Scientific Contributions Oil & Gas, Vol. 46. No. 3, December: 115-127

SCIENTIFIC CONTRIBUTIONS OIL AND GAS
Testing Center for Oil and Gas
LEMIGAS

Journal Homepage:http://www.journal.lemigas.esdm.go.id
ISSN: 2089-3361, e-ISSN: 2541-0520

Trap Prevention in Machine Learning in Prediction of Petrophysical Parameters: A Case Study in The Field X

Adam Putra Pratama Zainuri[1], Pahala Dominicus Sinurat[1], Dedy Irawan[1] and Hari Sasongko[2]

[1]Faculty of Mining and Petroleum Engineering
Ganesha 10 Street, Lb. Siliwangi, Coblong, Bandung City, West Java, 40132, Indonesia

[2]Infosys Consulting
818 Town & Country Blvd Suite 600, Houston, TX 77024, USA.

Corresponding author: hari.sasongko@infosys.com.

Manuscript received: October 19st, 2023; Revised: November 21st, 2023
Approved: December 14st, 2023; Available online: December 22st, 2023

ABSTRACT - Petrophysical parameters such as porosity and water saturation are vital in the petroleum industry for reservoir characterization. These aspects are typically assessed through laboratory measurements of core samples or intricate petrophysical calculations. Machine Learning (ML) offers a cost-effective and efficient approach as an alternative to conventional methods of predicting those parameters. However, developing ML models could be prone to invisible traps such as overfitting, underfitting, feature selection, and feature importance. This study aims to share how to identify the traps and their mitigation by establishing a synergistic workflow between ML and petrophysical theory. A model was developed based on data from several wells in the X field, where they are randomized and split into test and train data. Well-log normalization preceeded data splitting, and input features were normalized with outlier removal. A feature selection function was then employed to choose a specific amount of log data. Finally, the model selection function identified the highest-scoring model. Without a proper workflow, overfitting, irrelevant feature selection, and imprecise ranking issues emerged. However, these invisible traps were mitigated with the proper workflow, even with a relatively small data set. The final model could accurately predict porosity and water saturation.

Keywords: porosity, water saturation, machine learning, reservoir characterization, feature selection.

How to cite this article:

Adam Putra Pratama Zainuri, Pahala Dominicus Sinurat, Dedy Irawan and Hari Sasongko, 2023, Trap Prevention in Machine Learning in Prediction of Petrophysical Parameters: A Case Study in The Field X, Scientific Contributions Oil and Gas, 46 (3) pp. 115-127. DOI.org/10.29017/SCOG.46.3.1586.

INTRODUCTION

Reservoir characterization plays an important role in the petroleum industry. Precise reservoir characterization is a key in reservoir development, monitoring, management, and production optimization (Aminzadeh et al., 2013). Considering this importance, it is imperative to integrate all available geologic and petrophysical parameters at their respective scales. Conventional methods such as petrophysical logs and core analyses are expensive and time-consuming. With technological advancement, machine learning (ML) and artificial intelligence (AI) are becoming new additions to traditional reservoir characterization, which includes predicting petrophysical parameters such as porosity and water saturation. However, blindly using ML would result in an inaccurate prediction, leading to an unusable algorithm. This inaccuracy stems from the prevalent traps and pitfalls encountered in machine learning applications, especially in the scientific field. These traps and pitfalls challenge the development of an ML model (Vento et al., 2019). The primary objective of this study is to present an overview of the challenges encountered and the corresponding solutions adopted, focusing specifically on predicting porosity and water saturation. Additionally, a well-defined work flow will be provided to guide the development of a machine-learning model, specifically for predicting petrophysical parameters.

The dataset used in this study consists of depth, coordinate, formation, and log data obtained from 55 wells located in the X field. Some extensive data preparation techniques were employed to ensure data quality, including data cleaning and a custom feature selection function for log data. After the quality control, the resulting dataset was used to develop an ML model.

METHODOLOGY

The dataset used to develop a regression model is based on the depth, coordinate, formation, and well log data of 55 wells in the X field. The initial step involves data preparation, encompassing missing value cleaning, outlier detection, and well-log normalization. Subsequently, the dataset was split into training and testing sets and normalized. A customized feature selection function was employed to identify the relevant well logs for prediction. This step is followed by utilizing the customized model selection function to determine the optimal predictive model. Finally, the model was improved by hyperparameter tuning before being used to predict porosity and water saturation. The general workflow used in this study is illustrated in Figure 1.

Data preparation

The data preparation consists of several steps. The first is missing value removal, followed by well-log normalization, and finally, data outlier detection. The initial removal of missing values focuses on the formation column. This initial removal of the formation column is crucial for well-log normalization to work as intended. Next, missing value removal is performed only on the features column, excluding the target column (i.e., porosity and water saturation). A threshold of 50% is used. Columns with over 50% missing values are removed, and any rows containing missing values in the remaining columns are removed, leading to a complete and accurate dataset, which is important for the whole machine learning process, as shown in a recent study by (Gonzalez et al., 2023).

A well-log normalization is done on the clean dataset. Well-log normalization plays a critical role in rendering well-log data free of systematic errors so that it could be used to develop an accurate machine-learning model (Akkurt et al., 2019). The normalized value () of an unnormalized log curve () is given by,

$$V_{norm} = R_{min} + (R_{max} - R_{min})\frac{(V_{log} - W_{min})}{(W_{max} - W_{min})} \quad (1)$$

W_{min} is the value of a particular lithology found in each well, typically close to the minimum value for that curve within the interval. It is the value of another specific lithology in each well, usually near the maximum value for that curve within the interval. The best regional estimates of the accurate values for these two lithologies at that specific location. Well-log normalization is typically applied to gamma ray, neutron porosity, bulk density, sonic, and spontaneous potential logs. Only under specific requirements are resistivity logs subjected to normalization (Shier, 2004).

The last step of data preparation is the outlier detection. In this study, an isolation forest function was used to detect the outliers in the dataset. To maintain the accuracy of the outlier detection, the dataset's wells are segregated, and outlier detection is executed individually for each well. The next step involves replacing the outliers through the interpolation of two values. The separated result will then be merged again into a complete dataset. The detailed workflow of this data preparation step is illustrated in Figure 2.

