MethodsX 14 (2025) 103127

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/methodsx

Filling-well: An effective technique to handle incomplete well-log data for lithology classification using machine learning algorithms ☆,☆☆

Sherly Ardhya Garini [a,b], Ary Mazharuddin Shiddiqi [a,*], Widya Utama [b], Alif Nurdien Fitrah Insani [b]

[a] Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia
[b] Department of Geophysical Engineering, Institut Teknologi Sepuluh Nopember, Indonesia

ARTICLE INFO

Method name:
Filling-Well: An Effective Technique to Handle Incomplete Well-Log Data for Lithology Classification Using Machine Learning Algorithms

Keywords:
Lithology classification
Machine learning
Missing values
Well log

ABSTRACT

Lithology classification is crucial for efficient and sustainable resource exploration in the oil and gas industry. Missing values in well-log data, such as Gamma Ray (GR), Neutron Porosity (NPHI), Bulk Density (RHOB), Deep Resistivity (RS), Delta Time Compressional (DTCO), Delta Time Shear (DTSM), and Resistivity Deep (RD), significantly affect machine learning classification accuracy. This study applied three algorithms, extreme gradient boosting (XGBoost), K-nearest neighbours (KNN), and the artificial neural network (ANN), to handle missing values in well-log datasets, particularly datasets with extreme missing data (30 %). Results indicated that XGBoost was the most efficient and accurate, especially for RHOB, NPHI, DTCO, and DTSM, with the lowest Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) values. The ANN also performed effectively, particularly on the GR, RS, and RD features, after the use of preprocessing techniques such as isolation forest and bias correction. However, the ANN can suffer from overfitting and requires large datasets for optimal performance. In contrast, KNN struggled with missing-not-at-random (MNAR) data due to its reliance on the k parameter and distance metric, making it less effective in mapping missing data relationships.

- Missing values in well-log data can hinder lithology classification accuracy for efficient resource exploration in the oil and gas industry.
- This research aims to address the problem of missing values in well-log datasets by applying machine learning algorithms such as XGBoost, ANN, and KNN to enhance classification performance.
- XGBoost demonstrated superior performance in handling extreme missing data (30 %) in well-log datasets. ANN was effective but prone to overfitting for small datasets, while KNN struggled with missing-not-at-random (MNAR) data due to limitations in its distance-based approach.

☆ **Related research article:** None
☆☆ ***For a published article:*** None
* Corresponding author.
E-mail address: ary.shiddiqi@its.ac.id (A.M. Shiddiqi).

https://doi.org/10.1016/j.mex.2024.103127
Received 19 November 2024; Accepted 20 December 2024
Available online 21 December 2024

Specifications table

Subject area:	Computer Science
More specific subject area:	*Machine Learning*
Name of your method:	Filling-Well: An Effective Technique to Handle Incomplete Well-Log Data for Lithology Classification Using Machine Learning Algorithms
Name and reference of original method:	S. W. J. Nijman et al., "Missing data is poorly handled and reported in prediction model studies using machine learning: a literature review," J. *Clin. Epidemiol.*, vol. 142, pp. 218–229, 2022, doi: 10.1016/j.jclinepi.2021.11.023. R. Zhong, R. Johnson, and Z. Chen, "Generating pseudo density log from drilling and logging-while-drilling data using extreme gradient boosting (XGBoost)," *Int.* J. *Coal Geol.*, vol. 220, p. 103,416, 2020, doi: 10.1016/j.coal.2020.103416. A. Juna et al., "Water quality prediction using KNN imputer and multilayer perceptron," *Water (Switzerland)*, vol. 14, no. 17, pp. 1–19, 2022, doi: 10.3390/w14172592. A. K. A. Mohammed and M. K. Dhaidan, "Prediction of well logs data and estimation of petrophysical parameters of Mishrif Formation, Nasiriya Field, South of Iraq using artificial neural network (ANN)," *Iraqi J. Sci.*, vol. 64, no. 1, pp. 253–268, 2023, doi: 10.24996/ijs.2023.64.1.24.
Resource availability:	N/A

Background

Lithology classification based on well-log data is crucial to the oil and gas industry. It aims to understand the composition of underground rocks. This information is important for determining the location and potential of hydrocarbon reservoirs, allowing companies to plan more effective and efficient exploration and exploitation strategies. With accurate lithological analysis, companies can reduce operational risks, maximise resource recovery, and improve the sustainability of operations. Therefore, lithology classification is important in determining drilling techniques, rock formation assessment, and strategic decision-making in the oil and gas industry [1]. Missing values are a frequent problem in well-log datasets. The reasons for missing data can be technical, such as machine malfunctions or human error, and the probability of missing data increases with the volume of data logged [2–4]. Biased estimates indicate the impact of missing data, which includes the loss of information, decreased statistical accuracy, and weakened generalisability, affecting data quality [5–7]. Missing values in well-log data can reduce the accuracy of analysis and prediction, such as in the results of a study that estimated permeability, porosity, density, and intrinsic attenuation using seismic attributes and well-log data [8,9]. Research on seismic attenuation due to induced flow using sonic log data showed that missing values affected the accuracy of interpretation [10]. This background underscores the critical role of accurate well-log data, including the impact of missing values, in identifying reservoir characteristics [11]. Therefore, handling missing values is of great importance. Many studies highlight that it is crucial to use an appropriate imputation method to handle missing values in well-log data based on the specific characteristics of the well-log data. Imputation involves replacing missing values with feasible values that are as similar as possible to the original missing values [12,13]. Missing values in the dataset should be accounted for using suitable methods to improve the accuracy and performance of data exploration [14]. In addition, different ways of handling missing data can produce different results in statistical models, thus emphasising the importance of using appropriate techniques to handle missing values in well-recorded data [15,16]. Different imputation approaches, such as average imputation or model-based imputation, can handle missing data. In addition, advanced techniques such as machine learning have proven effective in forecasting missing values and improving model efficiency [17]. Machine learning-based methods provide a promising solution for accounting for missing values in datasets [18–21]. The study of prediction and classification models using machine learning to handle missing values is still open for further research, especially in lithology classification, where incomplete data often lead to inaccurate predictions. This research comprehensively evaluates various techniques to improve the accuracy of lithology classification, ultimately contributing to more efficient exploration and exploitation of oil and gas resources.

Method details

Many machine learning algorithms have been used to handle missing values in datasets. Handling missing values in well-log data poses a challenge, where one machine learning algorithm might not perform well on all parts of the dataset. We summarise the machine learning algorithms used for handling missing values in Table 1.

We use several machine learning algorithms to predict missing values based on patterns in existing data, i.e. extreme gradient boosting (XGBoost), K-nearest neighbours (KNN), and the artificial neural network (ANN). The selection of XGBoost, KNN, and ANN as algorithms in this research is based on their widespread use and proven effectiveness in addressing the challenges of Missing Not At Random (MNAR) data. The primary focus of this study is to evaluate the performance of these models— XGBoost, KNN, and ANN—which represent diverse algorithmic approaches: boosting, distance-based proximity, and deep learning. According to prior studies, XGBoost has demonstrated high effectiveness in handling MNAR missing values, achieving an accuracy improvement of up to 20% on datasets and maintaining stability even with a missing value rate of 40% [22]. This makes XGBoost an efficient solution for both classification and MNAR imputation [23]. In addition, KNN's flexibility in employing various distance functions and the number of neighbors allows it to capture MNAR patterns within data. Several studies indicate that KNN performs well with MNAR cases, particularly when supported by strong relationships between observed variables and missing data, especially in complex datasets that do not meet distributional assumptions [24–26]. ANN, on the other hand, is recommended due to its superior performance in handling MNAR missing values at a data loss level of 30% compared to LightGBoost. ANN recorded a Mean Absolute Error (MAE)

Table 1
Summary of methods for handling missing values (strengths and weaknesses).

Methods	Strengths	Weaknesses
Machine learning-based methods	Choosing the correct method can improve data quality and make analysis results more reliable	Needs a large dataset for model training
K-nearest neighbours (KNN) algorithm	Easy to implement and understand, non-parametric (does not assume a particular data distribution), flexible for various data types and scales	Takes a long time to process if the dataset is large, sensitive to unbalanced data, and performance degrades with high-dimensional data (curse of dimensionality)
Extreme gradient boosting (XGBoost) algorithm	High accuracy because it uses ensemble learning, handles overfitting well, and is efficient	Needs sweeping parameters to achieve optimal performance and requires a lot of memory and a high computation time for very large datasets
Artificial neural network (ANN) algorithm	Capable of modelling complex non-linear relationships, highly flexible and adaptive, can be improved by adding more layers and neurons, continuously learning, and updating the model with new data	Needs large datasets for practical training, requires significant resources, is challenging to interpret due to its 'black box' nature, and is vulnerable to overfitting if not properly organised

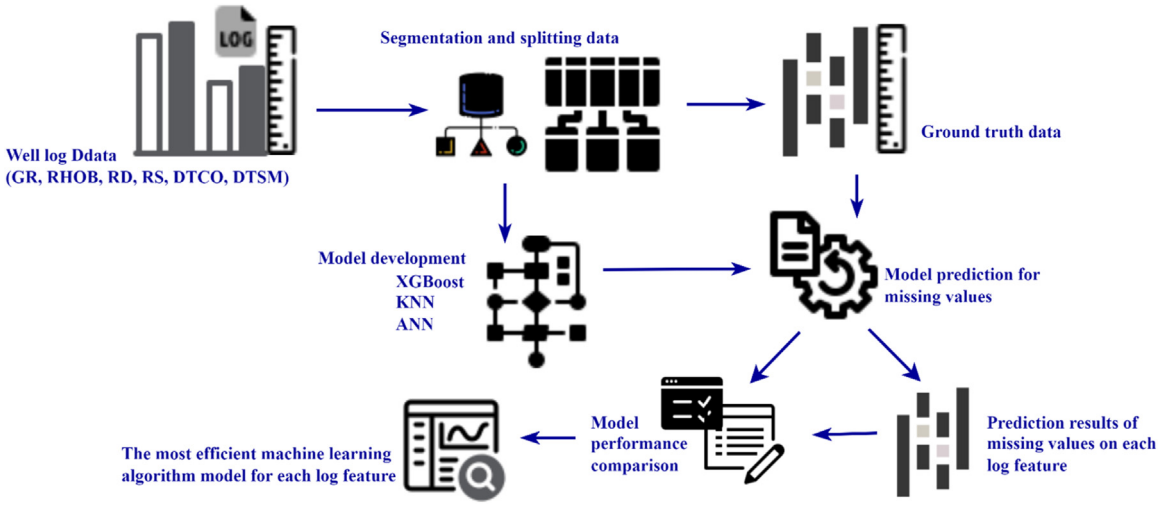

Fig. 1. Research workflow.

of 0.070 and a Root Mean Squared Error (RMSE) of 0.088, which are 1.73 and 1.80 times better, respectively, than those achieved by LightGBoost [27]. Furthermore, ANN has also shown effective performance in MNAR scenarios with up to 50% missing data, achieving high accuracy (>85%) and low MAE (0.12). Its ability to model non-linear relationships between variables makes it a preferred choice over traditional statistical methods. ANN's performance remains stable even with missing data rates as high as 50% [28]. Future research may explore additional models to further advance the understanding and handling of MNAR data. We aim to find the best algorithm that can fill in missing values in well-log data accurately by evaluating the effectiveness of the machine learning algorithms using the mean absolute percentage error (MAPE) and root mean square error (RMSE), which will be useful for lithology classification in the oil and gas sector. The workflow of the proposed method is shown in Fig. 1.

