Prediction of Sonic Log Values Using a Gradient Boosting Algorithm in the 'AB' Field

Nahari Rasif[1], *Widya* Utama[1*], *Sherly* Ardhya Garini[1,2], *Rista* Fitri Indriani[1], and *Dhea* Pratama Novian Putra[1]

[1]Department of Geophysics Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[2]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

Abstract. Expanding exploration activities into new fields has significantly boosted oil production. Well logging is a key method in petroleum exploration, used to evaluate hydrocarbon zones by analyzing parameters such as gamma ray, porosity, density, resistivity, and wave propagation velocity. These parameters are displayed as vertical log curves against well depth. However, logging tools sometimes fail to capture formation parameters accurately, creating gaps in well log data. Sonic log data are particularly prone to such gaps, as they are newer and less common in older wells. To address missing data, machine learning algorithms, like gradient boosting, provide an effective solution. Gradient boosting employs an ensemble of decision trees, iteratively correcting errors to model complex data patterns. This method is especially suitable for handling the intricate nature of well log data. In this study, Python was used to develop predictions for missing data, demonstrating the capability of machine learning to enhance data reliability and improve petroleum exploration processes. By bridging data gaps, machine learning ensures more accurate assessments of hydrocarbon zones, supporting better exploration outcomes.

1 Introduction

In recent years, the search for oil and gas has become more intense as energy demands rise and available hydrocarbon reserves dwindle. Finding new reserves has become increasingly difficult, but advancements in technology, particularly logging tools and machine learning, have provided new ways to better understand geological conditions. One of the methods that has significantly evolved over time is well logging. Porosity is one of the most critical parameters in evaluating reservoirs, as it determines how much fluid can be stored in the pore spaces of rocks. Accurately assessing porosity is essential for improving reservoir production performance [1]. Among the tools used for this purpose, the sonic log is particularly valuable. It works by measuring the time a compressional sound wave takes to travel through a rock formation and return to the receiver, providing key insights into the reservoir's characteristics.

However, in practical field applications, sonic log data are often unavailable, particularly in older wells or wells under development. This is because sonic log technology is relatively

* Corresponding author: widya@geofisika.its.ac.id

new. As a result, estimating sonic log values becomes necessary to fill these data gaps. This study aims to predict sonic log values to support the calculation of other parameters and supplement the existing well log data. The estimation process employs the gradient boosting algorithm, a machine learning technique designed to handle complex data [2]. This research will focus on developing a machine learning model to predict sonic log values, analyzing the effectiveness of the gradient boosting algorithm, and evaluating the impact of training data size and features on prediction performance. The findings are expected to contribute to improving data reliability in oil and gas exploration [1].

2 Methodology

The research process was thoughtfully designed and organized into clear stages to ensure each step effectively contributed to achieving the study's objectives. This systematic approach was intended to maintain consistency, accuracy, and reliability throughout the analysis. The overall research process is illustrated in the workflow shown in Fig. 1.

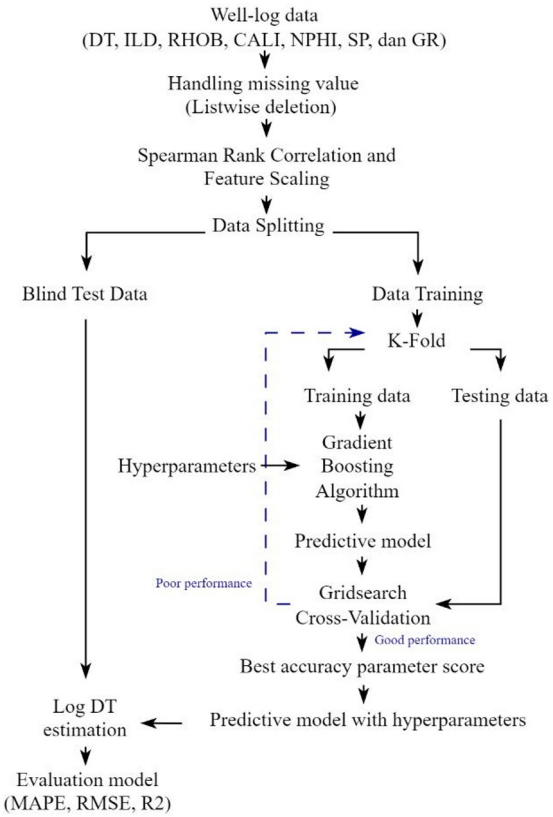

Fig. 1. Research Workflow

2.1 Pre-processing

In this study, machine learning was applied to well log data using the gradient boosting algorithm. However, before predictions could be made, several key procedures needed to be performed. These procedures included the following:

