covariance matrix. \( K, K_x, \) and \( K_{xx} \) are calculated from kernels. The assumptions of Gaussian (normal) prior and likelihood are essential to derive Eqs. (5 and 6). Moreover, the marginal likelihood can be used to optimize the hyperparameters.

Generally, a machine learning method that is reliable for a dataset may not be necessarily reliable for other datasets that it is common practice to develop different machine learning models to find the most suitable method. The exact reasons behind the suitability of a method for a particular dataset are usually unknown otherwise the most accurate method for a dataset would be known beforehand. In the next paragraph, we mention some reasons that may have enhanced GPR accuracy in the studied dataset.

Although GPR assumes a normal distribution, GPR has empirically demonstrated its reliability in applications that involve non-normal distributions. This can be explained from the central limit theorem perspective as it implies that the statistical assumptions of normal distributions can be valid for other distributions [36, 37]. GPR exploits the learned covariance relations (kernels) from the training dataset to infer unseen data points. For instance, the
datapoint at depth 1027 in figure has more similar covariance relations to the training dataset than the datapoint at depth 1055. This can be observed from the width of the confidence interval since the smaller confidence width indicates that the GPR model is more confident in this prediction. Moreover, covariance functions are potentially suited to spatially distributed data that can be inferred from their applications in geostatistics. The connection between GPR and geostatistics (kriging) is explained in other studies [38–40]. Furthermore, from ANN perspective, GPR can be viewed as a fully connected neural network that has an infinite hidden layer [41–43].

Despite the unique characteristic and advantages of GPR, its performance can be hindered in cases of high dimensional and big data due to the computational complexity of covariance matrix inversion. On a large scale, other machine learning algorithms could assist GPR models. Recommendation system (collaborative filtering) may be utilized to select one of the developed GPR models according to the field (location) and other features [44–47]. Similarly, clustering algorithms may assist in choosing the number of GPR models to be developed [48, 49]. Several GPR variants have been developed to alleviate shortcomings of GPR and improve its generalization. These variants could be investigated in future work. Some of these variants integrate GPR with neural networks through parameterizing kernels in GPR by neural network and using neural network output as prior for GPR [50, 51]. Moreover, scalable GPR methods are more suited to big data [52]. Other variants include deep GPR and deep sigma point process [53, 54].

6 Conclusion

In this work, we developed various models to predict PEF log. The results show that predicting PEF logs can be challenging because they cannot be predicted accurately for some zones by the common machine learning methods. Similar results have been reported by previous studies. To improve the accuracy of the models, we attempted several data preprocessing and transformation methods, but they were not effective at reducing the prediction error.

Therefore, automated machine learning and uncertainty quantification were proposed for robust prediction of the PEF values in our dataset. The automated learning code recommended the GPR model which outperformed the other machine learning methods in predicting the PEF values. Moreover, uncertainty quantification has allowed further reduction in error by considering the noise in the data, which is crucial for the troublesome reservoir layers. It could be used for risk assessment, hence decreasing the possibilities of misinterpreting the predicted PEF values.

The GPR model generates the most accurate PEF predictions for the dataset under investigation, but could this finding be generalized to other well logs, drilling and geomechanical parameters? More investigations may be required to confirm the reliability of the GPR model in predicting PEF logs for different wells, fields, data ranges, and geological layers. Additionally, the causes of the high error in the predicted PEF values using machine learning methods could be studied. This error may be due to the heterogeneity of the dataset, some special reservoir rocks, well (borehole) condition, heavy elements, and noise from the well-logging tools (PEF tool).

Data availability Most of the data are provided in the paper, and a detailed sample will be provided upon request.

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.