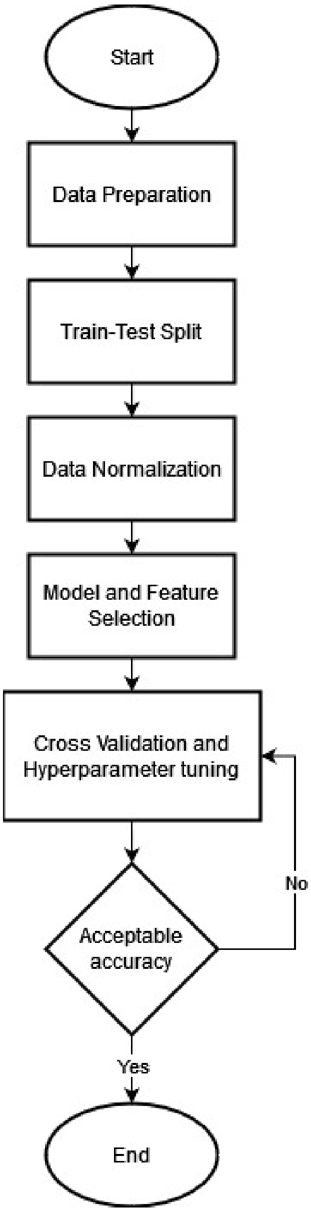


Figure 1
General workflow.

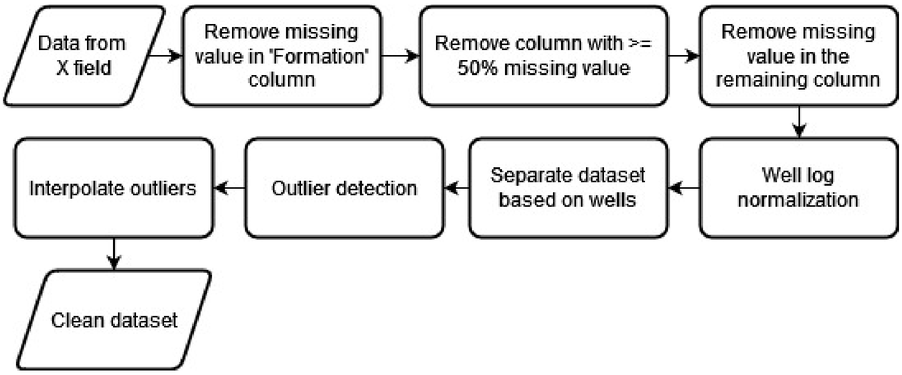

Data from X field
Remove missing value in 'Formation' column
Remove column with >= 50% missing value
Remove missing value in the remaining column
Well log normalization
Separate dataset based on wells
Outlier detection
Interpolate outliers
Clean dataset

Figure 2
Data preparation workflow.

Train-test split and data normalization

To ensure no data leaking, a specific method of splitting the training and testing was proposed by (Andersen et al., 2022). This step is illustrated in Figure 3.

One well is excluded from the dataset. This well will then be used as a test dataset labeled as MT.

Next, the dataset will be randomized and split into model datasets used to develop the regression model and the test dataset labeled MD-T, with a ratio of 90:10.

The model dataset will then be split into train (TR) and test (TE) datasets with a ratio of 80:20. The final regression model performance will be tested on TE and unseen data MT and MD-T.

The data normalization process was then fitted to the train dataset and then used to transform both the train and test datasets. The data normalization was done using the MaxAbsScaler function. The scaled value of this scaler depends on whether negative or positive values are present. If the dataset contains exclusively positive values, the range is confined to the interval from 0 to 1. Conversely, if the dataset comprises solely negative values, the range is limited from -1 to 0. However, if both negative and positive values are present, the range spans from -1 to 1. The formula for MaxAbsScaler is given by,

$$X_i' = \frac{X_i}{abs(X_{max})} \tag{2}$$

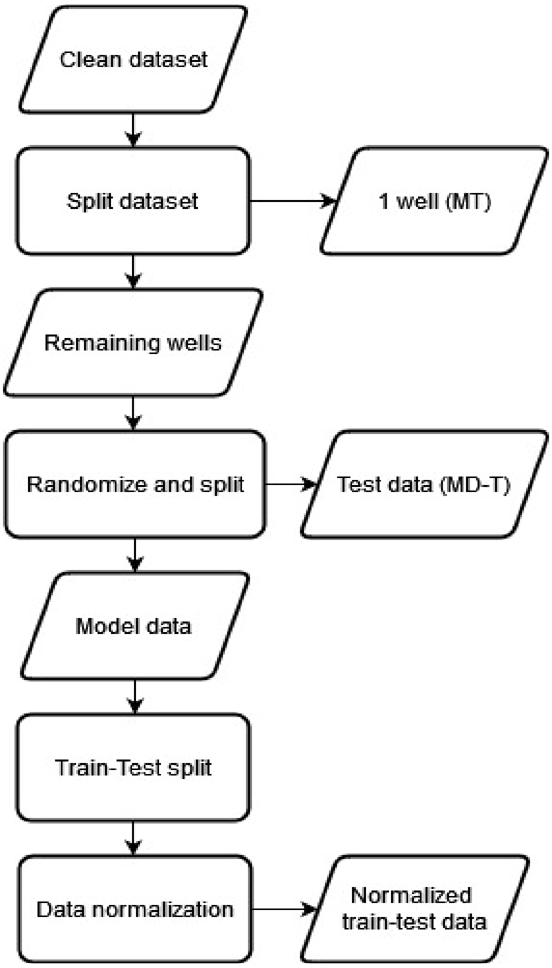


Figure 3
Train-test split and data normalization workflow.

Model and feature selection

High-dimensional data poses a challenge in developing an ML model. A common approach to tackle this issue is feature selection, which involves removing irrelevant and redundant data. This can lead to improved computation time, enhanced learning accuracy, and a better grasp of the learning model or data (Cai et al., 2018). We utilize a custom feature selection function, which incorporates feature ranking based on multiple parameters, as demonstrated in a recent study by (Miah et al., 2020).

This function determines the best features to use for a certain regression model. It would generate combinations based on the available features. These combinations will then be evaluated by applying them to the model. The highest performance feature combination would be used as the selected features. The parameters used to rank these feature combinations are Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and the p-value of the Kolmogorov-Smirnov test. The different combinations would be ranked first by RMSE and followed by p-value, MAPE, and MAE. There are no definitive answers as to what order works best. However, a p-value is placed second to ensure the model can predict a normal distribution. A p-value more significant than 0.05 indicates that the data is normally distributed.