Well-logging involves measuring physical parameters along the borehole, which vary with the depth of the well [21]. The results of these measurements are referred to as log data. A log is typically a graph or curve that functions based on depth or time, with each curve representing a parameter measured continuously within a well. In oil and gas fields, well-logging plays a crucial role in exploring and producing hydrocarbon resources, particularly in the detailed analysis of hydrocarbon reservoir lithology.

In the workflow in Fig. 1, it can be seen that the research focuses on effectively imputing missing values in well-log data using machine learning algorithm-based prediction results. The process starts with collecting and segmenting well-log data, followed by model development using the XGBoost, KNN, and ANN algorithms. The trained models are then used to predict the missing values in the well-log data, and the prediction results are evaluated based on the performance of each model. The final step is to compare the performances of the models to determine the most efficient algorithm for filling the missing values, resulting in a more complete and reliable well-log dataset for further analysis methods, such as lithology classification.

Well-log data

We used five public well-log datasets owned by ConocoPhillips in the Browse Basin region, Australia (https://www.occam.com.au/poseidondata). The well-log parameters used include the gamma ray (GR), bulk density (RHOB), neutron porosity (NPHI),

Table 2
Log parameters used in the research.

No.	Log Feature	Nomenclature	Units
1	GR	Gamma Ray	gAPI
2	RHOB	Bulk Density	g/cm3
3	NPHI	Neutron Porosity	pu
4	DTCO	Differential Sonic Travel Time	us/ft
5	DTSM	Differential Transit Time Shear	us/ft
6	RS	Resistivity	Ωm
7	RD	Resistivity-Density	Ωm

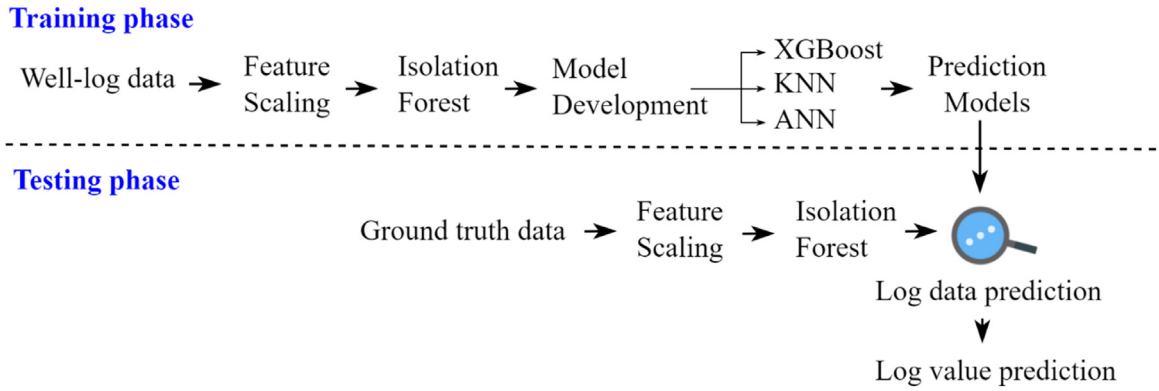

Fig. 2. Development flow of machine learning model for well-log data prediction.

sonic (DTCO and DTSM), and resistivity (RS and RD) logs (Table 2). The missing values are not random (missing not at random, MNAR) and are set to 30 % of the total data for all log features. The number of missing values was chosen to determine the most effective method and obtain realistic recommendations for datasets with extreme missing values.

Development of machine learning model

The development and testing flow of the machine learning model for predicting log features, which is then used to impute the missing values of well-log data, can be seen in Fig. 2. In the training phase, well-log data obtained from the drilling process are processed through several stages. First, feature scaling is performed to normalise the data so that the machine learning model can work more efficiently. Next, outliers are handled using the isolation forest algorithm. Then, the clean data are used to develop a log feature prediction model with the XGBoost, KNN, and ANN machine learning methods. Several stages and several techniques are used in the preprocessing stage to ensure the quality of the data used to train and test the prediction model in this research, including feature scaling and forest isolation.

Preprocessing

1. Feature scaling

Feature scaling is performed using the standard scaler method, where the scale between features in the dataset is equalised so that no feature dominates the processing using machine learning algorithms. This step aims to optimise the classification method. The standard scaler applies the Z-score principle to each feature in the dataset to normalise the data. The principle is to transform the data so that they have an average close to 0 and a standard deviation close to 1. Eq. (1) is used to calculate the Z-score:

$$Z = \frac{x - \mu}{\sigma}. \tag{1}$$

The Z-score is calculated by subtracting the average value (μ) from each individual data point (x) and then dividing the result by the standard deviation (σ) of the dataset [22].

2. Isolation forest algorithm

After feature scaling, outliers are handled using the isolation forest algorithm. The isolation forest algorithm is superior in anomaly detection; it is especially recognised for its efficiency and ability to detect outliers in datasets. Isolation forest is particularly beneficial because it requires low computational resources and can be easily applied to large datasets [23]. Unlike traditional methods that rely on distance or density measures, isolation forest uses only the concept of isolation, so it can effectively detect anomalies without the need for labelled training data. This algorithm creates a collection of isolation trees, where each tree is formed by randomly selecting a feature and a splitting value between the maximum and minimum values of the feature. This process continues until every data

point is isolated. The number of splits required to isolate a data point indicates an anomaly; the fewer the number of splits, the more likely it is that the point is an anomaly [24].

Training process

1. XGBoost

After applying isolation forest to identify and remove anomalies from the well-log data, the next step is to use the XGBoost machine learning algorithm in the model development stage. The XGBoost algorithm will be trained with clean data from the isolation forest to predict log features accurately. In this process, XGBoost will utilise its ability to capture non-linear relationships in the data and generate robust prediction models for the various log features available. Tree-based models such as XGBoost are known for capturing complex and non-linear data relationships [25]. The models excel in handling complicated data structures and correlations, making them well-suited for scenarios where the assumption of linearity may not hold. The iterative nature of the algorithm allows it to continuously improve the accuracy as the number of iterations increases [26,27]. The boosting process in the XGBoost algorithm is built gradually to minimise the loss function and improve the model's prediction performance through a gradient descent optimisation approach, as in Eq. (2):

$$F_m(X) = F_{m-1}(X) + \alpha_m h_m\left(X,\ r_{m-1}\right), \tag{2}$$

where

$F_m(X)$ is the combination of all decision trees created in each iteration;
$F_{m-1}(X)$ is the model created up to iteration $m-1$;
$h_m(X,\ r_{m-1})$ is the $m-1$ decision tree for predicting the r_{m-1} residuals;
α_m is the optimised parameter to reduce the loss function.

The loss function minimises the difference between the model prediction and the actual target value. In simple terms, the XGBoost model works iteratively by building a series of decision trees, where each tree tries to correct the error of the previous tree [28]. The model development process can be seen in the workflow in Fig. 1; it occurs after the well-log data (GR, RHOB, NPHI, RS, RD, DTCO, and DTSM) go through a preprocessing stage, including data segmentation, and outlier value handling using isolation forest. The data are then split into training and testing data with a ratio of 80:20. At the model development stage, feature scaling is performed to ensure a more even data distribution. After data splitting, the XGBoost algorithm is applied to the training data. Next, the parameters of XGBoost are optimised through a grid-search cross-validation technique, which aims to find the best combination of parameters that maximises the model's performance. After the model is trained, the prediction results are evaluated using evaluation metrics to calculate the accuracy and performance of the model.

2. KNN

Imputation techniques such as KNN imputation have been used to effectively manage missing values in datasets in predictive modelling tasks (Aljrees, 2024). The distance between data points in the KNN algorithm can be calculated using the Euclidean (Eq. (3)) and Manhattan formulas (Eq. (4)):

$$D_{euclidean}(x,\ y) = \sqrt{\sum_{j=1}^{n} |x-y|^2}, \tag{3}$$

$$D_{manhattan}(x,\ y) = \sum_{j=1}^{n} |x-y|, \tag{4}$$

where x is the sample data, and y is the new data to be predicted. D represents the distance between x and y, and n is the number of features in the dataset [29]. Imputing using KNN has several advantages, such as maintaining the integrity of the dataset without having to remove incomplete data and avoiding biases that may occur due to data deletion. In addition, imputing results have been shown to improve the performance of prediction models by providing more representative and comprehensive data [30]. Another study also highlighted that models using KNN imputers performed better than models that only removed missing data, with a significant increase in accuracy in water quality prediction [31]. KNN is one of the non-parametric algorithms used in machine learning and data mining tasks. KNN is an instance-based learning method, which means it does not build an explicit model but instead directly uses the training data to make predictions. KNN works by classifying unknown patterns or predicting class labels based on their nearest neighbours in the training data [32].