References

1. Bassiouni Z et al. (1994) Theory, measurement, and interpretation of well logs. In: Henry L. Doherty Memorial Fund of AIME, Society of Petroleum Engineers
2. Ellis DV, Singer JM (2007) Well logging for earth scientists. Springer, Berlin
3. Atlas D (1982) Well logging and interpretation techniques: The course for home study. Dresser Atlas.
4. Assoodeh M, Shadizadeh SR (2015) The prediction of photoelectric factor, formation true resistivity, and formation water saturation from petrophysical Well log data: a committee neural network approach. Energy Sour Part A: Recover Utilization Environ Eff 37:557–566. https://doi.org/10.1080/15567036.2011. 594859
5. Akinnikawe O, Lyne S, Roberts J (2018) Synthetic well log generation using machine learning techniques. In: Proceedings of the 6th Unconventional Resources Technology Conference. American Association of Petroleum Geologists, Tulsa, OK, USA
6. Amir SM, Khan MR, Pancharoensawad E, Kryvenko S (2020) Integration of petrophysical log data with computational intelligence for the development of a lithology predictor. In: Day 2 Tue, October 27, 2020. SPE
7. Merembayev T, Kurmangaliyev D, Bekbaev B, Amanbek Y (2021) A Comparison of machine learning algorithms in predicting lithofacies: case studies from Norway and Kazakhstan. Energies (Basel) 14:1896. https://doi.org/10.3390/en14071896
8. Hossain TM, Watada J, Aziz IA, Hermana M (2020) Machine learning in electrofacies classification and subsurface lithology interpretation: a rough set theory approach. Appl Sci 10:5940. https://doi.org/10.3390/app10175940
9. Miah MI, Zendehboudi S, Ahmed S (2020) Log data-driven model and feature ranking for water saturation prediction using machine learning approach. J Pet Sci Eng 194:107291. https://doi.org/10.1016/j.petrol.2020.107291

10. Hadavimoghaddam F, Ostadhassan M, Sadri MA et al (2021) Prediction of water saturation from well log data by machine learning algorithms: boosting and super learner. J Mar Sci Eng 9:666. https://doi.org/10.3390/jmse9060666

11. Tatsipie NRK, Sheng JJ (2021) Generating pseudo well logs for a part of the upper Bakken using recurrent neural networks. J Pet Sci Eng 200:108253. https://doi.org/10.1016/j.petrol.2020.108253

12. Yu Y, Xu C, Misra S et al (2021) Synthetic sonic log generation with machine learning: a contest summary from five methods. Petrophysics SPWLA J Formation Eval Reserv Descr 62:393–406. https://doi.org/10.30632/PJV/62N4-2021a4

13. Bahrpeyma F, Golchin B, Cranganu C (2013) Fast fuzzy modeling method to estimate missing logsin hydrocarbon reservoirs. J Pet Sci Eng 112:310–321. https://doi.org/10.1016/j.petrol.2013.11.019

14. Blanes de Oliveira LA, de Carvalho CC (2021) Synthetic geochemical well logs generation using ensemble machine learning techniques for the Brazilian pre-salt reservoirs. J Pet Sci Eng 196:108080. https://doi.org/10.1016/j.petrol.2020.108080

15. Rostamian A, Heidaryan E, Ostadhassan M (2022) Evaluation of different machine learning frameworks to predict CNL-FDC-PEF logs via hyperparameters optimization and feature selection. J Pet Sci Eng 208:109463. https://doi.org/10.1016/j.petrol.2021.109463

16. Wu L, Dong Z, Li W et al (2021) Well-logging prediction based on hybrid neural network model. Energies (Basel). https://doi.org/10.3390/en14248583

17. Jang J-SR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23:665–685. https://doi.org/10.1109/21.2565541

18. Gowida A, Elkatatny S, Gamal H (2021) Unconstrained compressive strength (UCS) prediction in real-time while drilling using artificial intelligence tools. Neural Comput Appl 33:8043–8054. https://doi.org/10.1007/s00521-020-05546-7

19. Gamal H, Abdelaal A, Elkatatny S (2021) Machine learning models for equivalent circulating density prediction from drilling data. ACS Omega 6:27430–27442. https://doi.org/10.1021/acsomega.1c04363

20. Desouky M, Tariq Z, Aljawad MS et al (2021) Machine learning-based propped fracture conductivity correlations of several shale formations. ACS Omega 6:18782–18792. https://doi.org/10.1021/acsomega.1c01919

21. Gunrey K (Kevin N) (1997) An introduction to neural networks. UCL Press

22. Hiba M, Ibrahim AF, Elkatatny S, Ali A (2022) Application of machine learning to predict the failure parameters from conventional well logs. Arab J Sci Eng. https://doi.org/10.1007/s13369-021-06461-2

23. Asante-Okyere S, Shen C, Ziggha YY et al (2018) Investigating the predictive performance of Gaussian process regression in evaluating reservoir porosity and permeability. Energies (Basel). https://doi.org/10.3390/en1123261