A regression model is needed as an input of the feature selection function. Therefore, in this case, a model selection should be done before feature selection. A custom function does the model selection. In this function, several regression models are evaluated and ranked by RMSE, p-value, MAPE, MAE, and its ability to select the relevant well logs to be used referring to the existing scientific knowledge, such as the ability to find the relationship between features. For example, model A selects resistivity, sonic, gamma ray, and neutron porosity logs to predict porosity. Based on existing petrophysical knowledge, the porosity prediction relies mainly on sonic, gamma ray, bulk density, and neutron porosity log. Model A could only select 3 of the 4 relevant well logs. The workflow of these two functions is illustrated in Figure 4 and Figure 5.

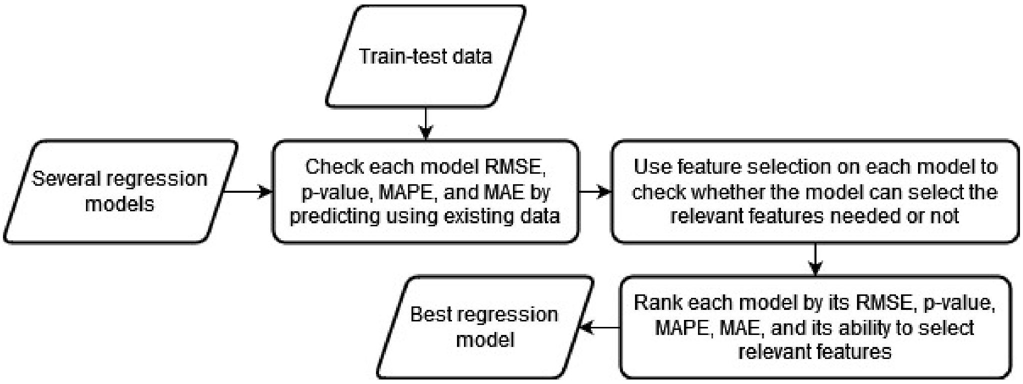

Train-test data
Several regression models
Check each model RMSE, p-value, MAPE, and MAE by predicting using existing data
Use feature selection on each model to check whether the model can select the relevant features needed or not
Rank each model by its RMSE, p-value, MAPE, MAE, and its ability to select relevant features
Best regression model

Figure 4. Model selection function workflow.

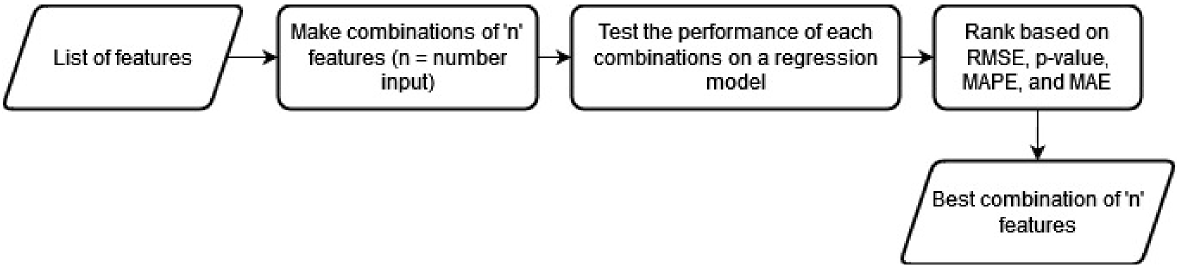

List of features
Make combinations of 'n' features (n = number input)
Test the performance of each combinations on a regression model
Rank based on RMSE, p-value, MAPE, and MAE
Best combination of 'n' features

Figure 5
Feature selection function workflow.

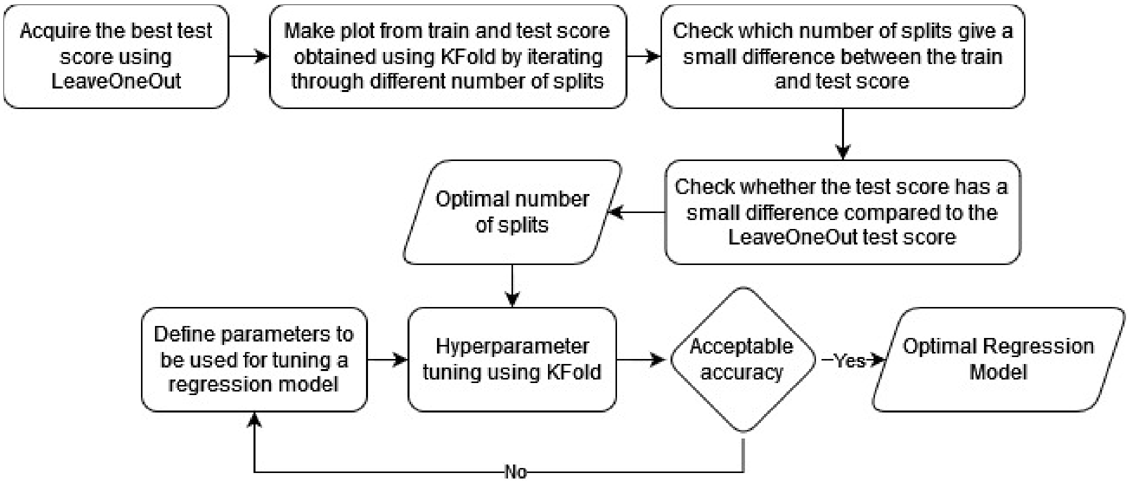

Acquire the best test score using LeaveOneOut
Make plot from train and test score obtained using KFold by iterating through different number of splits
Check which number of splits give a small difference between the train and test score
Check whether the test score has a small difference compared to the LeaveOneOut test score
Optimal number of splits
Define parameters to be used for tuning a regression model
Hyperparameter tuning using KFold
Acceptable accuracy
Yes
Optimal Regression Model
No

Figure 6
Cross-validation and hyperparameter tuning workflow.

Cross-validation and hyperparameter tuning

Cross-validation is essential in developing a machine-learning model. It is used to evaluate the model's effectiveness, particularly in scenarios where overfitting needs to be addressed (Brodeur et al., 2020). Additionally, it aids in identifying the optimal hyperparameters that minimize test error. The cross-validation used in this study is LeaveOneOut and KFold cross-validation. The scoring parameter used is Negative Root Mean Squared Error.