3. ANN

ANNs are powerful tools capable of modelling complex non-linear relationships due to their ability to learn and recognise complex patterns through experience [33,34]. An ANN is highly versatile and adaptable, as it can be designed with various architectures. The addition of layers and neurons increases this network's scalability [35]. These networks show continuous learning capabilities to adapt and improve performance without extensive experimentation [36]. ANNs in previous studies have been used to predict petrophysical properties such as the porosity [37] and total organic carbon (TOC) in reservoirs [38], impute missing well-log data, and estimate petrophysical parameters [39,40]. An ANN was used to predict missing well logs in the Mishrif reservoir at Well Ns-X, Nasiriya Oilfield, Iraq. The predicted log data included sonic, neutron, density, and resistivity logs, with the GR log as the only original log. The performance evaluation of the ANN model was conducted using the coefficient of determination (R^2) for each predicted log

type. The results showed the potential of ANNs in assisting reservoir development evaluation and planning [18]. In ANNs, the basic equation used for predicting a value with one hidden layer is given as Eq. (5):

$$net\_hidden = \sum_{j=1}^{J} C_{j,k} i_j + \theta_k, \tag{5}$$

where J represents the number of input nodes, and $C_{j,k}$ is the weight between input node j and hidden node k. The variable i_j refers to the input values (experimental parameters), while θ_k denotes the bias associated with hidden node k. The equation is used to calculate the net input to node k within the hidden layer. Additionally, Eq. 6 illustrates the process of determining the net input for unit z in the output layer:

$$net\_output = \sum_{k=1}^{K} D_{k,z} h_k + \emptyset_z, \tag{6}$$

where K is the number of nodes in the hidden layer, and the weight connecting hidden node k to output node z is represented by $D_{k,z}$. The output from hidden node k is denoted by h_k, while the bias applied to output node z is $\emptyset_z$. The final prediction is produced by the output layer after all the processes are finished, including the application of the activation function to the computed net input [39].

4. Hyperparameter search (grid-search cross-validation)

Hyperparameters are parameters that manage the running of a machine learning model during the training process on a dataset. By choosing the right hyperparameters for an algorithm, the performance of the algorithm can be significantly improved. This optimisation process has a major impact on the overall training process and processing time, as well as the model's ability to generalise. Due to the large number of variations in hyperparameters, specialised methods are required to find the optimal combination. In some research, grid-search cross-validation is used to determine the hyperparameter settings for each algorithm. This process systematically tries various hyperparameter combinations within a pre-determined range to find the most optimal value, with the aim of obtaining the best performance. Grid-search cross-validation ensures that the model is trained with appropriate hyperparameters, which plays an important role in improving the generalisation ability of the algorithm, resulting in a better prediction or classification accuracy [41,42]. In this research, hyperparameter tuning is used in two algorithms, XGBoost and KNN, for log feature prediction. After performing hyperparameter optimisation through grid-search cross-validation, the next step is to effectively optimise the algorithm to improve the model performance. This optimisation process plays an important role in improving the accuracy and overall performance of the model.

Post-training (bias correction)

Bias correction is a technique used in data processing and machine learning to address systematic errors in observation data or model predictions, so that the predicted results are closer to the actual values. These biases can be caused by various factors, such as tool calibration errors, model imperfections, or changes in external conditions that are not reflected well in the data. Mathematically, bias can be represented as in Eq. (7):

$$Bias\ (x) = E[\hat{y}(x)] - y(x), \tag{7}$$

where $E[\hat{y}(x)]$ is the predicted value of the model, and $y(x)$ is the actual value of the observation. In the bias correction approach, the bias is mathematically represented as a linear regression that depends on certain predictors, as represented by Eq. (8):

$$\tilde{H}(x, \beta) = H(x) + \sum_{i=0}^{N} \beta_i p_i(x), \tag{8}$$

where H(x) is the model prediction without bias correction, H̃(x,β) is the bias-corrected prediction, β_i is the bias coefficient that needs to be adjusted based on the data, and p_i (x) is the predictor used to estimate the bias. These bias coefficients are updated by minimising the difference between the prediction and the observed data [43].

Evaluation models

In this research, the MAPE and RMSE metrics were used to assess the performance of the prediction model. The MAPE measures the prediction error rate as a percentage of the actual value, thus providing an easier understanding of the magnitude of the error. The RMSE calculates the root mean square of the difference between the predicted and observed values, which is used to measure how far the model predictions deviate from the actual data. Eqs. (8) and (9) are the formulas used to calculate the MAPE and RMSE:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\left(\hat{y}_i - y_i\right)^2}{n}}, \tag{9}$$

where n is the number of data points, $\hat{y}_i$ is the predicted data, and y_i is the actual data:

$$MAPE = \sum_{i-1}^{n} \left| \frac{f_i - a_i}{a_i} \right| \times 100\%, \tag{10}$$

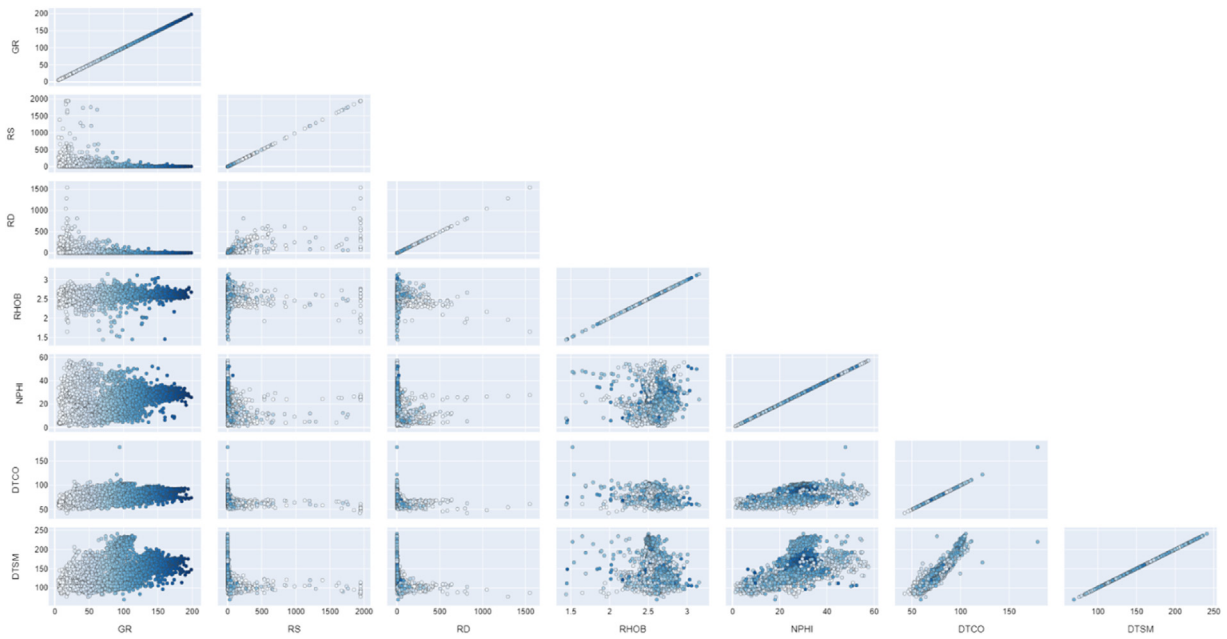

Fig. 3. Data distribution before feature scaling.

where f_i is the predicted data and a_i is the actual data [44]. The results of this evaluation provide a detailed overview of the prediction accuracy and the extent to which the model is able to handle the data well. Such metrics are an important guide in determining the most effective and robust model for log feature prediction.

Method validation

This research highlights three main stages that need to be discussed before going into a more detailed discussion of the results and analysis: the feature scaling process, the use of isolation forest in preprocessing, and the prediction evaluation of the XGBoost, KNN, and ANN algorithms. Each stage plays an important role in determining the prediction accuracy of the analysed well-log dataset.

The results of feature scaling (preprocessing step)

The distribution of the dataset before and after applying the standard scaler technique can be seen in Fig. 3 and 4. Each log feature has a diverse range of values; some features, such as log RS and RD, have a wide range of values and are widely distributed along the y-axis compared to other features (Fig. 3). For example, log RS ranges from 0 – 2000 Ωm, and log RD ranges from 0 – 1500 Ωm, where the data distribution shows high variation, as seen from the large amount of data spread along the y-axis. The diverse range of values is standardised using the standard scaler technique so that all log feature scales become uniform (Fig. 4), where the data distribution becomes more concentrated and uniform over a specific range for each log feature. This signature indicates that the standard scaler technique reduced the mean to near 0, and the scale distribution was uniform between features.

After the feature scaling process produces a more uniform data distribution, the next step is to handle outliers using the isolation forest algorithm. This process aims to identify and remove data that are considered anomalous so that the quality of the data used in the prediction model improves and can produce more accurate results.

The result of applying the isolation forest algorithm

In this research, several experimental scenarios were carried out related to the 'contamination' parameter to optimise the performance of the isolation forest model. The contamination parameter describes the proportion of anomalies in the dataset. This parameter helps the isolation forest algorithm determine the threshold for classifying points as anomalies [24]. Experiments were conducted with different contamination levels: 0.1, 0.2, 0.3, 0.4, and 0.5. If the contamination parameter value is 0.1, this means that 10 % of the dataset is classified as outliers. Table 3 shows the performance comparison results of three machine learning algorithms, i.e. XGBoost, KNN, and ANN, in predicting missing values in well-log data with different contamination levels, from 0.1 to 0.5. Each algorithm is evaluated based on two main metrics, which are the MAPE and RMSE.