24. Rasmussen C, Williams C (2006) Gaussian processes for machine learning. MIT press, Cambridge, MA

25. Onalo D, Adedigba S, Oloruntobi O et al (2020) Data-driven model for shear wave transit time prediction for formation evaluation. J Pet Explor Prod Technol 10:1429–1447. https://doi.org/10.11302/020-00843-2

26. Yu H, Wang Z, Rezaee R, et al. (2016) The Gaussian process regression for toc estimation using wireline logs in shale gas reservoirs. In: Day 1 Mon, November 14, 2016, IPTC

27. Yager RR, Filev DP (1994) Generation of fuzzy rules by mountain clustering. J Intell Fuzzy Syst 2:209–219

28. Chiu SL (1994) Fuzzy model identification based on cluster estimation. J Intell Fuzzy Syst 2:267–278. https://doi.org/10.3233/IFS-1994-2306

29. Feurer M, Klein A, Eggensperger K et al (2015) Efficient and Robust Automated Machine Learning. In: Cortes C, Lawrence N, Lee D et al (eds) Advances in neural information processing systems. Curran Associates Inc., New York

30. Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. Adv Neural Inf Process Syst 25

31. Mohammed R, Cawley G (2017) Over-fitting in model selection with Gaussian process regression. 192–205

32. Ebden M (2015) Gaussian processes: a quick introduction

33. Wang J (2020) An intuitive tutorial to Gaussian processes regression

34. Schulz E, Speekenbrink M, Krause A (2018) A tutorial on Gaussian process regression: modelling, exploring, and exploiting functions. J Math Psychol 85:1–16. https://doi.org/10.1016/j.jmp.2018.03.001

35. Melo J (2012) Gaussian processes for regression: a tutorial. Technical report

36. Kilian Weinberger (2018) Lecture 15: Gaussian processes

37. Do CB, Lee H (2007) Gaussian processes. Stanford University, Stanford, CA. Accessed 5 Dec 2017

38. De Iaco S, Hristopulos DT, Lin G (2022) Special issue: geostatistics and machine learning. Math Geosci 54:459–465. https://doi.org/10.1007/s11004-022-09998-6

39. Cui T, Pagendam D, Gülfedder M (2021) Gaussian process machine learning and Kriging for groundwater salinity interpolation. Environ Modell Softw 144:105170. https://doi.org/10.1016/j.envsoft.2021.105170

40. Christianson RB, Pollyea RM, Gramacy RB (2022) Traditional kriging versus modern Gaussian processes for large-scale mining data

41. Neal RM (1996) Priors for Infinite Networks. pp 29–53

42. Zeni C, Rossi K, Glielmo A, Baletto F (2019) On machine learning force fields for metallic nanoparticles. Adv Phys X 4:1654919. https://doi.org/10.1080/23746149.2019.1654919

43. Lee J, Bahri Y, Novak R, et al. (2017) Deep neural networks as Gaussian processes

44. Yu X, Peng Q, Xu L et al (2021) A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm. Inf Process Manag 58:102691. https://doi.org/10.1016/j.ipm.2021.102691

45. Yu X, Chu Y, Jiang F et al (2018) SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features. Knowl Based Syst 141:80–91. https://doi.org/10.1016/j.knosys.2017.11.010

46. Xu Y, Jiang F, Du J, Gong D (2019) A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains. Pattern Recognit 94:96–109. https://doi.org/10.1016/j.patcog.2019.05.030

47. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl Based Syst 46:109–132. https://doi.org/10.1016/j.knosys.2013.03.012

48. Jain AK, Murty MN, Flynn PJ (1999) Data clustering. ACM Comput Surv 31:324–323. https://doi.org/10.1145/331499.331504

49. Saxena A, Prasad M, Gupta A et al (2017) A review of clustering techniques and developments. Neurocomputing 267:664–681. https://doi.org/10.1016/j.neucom.2017.06.053

50. Wilson AG, Hu Z, Salakhutdinov R, Xing EP (2016) Stochastic variational deep kernel learning
51. Iwata T, Ghahramani Z (2017) Improving output uncertainty estimation and generalization in deep learning via neural network Gaussian processes
52. Liu H, Ong Y-S, Shen X, Cai J (2018) When Gaussian process meets big data: a review of scalable GPs
53. Jankowiak M, Pleiss G, Gardner JR (2020) Deep sigma point processes
54. Salimbeni H, Deisenroth M (2017) Doubly stochastic variational inference for deep Gaussian processes

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.