In investigating potential overfitting or underfitting in the model through KFold cross-validation, the primary step involves acquiring the best test score utilizing the LeaveOneOut method. This process will take a long time as it is computationally expensive. Thus, it is recommended only for small datasets. Subsequently, the train and test score plots facilitate a comprehensive assessment of overfitting or underfitting tendencies. From these plots, the optimal number of splits can be obtained. A good model is indicated by the slight difference between the train and test scores, showing its good performance on both datasets. An additional but optional parameter is also used to obtain the optimal number of splits, which is the difference between the test score and the optimal test score from LeaveOneOut. The optimal number of splits can be found by ranking the number of splits based on the difference in train and test scores alongside the difference between test scores and optimal test scores. Using the obtained number of splits, hyperparameter tuning can then be done on the regression model using KFold cross-validation. This step is illustrated in Figure 6.

RESULTS AND DISCUSSION

Porosity prediction

Using the described workflow, only a small percentage of data was clean and ready to be used to develop the regression model. There are several reasons for the missing data in well logging, such as tool failures, human error, and borehole environment issues.

Imputation of missing value was not performed due to bias tendency. This resulted in a small dataset that is susceptible to overfitting. To prevent that, further value removal should not be done to preserve the existing data and improve the dataset's performance. This was done by interpolating instead of removing outliers, which were found by utilizing isolation forest shown in Figure 7.

Before outlier detection, a well-log normalization was conducted. A specific well was chosen as the key well, which will be used as a reference for the other well's log data. This was because this specific well was found to have a large amount of data in the dataset compared to the other wells. This specific well also has almost every formation in the X field. Next, using the model selection function, it was found

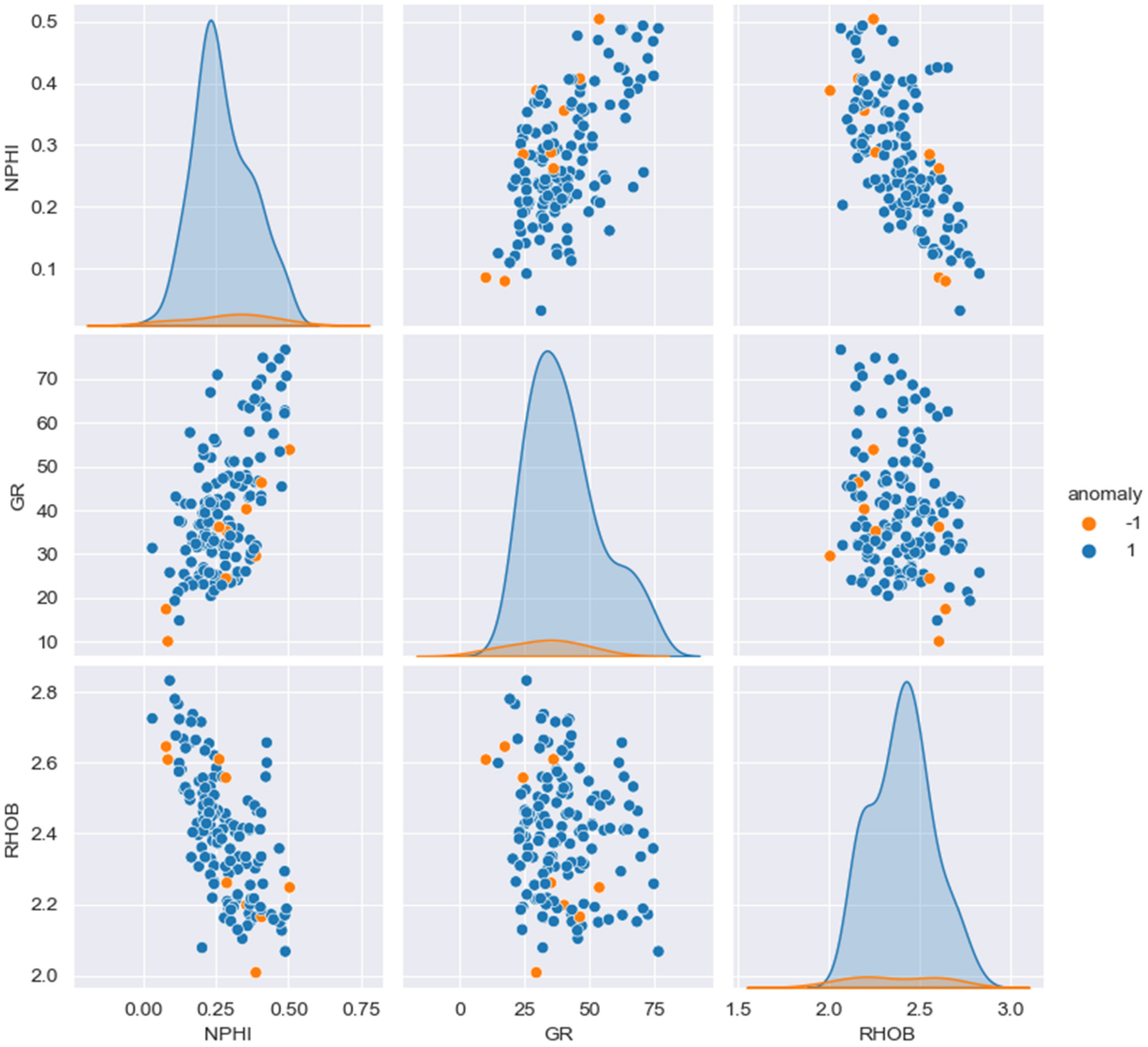


Figure 7
Detected outliers in the dataset from 3 selected well logs after utilizing isolation forest (outliers are shown in orange).

that the Light GBM regressor outranks other models. Table 1 shows the performance of each regression model. The error metrics here used negative values instead of their usual positive counterparts. It was shown that Light GBM outperforms every other model by first looking at the log selected. Light GBM shows that it can select relevant well-log data to predict porosity. Next, the models were ranked based on their RMSE. Light GBM only lost to RandomForest in terms of RMSE. Then, the p-value based on the Kolmogorov–Smirnov test was used to ensure each model prediction's normal distribution. MAPE and MAE were the last for the model to be ranked. Feature selection was then applied to the dataset with the selected model. Several well-log data combinations were ranked based on their RMSE, p-value, MAPE, and MAE. Table 2 shows the performance of the top 5 combinations from all possible well-log data combinations. It was then determined that the combination of DTC, GR, and RHOB works best for the selected model to predict porosity.