The experimental results show that the best model performance, as indicated by the MAPE and RMSE evaluation values, is achieved with the contaminant level set at 0.4. A contaminant level of 0.4 generally shows better performance across various features, with lower MAPE and RMSE values compared to contaminant levels of 0.1 and 0.5. This shows that using 40 % of the data as the proportion

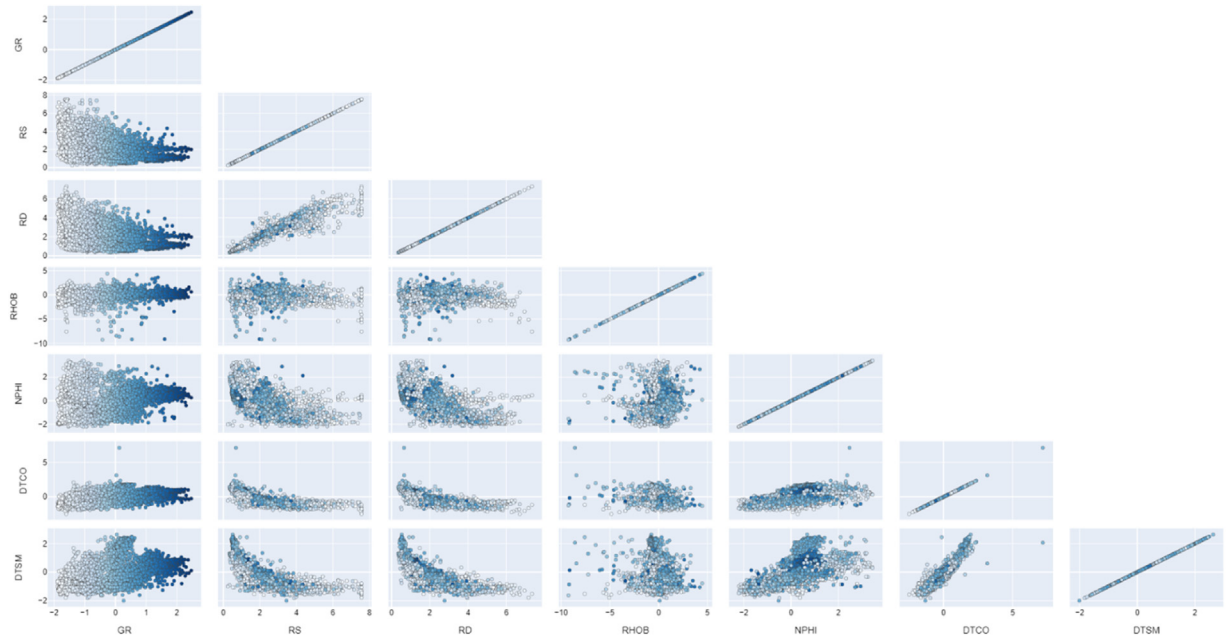

Fig. 4. Data distribution after feature scaling.

of anomalies results in the best log feature prediction performance for the dataset used in this research, as shown in Table 3, making the contaminant parameter value of 0.4 the most optimal choice for this dataset.

After the outlier handling phase using the isolation forest method, the cleaned data are ready for the next phase: training with machine learning and deep learning algorithms. This process will involve using the XGBoost, KNN, and ANN algorithms to build an accurate prediction model based on the processed well-log data and evaluating each algorithm's prediction performance.

Hyperparameter tuning results using grid-search cross-validation

Following the outlier removal process using the isolation forest technique, the refined dataset is prepared for the next phase: training with machine learning and deep learning models. This stage will utilize the XGBoost, KNN, and ANN algorithms to develop a reliable predictive model based on the processed well log data, with each model's prediction accuracy thoroughly evaluated. For two machine learning algorithms, XGBoost and KNN, hyperparameter optimisation is performed using the grid-search cross-validation technique to get the best parameters to improve the prediction accuracy (Tables 4 and 5).

After obtaining the best hyperparameters from the grid-search cross-validation results, the next step is to train the model using the XGBoost and KNN machine learning algorithms. This training aims to build a more optimal prediction model by utilising the adjusted hyperparameters, which is expected to improve the prediction accuracy on the processed well-log data.

Missing value prediction results using XGBoost

In this research, three main approaches were taken to use XGBoost to predict missing values in well-log features (GR, RHOB, NPHI, RD, RS, DTCO, and DTSM). The first approach is to apply XGBoost directly after preprocessing with the isolation forest technique to identify and handle outliers in the dataset. Furthermore, to improve the performance of the model, in addition to handling outliers using isolation forest, the application of hyperparameter tuning techniques using grid-search cross-validation is also added; it aims to find the optimal combination of parameters for the XGBoost prediction model (second approach). The last approach (third approach) is the addition of bias correction techniques after applying isolation forest and grid-search cross-validation; it aims to reduce prediction errors that may be caused by bias in the model. The results of these three approaches will be compared to see the effect of each technique in improving the prediction accuracy, which is evaluated using the MAPE and RMSE metrics. Table 6 shows the results of the XGBoost model's missing value prediction evaluation on various log features (GR, RHOB, NPHI, RD, RS, DTCO, and DTSM) under each approach. The curve visualisation of the experimental results can be seen in Fig. 5.

Based on the results presented in Table 6 and Fig. 5, there is a significant improvement in most of the features, especially the GR, DTCO, and DTSM features, which get the most significant error reduction in the third approach (the approach that involves applying isolation forest, grid search, and bias correction). For the GR feature, there was a decrease in the MAPE from 24.19 % to 12.67 %, which is a decrease of 47.6 %. The RMSE also dropped from 45.44 to 17.17, or about 62.2 %. As for the DTCO feature, the MAPE dropped from 5.95 % to 2.61 %, a decrease of 56.1 %, while the RMSE dropped from 6.58 to 3.32, or about 49.5 %. The DTSM feature

Table 3
Performance comparison of machine learning algorithms for log feature prediction with varying contamination levels.

Log Feature	Contamination	XGBoost		KNN		ANN	
		MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
GR	0.1	16.1	22.9	16.2	24.7	15.4	24.5
	0.2	16.1	21.2	15.3	22.5	19.4	25.3
	0.3	13.2	18.2	15.2	22.0	16.3	21.9
	0.4	**12.7**	**17.2**	**14.4**	**21.2**	**10.6**	**16.9**
	0.5	14.6	20.3	13.9	18.3	12.8	17.4
RHOB	0.1	1.5	0.1	1.8	0.1	1.7	0.1
	0.2	1.6	0.1	1.8	0.1	1.8	0.1
	0.3	1.4	0.1	1.6	0.1	1.7	0.1
	0.4	**1.4**	**0.1**	**1.6**	**0.1**	**1.5**	**0.1**
	0.5	1.5	0.1	1.5	0.1	2.0	0.1
NPHI	0.1	18.6	6.0	21.6	6.9	14.3	4.8
	0.2	13.2	4.4	10.9	4.0	10.5	3.5
	0.3	10.1	3.6	8.6	3.1	10.4	3.5
	0.4	**9.4**	**3.3**	**9.9**	**3.4**	**10.3**	**3.5**
	0.5	9.8	3.5	9.5	3.3	8.9	3.1
DTSM	0.1	5.1	12.2	5.6	13.0	5.0	11.9
	0.2	5.2	12.4	5.8	13.3	4.6	10.9
	0.3	5.0	11.5	4.8	11.5	4.5	10.4
	0.4	**4.3**	**9.9**	**5.3**	**12.0**	**4.7**	**10.9**
	0.5	4.3	10.4	5.4	12.1	4.9	11.5
DTCO	0.1	3.3	4.4	3.1	3.7	3.0	3.9
	0.2	3.2	4.2	3.0	3.6	2.5	3.1
	0.3	3.0	3.7	2.9	3.5	2.8	3.4
	0.4	**2.6**	**3.3**	**2.9**	**3.4**	**2.8**	**3.4**
	0.5	3.0	3.8	2.8	3.4	2.8	3.5
RS	0.1	52.0	1.4	16.2	0.5	22.0	0.8
	0.2	43.1	1.4	18.7	0.7	10.9	0.7
	0.3	56.8	1.9	14.6	0.7	8.0	0.7
	0.4	**19.5**	**1.0**	**18.2**	**0.7**	**11.5**	**0.7**
	0.5	44.1	1.3	14.2	0.7	13.2	0.7
RD	0.1	83.9	2.9	15.9	0.7	14.3	0.6
	0.2	14.7	0.5	15.8	0.7	7.9	0.5
	0.3	19.0	0.8	13.8	0.6	7.5	0.6
	0.4	**27.2**	**1.1**	**14.1**	**0.6**	**8.0**	**0.7**
	0.5	29.3	0.8	16.0	0.6	11.0	0.6

Table 4
Hyperparameter optimisation results of XGBoost algorithm with grid-search cross-validation.

Parameter	Hyperparameters						
	GR	RHOB	NPHI	RD	RS	DTCO	DTSM
n_estimator	300	100	100	100	100	100	100
max_depth	6	6	2	2	2	6	6
eta	0.01	0.1	0.1	0.1	0.1	0.1	0.1

Table 5
Hyperparameter optimisation results of KNN algorithm with grid-search cross-validation.

Log Feature	Hyperparameters		
	Metric	n_neighbours	Weights
GR	Manhattan	7	distance
RHOB	Manhattan	11	distance
NPHI	Manhattan	11	distance
RD	Manhattan	9	distance
RS	Manhattan	7	distance
DTCO	Manhattan	7	distance
DTSM	Manhattan	9	distance

Table 6
Comparative evaluation of XGBoost prediction result metrics.

Feature Log Prediction	1st Approach		2nd Approach		3rd Approach	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
GR	24.19	45.44	24.04	41.39	**12.67**	**17.17**
RHOB	1.90	0.60	1.59	0.05	**1.44**	**0.05**
NPHI	17.08	5.70	22.88	7.10	**9.35**	**3.31**
RD	**18.19**	**0.93**	20.96	1.11	27.2	1.05
RS	31.98	1.59	24.98	1.03	**19.49**	**1.00**
DTCO	5.95	6.58	5.82	6.38	**2.61**	**3.32**
DTSM	9.79	19.2	9.03	17.87	**4.29**	**9.91**

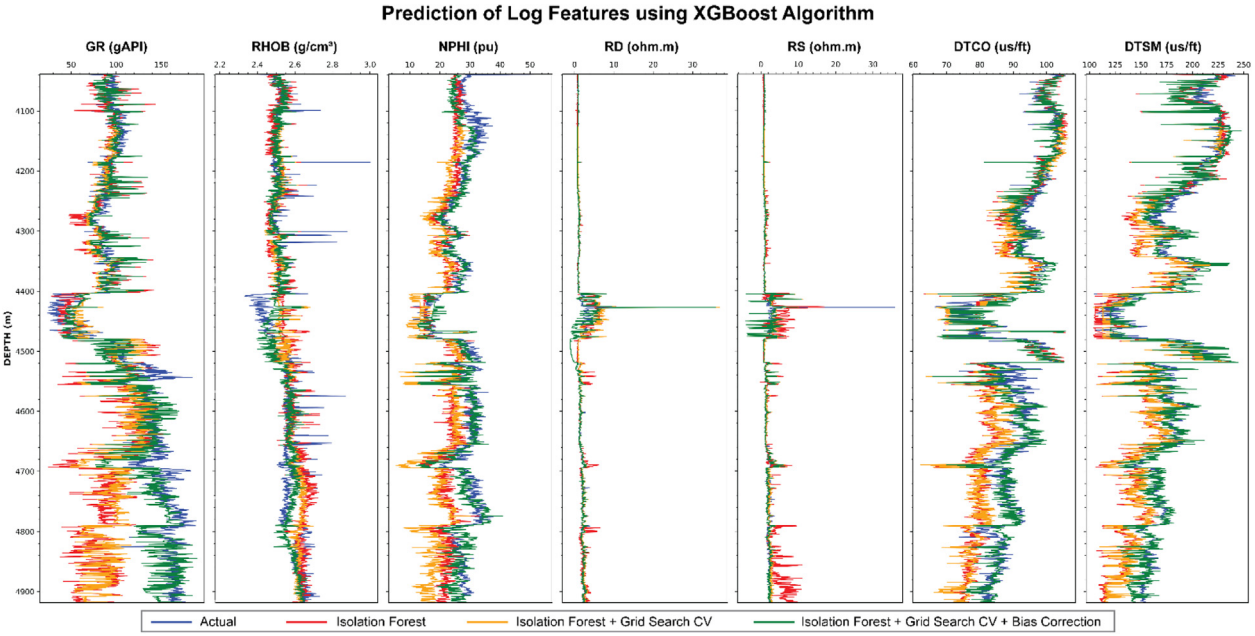

Fig. 5. Comparison of predicted and actual results of log features using XGBoost algorithm. The blue line represents the actual value, while the prediction result using the first approach is in red, the second approach is in orange, and the third approach is in green.

also shows a significant reduction, with the MAPE dropping from 9.79 % to 4.29 %, i.e. a decrease of 56.2 %, and the RMSE dropping from 19.20 to 9.91, or about 48.4 %. This significant reduction in error shows that the methodology in the third approach is able to effectively improve the prediction performance on these features very well. The increase in error on the RD features indicates that the model may be overfitting or that the data have unique properties that need to be further addressed in developing a more optimised model. The RD features had an increased error of 49.5 % in terms of the MAPE and 12.9 % in terms of the RMSE, indicating that this approach is unable to handle the prediction of RD features.

Missing value prediction results using KNN algorithm

The approach to predicting missing values in well-log features with KNN follows the same steps as the previous approach with the XGBoost algorithm. The first approach involves applying KNN after preprocessing with isolation forest to handle outliers. The second approach, after applying isolation forest, proceeds by adding hyperparameter tuning using grid-search cross-validation to find the optimal combination of parameters, while the third approach, after applying isolation forest and grid search, proceeds by incorporating bias correction techniques to reduce the prediction error. The results of these three approaches were evaluated using the MAPE and RMSE metrics, as shown in Table 7.

Based on the experimental results in Table 7 and Fig. 6, applying KNN for missing value prediction with the three approaches improves the log features. In general, using the KNN algorithm in the third approach successfully reduced the prediction error in terms of the MAPE and RMSE for most of the log features, although some features, such as the RHOB and RD features, showed a slight increase in the RMSE. Features such as the GR, NPHI, and DTCO features significantly improved after applying the isolation forest technique, hyperparameter tuning, and bias correction (third approach). The GR feature decreased the MAPE by 42.80 % and the RMSE by 53.89 %. For the NPHI feature, a decrease in the MAPE of 38.90 % and a decrease in the RMSE of 36.79 % were achieved. For the log DTCO feature, there was a significant decrease in the MAPE of 47.17 % and a decrease in the RMSE of 43.67 %.

Table 7
Comparative evaluation of KNN prediction result metrics.

Feature Log Prediction	1st Approach		2nd Approach		3rd Approach	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
GR	25.21	45.91	24.92	44.91	**14.42**	**21.17**
RHOB	2.36	0.05	1.86	0.06	**1.64**	**0.06**
NPHI	16.12	5.3	15.98	5.27	**9.85**	**3.35**
RD	17.19	0.58	18.05	0.63	**14.10**	**0.59**
RS	25.82	0.79	27.69	0.81	**18.17**	**0.72**
DTCO	5.47	6.00	4.90	5.51	**2.89**	**3.38**
DTSM	7.56	16.22	6.75	15.17	**5.32**	**11.95**

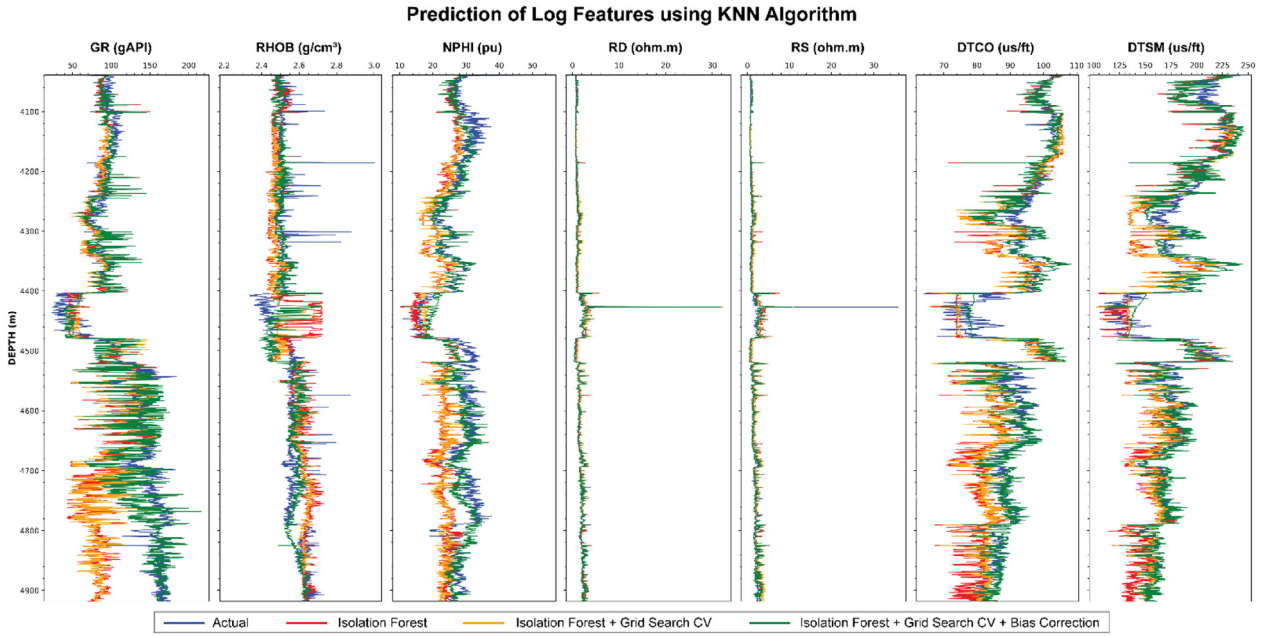

Fig. 6. Comparison of predicted and actual results of log features using KNN algorithm. The blue line represents the actual value, while the prediction result using the first approach is in red, the second approach is in orange, and the third approach is in green.

Meanwhile, features such as the RHOB feature were good enough from the beginning. However, improvements were insignificant for features such as the RS and RD features. The RS feature showed a decrease in the MAPE of 29.63 % and a slight decrease in the RMSE of 8.86 %. The MAPE value decreased by 17.98 % for the RD feature, but the RMSE value increased slightly by 1.72 %. This may be due to the complexity of the feature characteristics. Although the RS and RD features have similar data trends, the significant difference in the prediction error between the two features could be caused by several factors, one of which is the difference between the outlier distributions. Suppose that the RD data contain more outliers that the isolation forest technique does not detect well. In this case, it would be more difficult for the algorithm to predict the missing values correctly, resulting in a more significant error. In addition, there is an overfitting factor in the training data.

Missing value prediction results using ANN algorithm

In this experiment, missing values were predicted by applying the ANN prediction algorithm to two approaches. The first approach adds the isolation forest technique, and the second approach adds the isolation forest technique and bias correction at the end of the prediction process. The ANN model was designed with multiple fully connected layers and one flattened layer, using the ReLU activation function in each dense layer except the output layer. The parameters used in each layer in this research can be seen in Table 8, which includes the weight and bias parameters used in the training process to build the log feature missing value prediction model. The table shows the order of the layers in the ANN model, the type of layer, the number of neurons, the activation function, the input shape, the output shape, and the weight and bias parameters for each layer. The total number of parameters in the model is 132,481, which indicates the complexity and capacity of the model to capture patterns/trends in the data.

In general, after applying isolation forest and bias correction, the accuracy of the prediction model increased significantly (approach 2) compared to approach 1. This is indicated by errors on various logs showing a significant decrease, especially for the

Table 8
Architectural summary of ANN model layers and parameters.

Layer	Type of layer	Number of Neurons	Activation Function	Input Shape	Output Shape	Number of Parameters
1	Dense	128	ReLU	(6,1)	(128,)	896
2	Dense	128	ReLU	(128,)	(128,)	16,512
3	Flatten	–	–	(128,)	(128,)	0
4	Dense	128	ReLU	(128,)	(128,)	98,432
5	Dense	128	ReLU	(128,)	(128,)	16,512
6	Dense	1	–	(128,)	(1,)	129
Total Number of Parameters						**132,481**

Table 9
Comparative evaluation of ANN prediction result metrics.

Feature Log Prediction	1st Approach		2nd Approach	
	MAPE (%)	RMSE	MAPE (%)	RMSE
GR	22.2	44.81	**10.63**	**16.9**
RHOB	2.11	0.07	**1.47**	**0.05**
NPHI	26.46	7.91	**10.33**	**3.46**
RD	**7.91**	0.68	7.96	**0.67**
RS	12.46	0.69	**11.46**	**0.66**
DTCO	7.43	8.08	**2.77**	**3.4**
DTSM	10.32	19.46	**4.74**	**10.94**

Table 10
Comparative evaluation of the prediction result metrics of each machine learning model.

Feature Log Prediction	XGBoost		KNN		ANN	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
GR	12.67	17.17	14.42	21.17	10.63	16.19
RHOB	1.44	0.05	1.64	0.06	1.47	0.05
NPHI	9.35	3.31	9.85	3.35	10.33	3.46
RD	27.20	1.05	14.10	0.59	7.96	0.67
RS	19.49	1.00	18.17	0.72	11.46	0.66
DTCO	2.61	3.32	2.89	3.38	2.77	3.40
DTSM	4.29	9.91	5.32	11.95	4.74	10.94

DTCO and NPHI features, which showed the largest error reduction. The DTCO feature is the log feature with the most significant increase in accuracy, as shown by a decrease in the MAPE of 62.72 % and a decrease in the RMSE of 57.92 %. The NPHI feature also experienced a significant decrease, with the MAPE dropping by 60.96 % and the RMSE dropping by 56.26 %. The curve visualisation of the experimental results can be seen in Table 9 and Fig. 7.