Table 1
Regression models performance.

Model	Count	RMSE	MAPE	MAE	p-value	Log
Light GBM	3	-0.030808	-0.164726	-0.030808	9.26604E-22	DTC, GR, RHOB
RandomForest	2	-0.030734	-0.162285	-0.03066	1.89775E-22	RMED, DTC, GR
GradientBoosting	2	-0.035082	-0.185432	-0.035111	3.98355E-23	DTC, RHOB, DRHO
K-nearest Neighbours	2	-0.036898	-0.192354	-0.036898	1.67426E-20	NPHI, GR, CALI
Multi-Layer Perceptron	2	-0.072015	-0.383556	-0.067583	3.52758E-24	NPHI, RHOB, DRHO
CatBoost	1	-0.032768	-0.173822	-0.032768	1.58323E-23	SP, RHOB, DRHO
XGBoost	0	-0.036147	-0.186993	-0.036147	1.40853E-23	RDEEP, DRHO, CALI
DecisionTree	0	-0.038140	-0.206189	-0.03748	1.21314E-23	RDEEP, RMED, SP
SupportVector	0	-0.043031	-0.201159	-0.043031	4.46249E-20	RDEEP, RMED, DRHO
GaussianProcess	0	-0.247124	-1.077932	-0.247124	1.2286E-23	RMED, SP, DRHO

Table 2
Top 5 combination performance.

Log Combination	RMSE	MAPE	MAE	p-value
DTC, GR, RHOB	-0.024197	-0.102765	-0.018921	7.048761E-22
DTC, NPHI, RHOB	-0.024337	-0.105002	-0.019293	9.049481E-22
RMED, DTC, NPHI	-0.024389	-0.104832	-0.019351	5.088636E-22
SP, DTC, GR	-0.024462	-0.105847	-0.019585	7.247938E-22
DTC, NPHI, GR	-0.024463	-0.104773	-0.019201	5.389149E-22

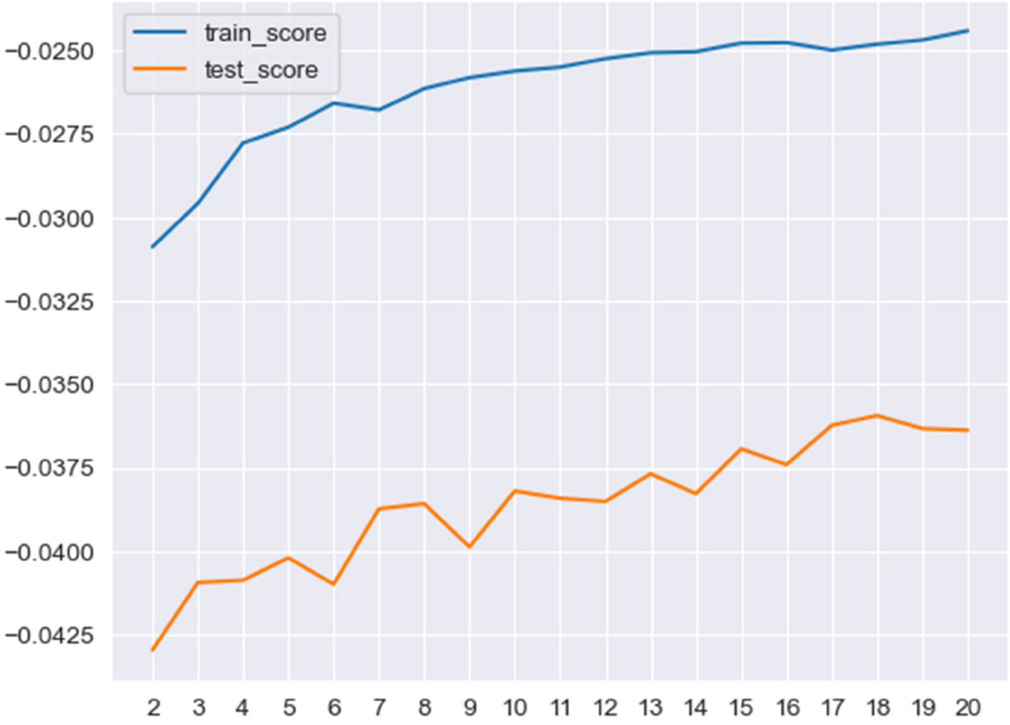


Figure 8
Cross-validation plot of light GBM regressor.

| DOI.org/10.29017/SCOG.46.3.1586

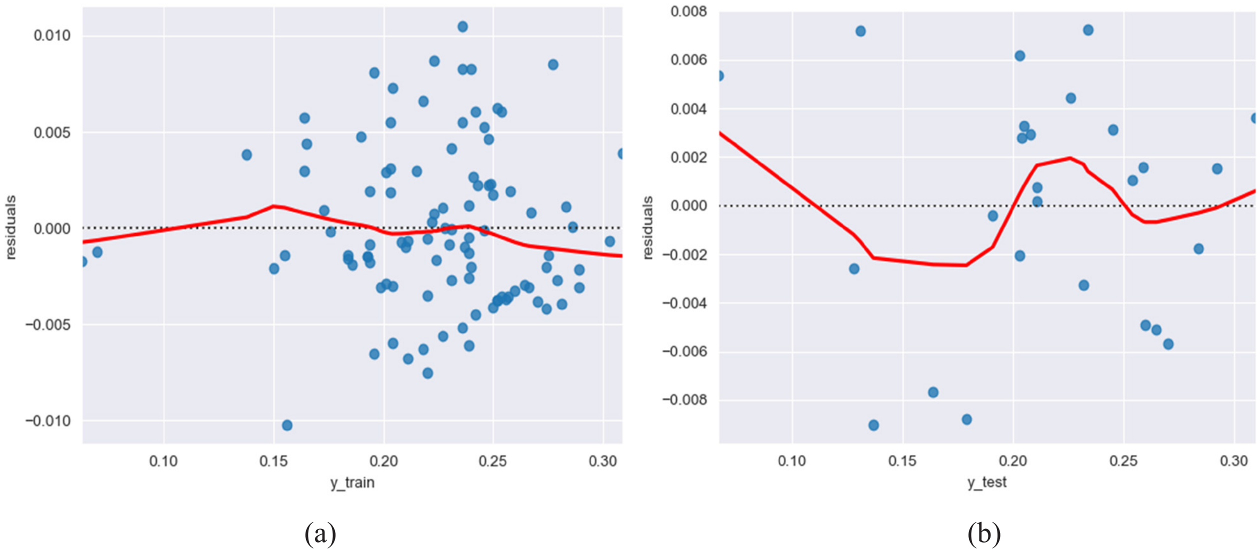


Figure 9
Light GBM residual plot. (a) Train dataset (b) test dataset.