In this research, the prediction of missing values in log features is carried out in conditions where the missing values are set as MNAR, with a percentage of 30 % of the total data (extreme case). In general, the addition of isolation forest, grid-search cross-validation, and bias correction techniques significantly improved the accuracy of the log feature prediction model in the XGBoost and KNN machine learning algorithms. Good results were also obtained for the ANN algorithm, which applied the isolation forest technique and bias correction. Table 10 compares the results of log feature prediction using three algorithms: XGBoost, KNN, and the ANN (all prediction results using bias correction). The three algorithms have different concepts: XGBoost is tree-based learning [25], KNN is example-based learning [32], and the ANN is connectionist learning or network-based learning [16]. The evaluation is based on two metrics: the MAPE and RMSE.

Based on the evaluation metric values (MAPE and RMSE) in Table 10 and Fig. 8, the methodology applied to the XGBoost algorithm overall works best for the case of log feature prediction compared to other algorithms. This is indicated by the fact that the lowest MAPE and RMSE values were achieved by XGBoost, which means that the log feature prediction results are more accurate on four features, namely the RHOB, NPHI, DTCO, and DTSM features, out of a total of seven features. On the RHOB log feature, XGBoost successfully achieved a MAPE value of 1.44 % and an RMSE of 0.05, much lower than KNN and the ANN. On the NPHI log feature, XGBoost also excels, with a MAPE value of 9.35 % and an RMSE of 3.31. XGBoost also gave the best results on the log DTCO and DTSM features. The DTCO feature has a MAPE value of 2.61 % and an RMSE of 3.32, while the log DTSM feature achieves a MAPE of 4.29 % and an RMSE of 9.91 %. XGBoost is a tree-based algorithm that utilises an ensemble learning approach to improve the prediction accuracy, where each model in the ensemble learns from the previous model's error by combining many weak decision tree models into a strong model through a boosting approach. Because it works iteratively and is able to optimise the loss function

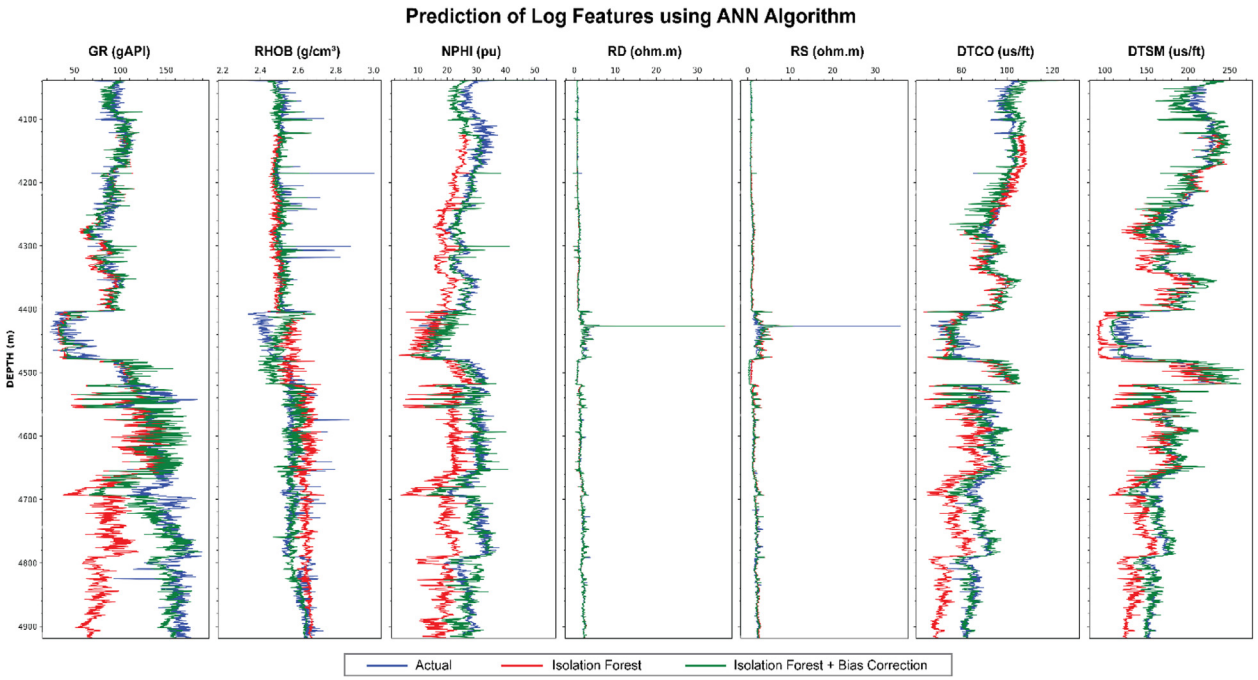

Fig. 7. Comparison of predicted and actual results of log features using ANN algorithm. The blue line represents the actual value, while the prediction result using the first approach is in red, and the second approach is in green.

effectively, XGBoost is very good at handling complex data that have non-linear patterns [45–47]. This makes XGBoost's prediction results superior to those of single models such as KNN or the ANN.

The RHOB feature is the most stable, with relatively strong predictive performance across all three algorithms, achieving a MAPE value below 2% and an RMSE close to zero. The RHOB log feature represents the density of rocks and fluids within pore spaces, where the physical parameters tend to remain stable with smaller fluctuations compared to other log features [48]. Density data is less affected by extreme environmental changes compared to resistivity, making it easier for machine learning algorithms to predict [49]. In contrast, features like RD, which exhibit higher variability and are more influenced by fluid conditions and rock heterogeneity, have proven to be more challenging for the three algorithms to predict [50,51]. This underscores the robustness of RHOB as a reliable feature for predictive modeling in machine learning applications.

The RHOB feature is the most stable, with relatively strong predictive performance across all three algorithms, achieving a MAPE value below 2% and an RMSE close to zero. The RHOB log feature represents the density of rocks and fluids within pore spaces, where the physical parameters tend to remain stable with smaller fluctuations compared to other log features . Density data is less affected by extreme environmental changes compared to resistivity, making it easier for machine learning algorithms to predict [49]. In contrast, features like RD, which exhibit higher variability and are more influenced by fluid conditions and rock heterogeneity, have proven to be more challenging for the three algorithms to predict [50,51]. This underscores the robustness of RHOB as a reliable feature for predictive modeling in machine learning applications.

Limitations

While the ANN algorithm is quite robust in handling non-linear patterns and complex data, it is highly dependent on the amount of data available to achieve an optimal performance. It is more prone to overfitting than XGBoost. Overall, the ANN performs relatively well on features that have more regular and easy-to-learn patterns, as well as features that have non-linear complexity, provided that the processed data have undergone a good preprocessing stage, such as forest isolation and bias correction, which is demonstrated by the GR, RS, and RD features. For the log GR feature, the ANN produces a MAPE of 10.63 % and an RMSE of 16.19. On the RS feature, the ANN produced a MAPE of 11.46 % and an RMSE of 0.66. Although XGBoost is superior in terms of the MAPE, the ANN provides a lower RMSE value than the other two algorithms, indicating that the ANN is more effective in minimising the prediction error for RS features. Therefore, the ANN performs relatively well in predicting log features with high heterogeneity, such as the GR, RS, and RD features.

Meanwhile, KNN did not show any significant advantage over the other two algorithms, making it less efficient and more susceptible to missing data in the case of MNAR. KNN is considered one of the simplest non-parametric classification algorithms; it becomes less effective on this dataset with an extreme number of missing values (30 %). In addition, the performance of KNN is highly dependent on the selection of the k value (number of 'neighbours') and the distance metric used to measure the similarity

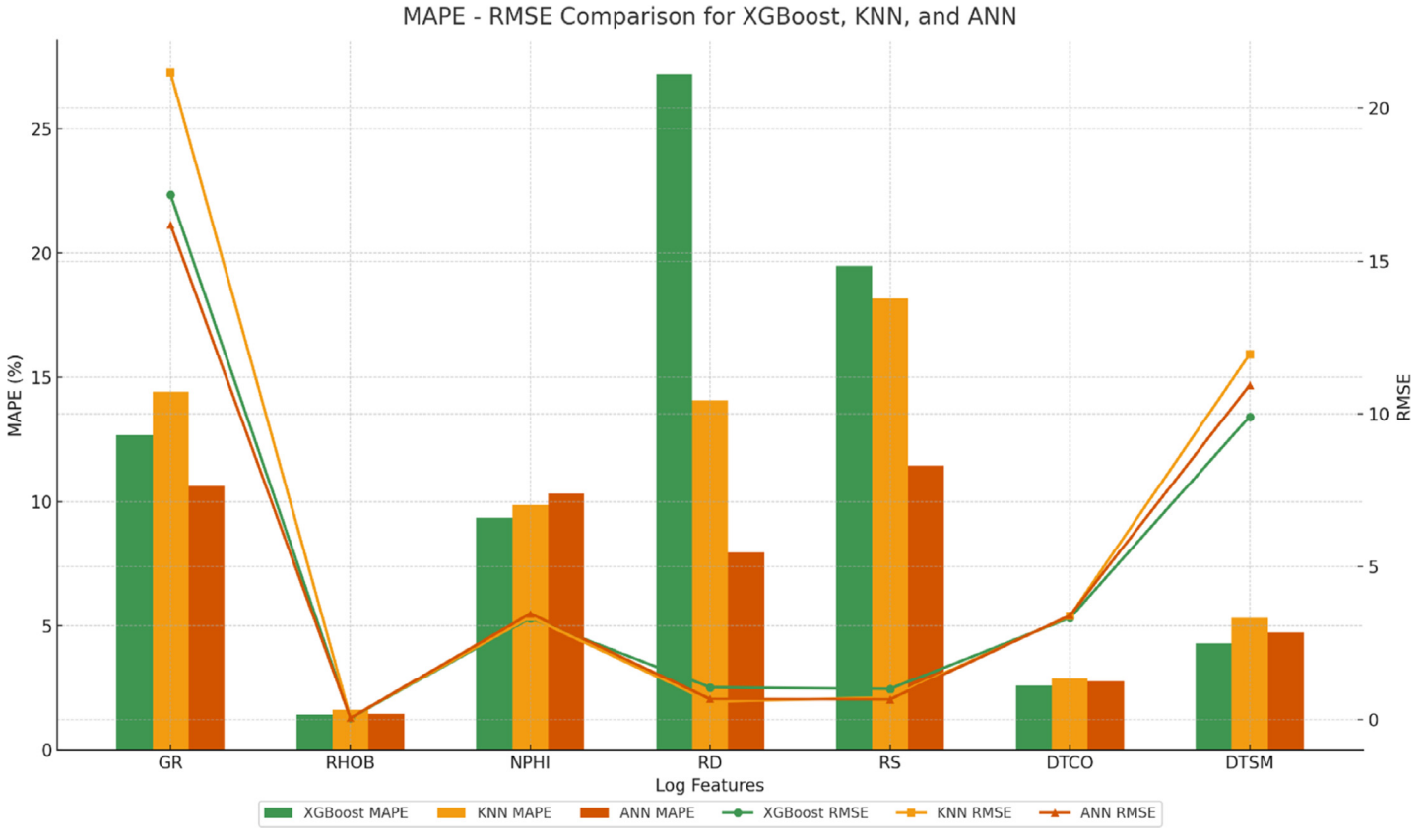

Fig. 8. Comparing the performance of predicted and actual log features using several machine learning models.

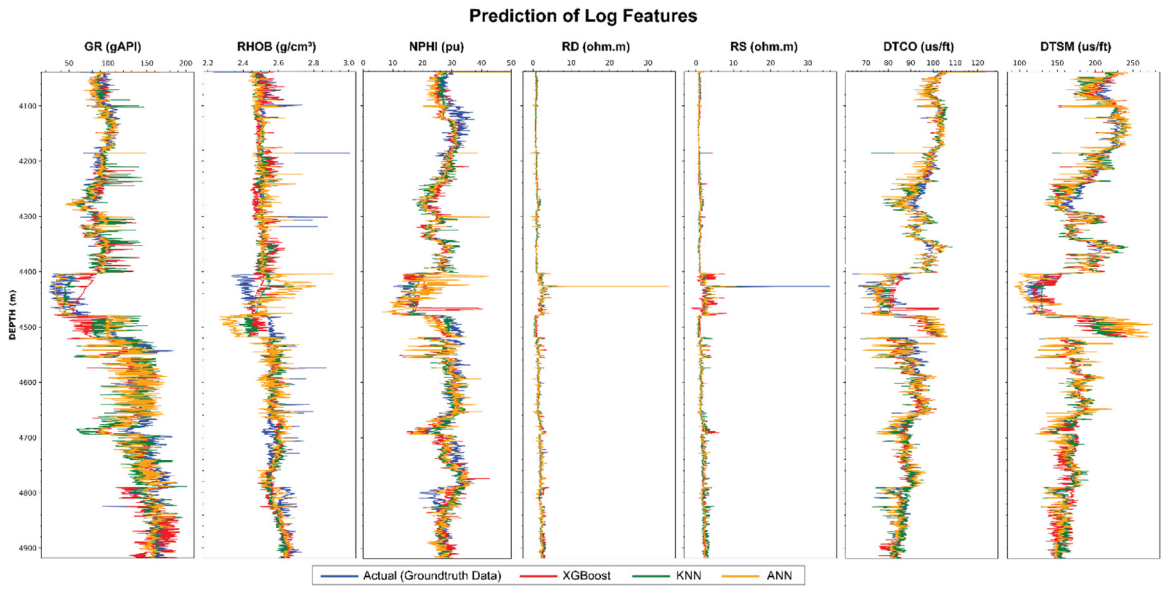

Fig. 9. Comparing the prediction performances of several machine learning models.

between data. Thus, when data are missing non-randomly, the relationship between data points becomes more difficult to map. This causes the performance of KNN to degrade drastically, especially on well-log features that have high fluctuations, such as the GR or RD features.

From the log feature prediction results shown in Table 10, Fig. 8, and 9, the RHOB log features show relatively good, consistent prediction results with very low error rates (MAPE below 2 % and RMSE close to zero) for all three algorithms (XGBoost, KNN, and ANN). This indicates that the RHOB data have a simpler pattern or are more easily recognised by the three models. The RHOB log feature describes the density of the rock. Geophysically, the RHOB feature is a relatively stable physical parameter and has lower fluctuations compared to other features. Density data are not affected by extreme environmental variations as much as resistivity, which makes the prediction of density data easier for all algorithms. Besides the RHOB feature, other log features that are more stable and easier to predict for all three algorithms are the DTCO and DTSM features, as their physical characteristics tend to be consistent and less affected by changing environmental conditions.

The prediction results for the RD (deep resistivity) feature are relatively consistent across the three algorithms; these predictions could be more optimal. The RD feature describes deeper formation resistivity, which is influenced by the fluid type and formation composition at greater depths. Deep resistivity has greater variation than shallow resistivity (RS). It is more difficult to predict due to more complex changes in fluid conditions in the subsurface formation and rock heterogeneity, as well as MNAR data conditions that further deteriorate the accuracy of the prediction model. Algorithms such as XGBoost, KNN, and ANNs need help to capture these highly non-linear and inconsistent patterns, especially in situations where significant amounts of data are missing.

Overall, MNAR data have a major impact on the model prediction results because the missing values depend on the variable itself or other variables, thus creating a more complicated pattern of missing data. As a result, estimating missing values becomes more difficult compared to the missing completely at random (MCAR) or missing at random (MAR) cases. For example, in resistivity log features such as the RS or RD features, missing data related to specific geological formation characteristics require the model to understand the context for accurate predictions. With preprocessing techniques such as isolation forest, grid-search cross-validation, and bias correction, models such as KNN or the ANN tend to perform better, especially when faced with extreme conditions such as 30 % missing data, as was the case in this study. While techniques such as isolation forest can help deal with outliers, the MNAR case still requires additional steps, such as hyperparameter tuning and bias correction, to improve the prediction accuracy. More stable features, such as the RHOB feature, show better prediction results when proper preprocessing is used. Overall, the application of preprocessing and bias correction techniques is crucial to overcome the impact of MNAR data, especially in more complex models such as the ANN and XGBoost, which show better performance when these measures are applied.

Conclusion

Overall, the RHOB feature is the most stable feature, with a relatively good prediction performance by all three algorithms, which achieved MAPE values below 2 % and an RMSE close to zero. The RHOB log feature describes the rock density, which is a physical parameter that tends to be stable, with less fluctuation than other features. Density data are less affected by extreme environmental changes compared to resistivity, making these data easier for machine learning algorithms to predict. In contrast, features such as the RD feature that have higher variations, which are more influenced by fluid conditions and rock heterogeneity, proved more difficult

to predict for the three algorithms. This is shown by the higher MAPE and RMSE values, even after the application of preprocessing and bias correction techniques, in MNAR conditions with 30 % missing data (extreme).

Based on the experimental results, several practical benefits are evident, particularly for the oil and gas exploration industry. Companies can prioritize more reliable well-log parameters, such as the RHOB feature, to simplify the analysis process. The RHOB feature demonstrates high stability, with a MAPE below 2% and an RMSE close to zero, making it an ideal choice for field measurements. Its resistance to extreme conditions and ease of prediction compared to other features, such as RD, enhance its reliability. Additionally, preprocessing techniques such as isolation forest, grid-search cross-validation, and post-training methods like bias correction enable the model to handle up to 30% missing data without requiring additional field measurements. These steps improve efficiency and are expected to significantly reduce operational costs.

The ANN and XGBoost algorithms, as more complex models, showed significant performance improvement after these measures were applied. XGBoost, in particular, was able to deliver accurate predictions even under imperfect data conditions. By implementing these methods, the oil and gas exploration industry can make more efficient and cost-effective decisions, increasing confidence in data analysis results and operational workflows.

Declaration of competing 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.

Ethics statements

This work did not involve human subjects, animal experiments data, and data collected from social media platforms.

CRediT author statement

Sherly Ardhya Garini: Conceptualization, Methodology, Investigation, Writing – Original Draft, Data Curation; **Ary Mazharuddin Shiddiqi:** Supervision, Conceptualization, Methodology, Writing – Review & Editing, Validation, Project Administration, Resources; **Widya Utama:** Supervision, Writing – Review & Editing, Validation, Resources; **Alif Nurdien Fitrah Insani:** Software, Validity Test, Data Curation.

Acknowledgments

The research was funded by the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek) in 2024, under the Doctoral Dissertation Research (PDD) scheme.

References

[1] S.A. Garini, A.M. Shiddiqi, W. Utama, O.A. Jabar, A.N.F. Insani, Enhanced lithology classification in well log data using ensemble machine learning techniques, in: IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS), 2024, pp. 1–8, doi:10.1109/AIMS61812.2024.10512485.
[2] H. Horita, Y. Kurihashi, N. Miyamori, Extraction of missing tendency using decision tree learning in business process event log, Data 5 (3) (2020) 1–12, doi:10.3390/data5030082.
[3] I.R. White, P. Royston, A.M. Wood, Multiple imputation using chained equations: Issues and guidance for practice, Stat. Med. 30 (4) (2010) 377–399, doi:10.1002/sim.4067.
[4] J. Xu, J. Liu, A profile clustering based event logs repairing approach for process mining, IEEE Access 7 (2019) 17872–17881, doi:10.1109/ACCESS.2019.2894905.
[5] S. Arciniegas-Alarcón, M. García-Peña, W.J. Krzanowski, C. Rengifo, Missing value imputation in a data matrix using the regularised singular value decomposition, MethodsX 11 (May) (2023), doi:10.1016/j.mex.2023.102289.
[6] Y. Dong, C.Y.J. Peng, Principled missing data methods for researchers (expectation maximization explained), Springerplus 2 (1) (2013) 1–17.
[7] J.L. Schafer, J.W. Graham, Missing data: Our view of the state of the art, Psychol. Methods 7 (2) (2002) 147–177, doi:10.1037/1082-989X.7.2.147.
[8] U. Iturrarán-Viveros, J.O. Parra, Artificial neural networks applied to estimate permeability, porosity and intrinsic attenuation using seismic attributes and well-log data, J. Appl. Geophys. 107 (2014) 45–54, doi:10.1016/j.jappgeo.2014.05.010.
[9] R. Zhong, R. Johnson, Z. Chen, Generating pseudo density log from drilling and logging-while-drilling data using extreme gradient boosting (XGBoost), Int. J. Coal Geol. 220 (July 2019) (2020) 103416, doi:10.1016/j.coal.2020.103416.
[10] S.R. Pride, J.G. Berryman, J.M. Harris, Seismic attenuation due to wave-induced flow, J. Geophys. Res. Solid Earth 109 (B1) (2004) 1–19, doi:10.1029/2003jb002639.
[11] D. Xing, et al., A combined method for gas-bearing layer identification in a complex sandstone reservoir, Front. Earth Sci. 10 (July) (2022) 1–9, doi:10.3389/feart.2022.942895.
[12] D. Aureli, R. Bruni, C. Daraio, Optimization methods for the imputation of missing values in educational institutions data, MethodsX 8 (2021) 101208, doi:10.1016/j.mex.2020.101208.
[13] V.A. Nordloh, A. Roubíčková, N. Brown, Machine learning for gas and oil exploration, Front. Artif. Intell. Appl. 325 (2020) 3009–3016, doi:10.3233/FAIA200476.
[14] K.M. Fouad, M.M. Ismail, A.T. Azar, M.M. Arafa, Advanced methods for missing values imputation based on similarity learning, PeerJ Comput. Sci. 7 (2021) 1–38, doi:10.7717/PEERJ-CS.619.
[15] D.L. Poston, E. Conde, Missing data and the statistical modeling of adolescent pregnancy, J. Mod. Appl. Stat. Methods 13 (2) (2014) 464–478, doi:10.22237/jmasm/1414815960.
[16] O.I. Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A.E. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, Heliyon 4 (11) (2018) e00938, doi:10.1016/j.heliyon.2018.e00938.
[17] E. Eldeeb, H. Alves, LoRaWAN-enabled smart campus: The data set and a people counter use case, IEEE Internet Things J 11 (5) (2024) 8569–8577, doi:10.1109/JIOT.2023.3320182.