Table 3
Light GBM regressor performance.

	MAE	MSE	RMSE
Train dataset	0.029548	0.001515	0.038918
Test dataset	0.042652	0.003024	0.054994
MT dataset	0.012047	0.000190	0.013781
MD-T dataset	0.029956	0.001595	0.039941

Using the selected model and features, cross-validation using negative RMSE was executed. Using LeaveOneOut an optimal test score of 0.031517 was obtained. Using KFold with a split ranging from 2 to 20, Figure 8 shows a plot of train and test scores. Upon closer examination of the plot, it can be concluded that the model is overfitting. The lack of data caused this to develop the model.

By comparing the difference between train and test score and the difference between test score and the optimal test score for each split, it was found that 18 split is the optimal value. This means that with 18 splits of KFold, the model was less overfitting compared to the other number of splits and closer to the optimal value of the test score. The Light GBM regressor was tuned to achieve optimal performance using this number of splits. Table 3 shows the train and test dataset's MAE, MSE, and RMSE scores. Despite the overfitting, the model could perform well on all 3 of the test datasets. The prediction made by the model also follows a good distribution, not following a significant pattern. This could be seen in Figure 9. The feature importance of this model could be seen in Figure 10. According to petrophysical knowledge, well-log data should significantly predict petrophysical parameters. This was also shown in several studies by Al-Qahtani et al. (2019). The model could synergize with this, demonstrated by its tendency to rank well log higher than X-Y coordinates and depth.

Water saturation prediction

The prediction of water saturation follows the same workflow as porosity prediction. The regressor model CatBoost was selected, and a combination of spontaneous potential, sonic, and medium resistivity logs was chosen. According to petrophysics, water saturation prediction relies on the same well log as porosity prediction, excluding spontaneous log and with an addition of resistivity log. From cross-validation, 20 splits were used to perform hyperparameter tuning on the model. Figure 11, Figure 12, Figure 13, and Table 4 show the cross-validation plot, residual plot, feature importance, and model's score, respectively.

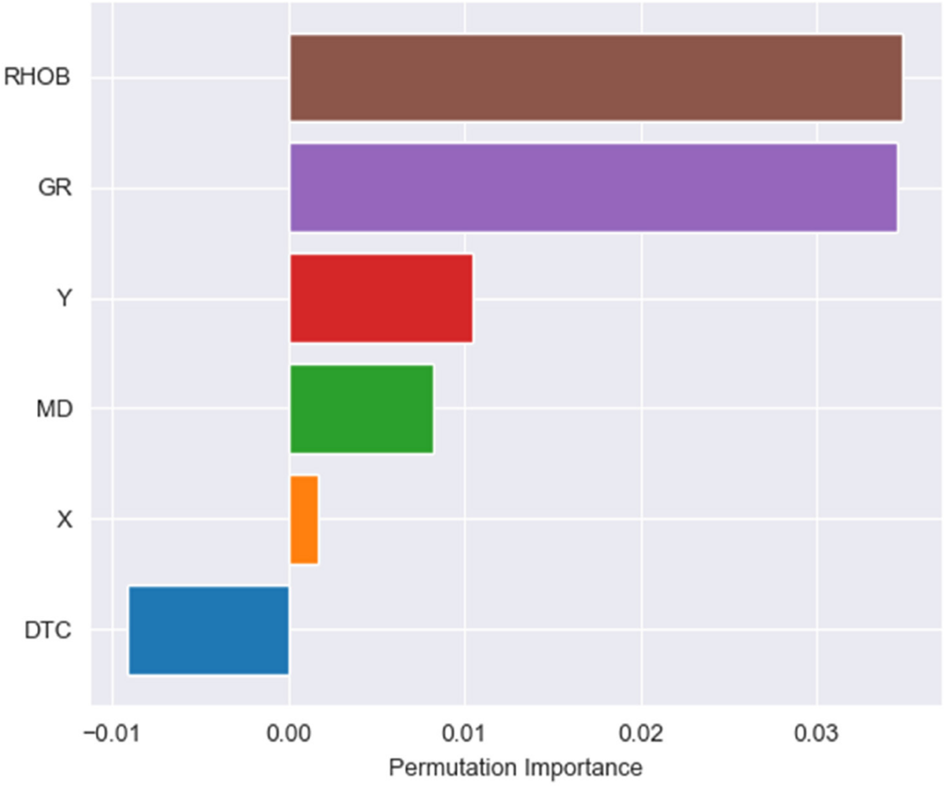


Figure 10
Porosity prediction feature importance.