[18] S.E. Awan, M. Bennamoun, F. Sohel, F. Sanfilippo, G. Dwivedi, A reinforcement learning-based approach for imputing missing data, Neural Comput. Appl. 34 (12) (2022) 9701–9716, doi:10.1007/s00521-022-06958-3.
[19] J. Huang, et al., Cross-validation based K nearest neighbor imputation for software quality datasets: An empirical study, J. Syst. Softw. 132 (2017) 226–252, doi:10.1016/j.jss.2017.07.012.
[20] F. Zhang, M. Petersen, L. Johnson, J. Hall, R.F. Palmer, S.E. O'Bryant, A machine learning-based multiple imputation method for the health and aging brain study–Health disparities, Informatics 10 (4) (2023), doi:10.3390/informatics10040077.
[21] S.W.J. Nijman, et al., Missing data is poorly handled and reported in prediction model studies using machine learning: A literature review, J. Clin. Epidemiol. 142 (2022) 218–229, doi:10.1016/j.jclinepi.2021.11.023.
[22] D.A. Anggoro, W. Supriyanti, Improving accuracy by applying Z-score normalization in linear regression and polynomial regression model for real estate data, Int. J. Emerg. Trends Eng. Res. 7 (11) (2019) 549–555, doi:10.30534/ijeter/2019/247112019.
[23] G. Hannák, G. Horváth, A. Kádár, M.D. Szalai, Bilateral-weighted online adaptive isolation forest for anomaly detection in streaming data, Stat. Anal. Data Min. 16 (3) (2023) 215–223, doi:10.1002/sam.11612.
[24] F.T. Liu, K.M. Ting, Z.H. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data 6 (1) (2012), doi:10.1145/2133360.2133363.
[25] R.Wieland Batunacun, T. Lakes, C. Nendel, Using Shapley additive explanations to interpret extreme gradient boosting predictions of grassland degradation in Xilingol, China, Geosci. Model Dev. 14 (2021) 1493–1510.
[26] J. Liu, Z. Zhou, S. Kong, Z. Ma, Application of random forest based on semi-automatic parameter adjustment for optimization of anti-breast cancer drugs, Front. Oncol. 12 (July) (2022) 1–13, doi:10.3389/fonc.2022.956705.
[27] Z. Wang, C. Zhang, Y. Ding, Applied mathematics and nonlinear sciences, Appl. Math. Nonlinear Sci. 8 (2) (2023) 3383–3392.
[28] A.Castro Garcia, S. Cheng, S.E. McGlynn, J.S. Cross, Machine learning model insights into base-catalyzed hydrothermal lignin depolymerization, ACS Omega 8 (35) (2023) 32078–32089, doi:10.1021/acsomega.3c04168.
[29] Z. Li, Z. Li, Linear programming-based scenario reduction using transportation distance, Comput. Chem. Eng. 88 (2016) 50–58, doi:10.1016/j.compchemeng.2016.02.005.
[30] T. Aljrees, Improving prediction of cervical cancer using KNN imputer and multi-model ensemble learning, PLoS One 19 (1) (2024) 1–24, doi:10.1371/journal.pone.0295632.
[31] A. Juna, et al., Water quality prediction using KNN imputer and multilayer perceptron, Water (Switzerland) 14 (17) (2022) 1–19, doi:10.3390/w14172592.
[32] Y. Gao, P. Yan, J. Pan, Nearest neighbor classification method based on the mutual information distance measure, Proc. World Congr. Intell. Control Autom. 2015-March (March) (2015) 3246–3250, doi:10.1109/WCICA.2014.7053251.
[33] S. Keskes, S. Hanini, M. Hentabli, M. Laidi, Artificial intelligence and mathematical modelling of the drying kinetics of pharmaceutical powders, Kem. u Ind. 69 (3–4) (2020) 137–152, doi:10.15255/kui.2019.038.
[34] A. Mehrabi, M. Bagheri, M.N. Bidhendi, E.B. Delijani, and M. Behnoud, "Improved porosity estimation in complex carbonate reservoirs using hybrid CRNN deep learning model," pp. 1–27, 2024.
[35] M.M. Islam, M.A. Sattar, M.F. Amin, X. Yao, K. Murase, A new adaptive merging and growing algorithm for designing artificial neural networks, IEEE Trans. Syst. Man, Cybern. Part B Cybern. 39 (3) (2009) 705–722, doi:10.1109/TSMCB.2008.2008724.
[36] N. Prakash, S.A. Manikandan, L. Govindarajan, V. Vijayagopal, Prediction of biosorption efficiency for the removal of copper(II) using artificial neural networks, J. Hazard. Mater. 152 (3) (2008) 1268–1275, doi:10.1016/j.jhazmat.2007.08.015.
[37] Z. Yu, Y. Sun, J. Zhang, Y. Zhang, Z. Liu, Gated recurrent unit neural network (GRU) based on quantile regression (QR) predicts reservoir parameters through well logging data, Front. Earth Sci. 11 (January) (2023) 1–8, doi:10.3389/feart.2023.1087385.
[38] J. Rong, et al., Machine learning method for TOC prediction: Taking Wufeng and Longmaxi shales in the Sichuan Basin, Southwest China as an example, Geofluids 2021 (2021), doi:10.1155/2021/6794213.
[39] A.K.A. Mohammed, M.K. Dhaidan, Prediction of well logs data and estimation of petrophysical parameters of Mishrif Formation, Nasiriya Field, South of Iraq using artificial neural network (ANN), Iraqi J. Sci. 64 (1) (2023) 253–268, doi:10.24996/ijs.2023.64.1.24.
[40] D. Onalo, S. Adedigba, F. Khan, L.A. James, S. Butt, Data driven model for sonic well log prediction, J. Pet. Sci. Eng. 170 (2018) 1022–1037, doi:10.1016/j.petrol.2018.06.072.
[41] Y. Ao, L. Zhu, S. Guo, Z. Yang, Computers and geosciences probabilistic logging lithology characterization with random forest probability estimation, Comput. Geosci. 144 (August) (2020) 104556, doi:10.1016/j.cageo.2020.104556.
[42] Y. Xie, C. Zhu, W. Zhou, Z. Li, X. Liu, M. Tu, Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances, J. Pet. Sci. Eng. 160 (March 2017) (2018) 182–193, doi:10.1016/j.petrol.2017.10.028.
[43] T. Auligné, A.P. McNally, D.P. Dee, Adaptive bias correction for satellite data in a numerical weather prediction system, Q. J. R. Meteorol. Soc. 133 (2007) 631–642, doi:10.1002/qj.56.
[44] S.M. Malakouti, M.B. Menhaj, A.A. Suratgar, The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction, Clean. Eng. Technol. 15 (February) (2023) 100664, doi:10.1016/j.clet.2023.100664.
[45] Y. Sakai, et al., MRI radiomic features to predict IDH1 mutation status in gliomas: A machine learning approach using gradient tree boosting, Int. J. Mol. Sci. 21 (8004) (2020) 1–22.
[46] Y. Liu, T. Liang, M. Zhang, N. Jing, Y. Xia, Q. Ding, Fault diagnosis of centrifugal chiller based on extreme gradient boosting, Buildings 14 (1835) (2024).
[47] L. Xiang, et al., Machine learning for early warning of septic shock in children with hematological malignancies accompanied by fever or neutropenia: A single center retrospective study, Front. Oncol. 11 (June) (2021) 1–9, doi:10.3389/fonc.2021.678743.
[48] M. Stadtmüller, J.A. Jarzyna, Estimation of Petrophysical Parameters of Carbonates Based on Well Logs and Laboratory Measurements, a Review, Energies 16 (10) (2023), doi:10.3390/en16104215.
[49] S. Seladji, P. Cosenza, A. Tabbagh, J. Ranger, G. Richard, The effect of compaction on soil electrical resistivity: A laboratory investigation, Eur. J. Soil Sci. 61 (6) (2010) 1043–1055, doi:10.1111/j.1365-2389.2010.01309.x.
[50] G. Aghli, R. Moussavi-Harami, R. Mohammadian, Reservoir heterogeneity and fracture parameter determination using electrical image logs and petrophysical data (a case study, carbonate Asmari Formation, Zagros Basin, SW Iran), Pet. Sci. 17 (1) (2020) 51–69, doi:10.1007/s12182-019-00413-0.
[51] P. Avseth, T. Mukerji, G. Mavko, J. Dvorkin, Rock-physics diagnostics of depositional texture, diagenetic alterations, and reservoir heterogeneity in high-porosity siliciclastic sediments and rocks - A review of selected models and suggested work flows, Geophysics 75 (5) (2010), doi:10.1190/1.3483770.