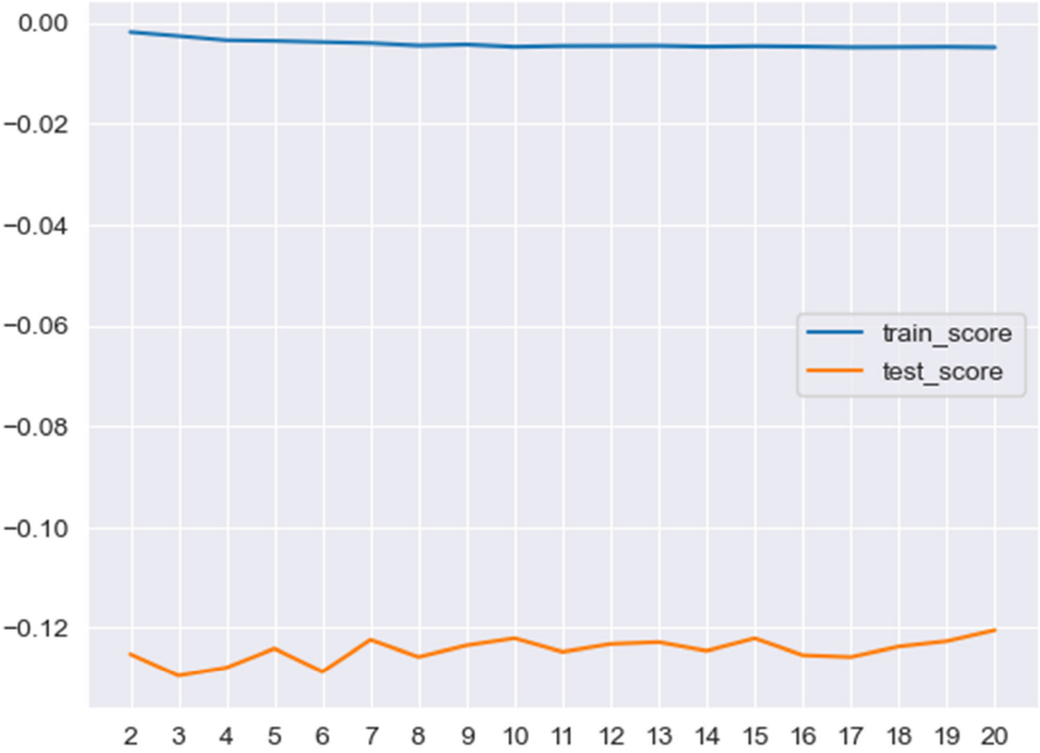
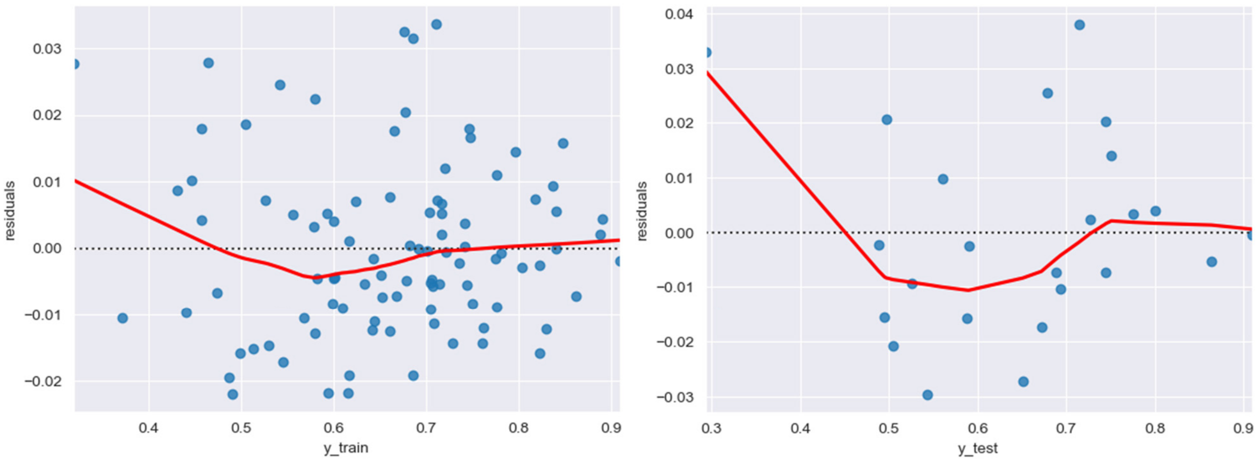


Figure 11
Cross-validation plot of CatBoost regressor.



(a) (b)

Figure 12
CatBoost residual plot. (a) Train dataset (b) test dataset.

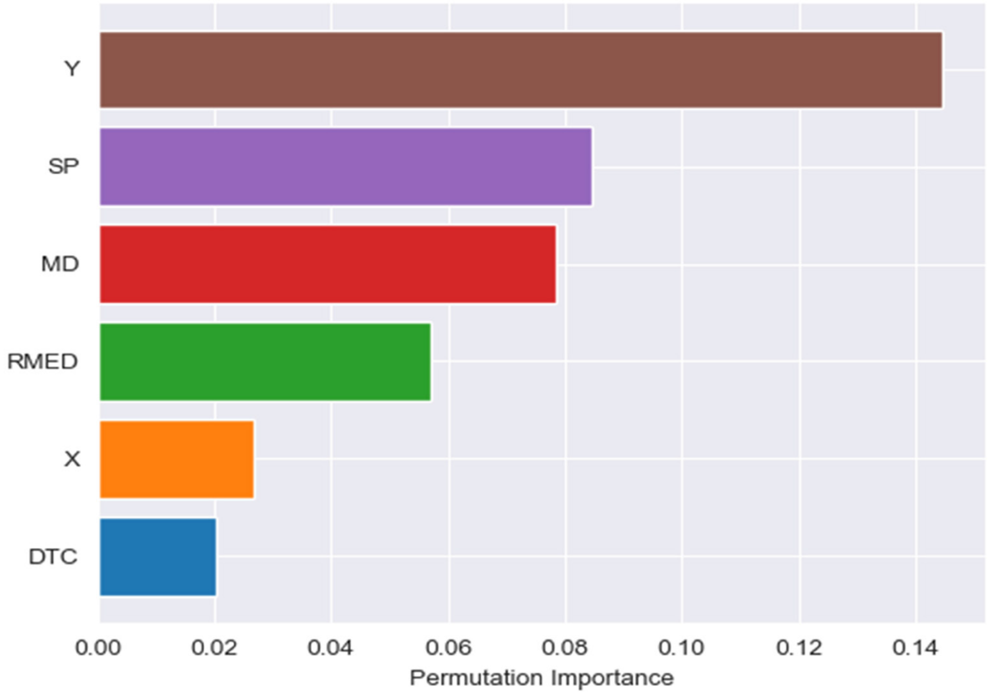


Figure 13
Water saturation feature importance.

Table 4
CatBoost regressor performance.

	MAE	MSE	RMSE
Train dataset	0.085306	0.010847	0.104150
Test dataset	0.106524	0.015872	0.125986
MT dataset	0.102435	0.017520	0.132363
MD-T dataset	0.103773	0.015775	0.125599

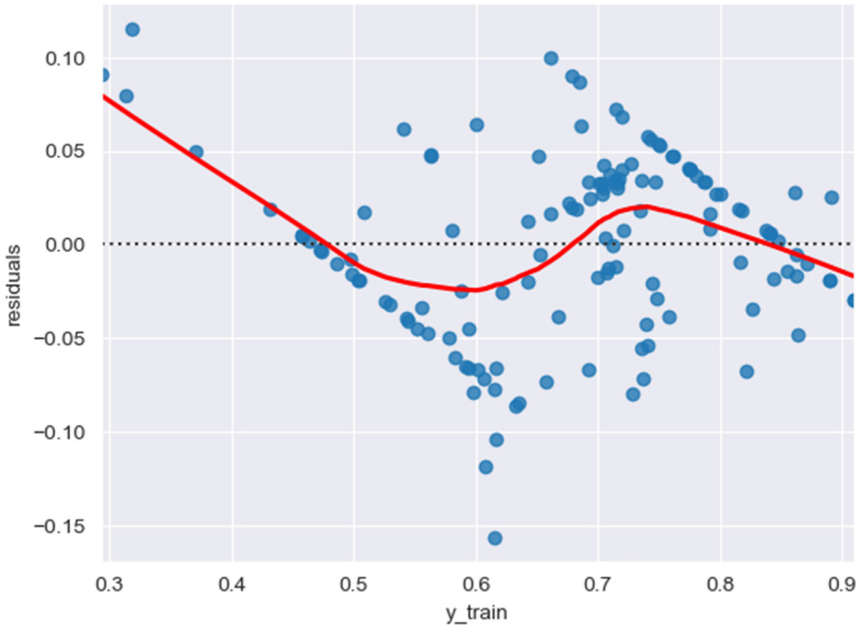


Figure 14
Residual plot from a model developed without proper workflow.

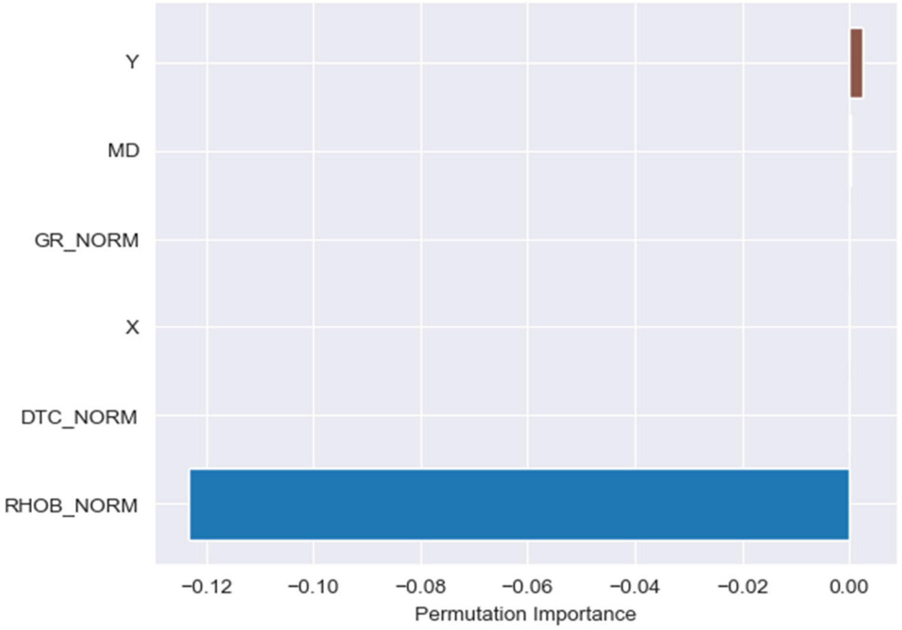


Figure 15
Feature importance from a model developed without proper workflow.

As shown in the cross-validation plot, the model is overfitting, which is caused by the lack of data. The residual plots show that the predictions were well-distributed and did not follow a certain pattern. The feature importance of this prediction relies heavily on Y coordinates. This should not be true, as water saturation prediction relies mainly on well logs, especially resistivity logs. This problem is also caused by overfitting.

Comparison With and Without Proper Workflow

Machine learning cannot be blindly used. Problems such as overfitting and underfitting can happen to the model without a proper workflow. Feature selection will not work as intended, as it could select irrelevant features to be used. This will result in an overlearning model that cannot be used outside the original dataset. In this case, feature ranking should rank well-log data above coordinate and depth data. However, improper workflow would result in an undesired feature rank. Additionally, when observing the residual plot, a flawed workflow would lead to distribution with a noticeable pattern, which is undesirable. Figure 14 and Figure 15 show a residual plot and feature ranking from a model developed without a proper workflow.

CONCLUSION

A calculated and precise workflow is essential in developing a machine-learning model. With the correct workflow, traps and pitfalls could be prevented or minimized. In this case, the effect of overfitting in the relatively small dataset could be minimized, and further problems were prevented. The model could predict porosity and water saturation value based on relevant well logs with an acceptable accuracy.

ACKNOWLEDGE

This work was financially supported by the Research, Community Service, and Innovation Program, The Faculty of Mining and Petroleum Engineering, Institute of Technology Bandung (ITB).

REFERENCES

Akkurt, R., Miller, M., Hodenfield, B., Pirie, I., Farnan, D., & Koley, M. (2019). Machine learning for well Log Normalization. *Day 3 Wed, October 02, 2019*.

Al-Qahtani, F. A. (2019). *Porosity distribution prediction using artificial neural networks*. West Virginia University Libraries.

Aminzadeh, F., & Dasgupta, S. N. (2013). Reservoir Characterization. In *Developments in Petroleum Science* (pp. 151–189). Elsevier.

Andersen, P. Ø., Skjeldal, M., & Augustsson, C. (2022). Machine learning based prediction of porosity and water saturation from Varg field reservoir well logs. *Day 4 Thu, June 09, 2022*.

Brodeur, Z. P., Herman, J. D., & Steinschneider, S. (2020). Bootstrap aggregation and cross-validation methods to reduce overfitting in reservoir control policy search. *Water Resources Research*, *56*(8). https://doi.org/10.1029/2020wr027184

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

Gonzalez, K., Brusova, O., & Valenciano, A. (2023). A machine learning workflow for log data prediction at the Basin scale. *First Break*, *41*(2), 73–80. https://doi.org/10.3997/1365-2397.fb2023015

Miah, M. I., Zendehboudi, S., & Ahmed, S. (2020). Log data-driven model and feature ranking for water saturation prediction using machine learning approach. *Journal of Petroleum Science & Engineering*, *194*(107291), 107291. https://doi.org/10.1016/j.petrol.2020.107291

Shier, D. E., (2004), Well log normalization: methods and guidelines, *Petrophysics*, *45*(03), 268-280.

Vento, D. D., & Fanfarillo, A. (2019). Traps, pitfalls and misconceptions of machine learning applied to scientific disciplines. *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)*.