2.1.1 Listwise Deletion

For the machine learning process to function effectively, the completeness of the data is crucial. This is because the library used for the machine learning process, Scikit-Learn, requires all values in the array to be numeric. Missing or incomplete data can disrupt the process, leading to errors or unreliable results. Similarly, correlation calculations also rely on a complete data set without any missing values (NaN). One straightforward solution to address missing data is listwise deletion, which involves removing rows with empty values from the data set. In Python, missing values can be identified and visualized using the Missingno library. This library provides an effective way to assess the completeness of a data set, including well log data, and facilitates the preparation of data for machine learning applications [3, 4].

2.1.2 Spearman Rank Correlation

For the machine learning process to function effectively, the completeness of the data is crucial. Correlation measures the monotonic relationship between two variables. In statistics, the Spearman rank correlation method is commonly used to assess the strength or weakness of this relationship. This method evaluates the monotonic relationship by ranking the variable values, making it less dependent on normal distribution assumptions and less influenced by outliers [5, 6]. To calculate Spearman rank correlation, represented as "rs," certain conditions must be met. These include the presence of paired data points, variables that are either continuous (ratio) or ordinal, and the absence of NaN values in the dataset.

A variable that is perfectly correlated with another will have a Spearman rank correlation coefficient (rs) of either -1 or +1. If the rs value is close to 0, it indicates a weak relationship between the two variables. Conversely, as the rs value moves further away from 0, the strength of the relationship increases. Spearman expressed this relationship mathematically, as shown in Equation 1.

$$r_s \frac{6 \sum D^2}{n(n^2-1)} \quad (1)$$

in this formula, D represents the difference in ranks, nnn denotes the number of data pairs, and the number 6 is a constant. The Spearman rank correlation coefficient used in this study is evaluated according to the criteria outlined in Table 1.

Table 1. Interpretation of Spearman Rank Correlation Coefficient Values [7]

Correlation Coefficient Scale	Value
≥0.70	The correlation is very strong
0.40 – 0.69	Strong correlation
0.30 – 0.39	Moderate correlation
0.20 – 0.29	Weak correlation
0.01 – 0.19	The correlation is very weak

2.1.3 Feature Scaling

Raw data obtained directly from field measurements often exhibit abnormal value distributions. Additionally, the input features in the data may have different units or scales of measurement, resulting in unique value distributions across various features. When a predictive model is built using a dataset with features that have varying distributions, the model may become biased toward features with larger values or greater variances. Feature scaling addresses this issue by standardizing the data to a specific scale, ensuring that the value distributions across all features are consistent. This process helps minimize bias in the model and ensures equal consideration for all features. Feature scaling, also referred to as data normalization, can be implemented using various methods. In this study, the following method was applied:

a. Logarithmic Transformation
   Log transformation is one of the most commonly used techniques among data transformation methods. This approach converts right-skewed data distributions into a more normal distribution. If the original dataset follows a log-normal distribution, the transformed data will follow, or closely approximate, a normal distribution. In such cases, log transformation helps to eliminate or significantly reduce the skewness of the data distribution [8, 9].
b. Yeo-Johnson Transformation
   This method is among the most recent and effective techniques for transforming data into a normal distribution in general cases [10, 11]. It builds upon a prior transformation technique known as the Box-Cox transformation. The mathematical formula for the Box-Cox transformation is shown in Equation 2.

$$y_i^{(\lambda)} \begin{cases} \frac{y_i^{(\lambda)}}{\lambda} if\ \lambda \neq 0 \\ \ln(y_i)\ if\ \lambda = 0 \end{cases} \quad (2)$$

in the Box-Cox transformation, yi represents the value of the data point at the i-th position, and λ is the exponential parameter that ranges between -5 and 5. To apply this transformation, the most optimal value of λ must be determined, typically using the Maximum Likelihood Estimation (MLE) technique. However, one limitation of the Box-Cox transformation is that it cannot handle negative values in the dataset. To address this limitation, the Yeo-Johnson transformation was developed as an alternative. This method extends the capabilities of the Box-Cox transformation by allowing the inclusion of negative data values. The formula for the Yeo-Johnson transformation is provided in Equation 3.

$$y_i^{(\lambda)} \begin{cases} |(x_i+1)^{\lambda} - 1|/\lambda & if\ \lambda \neq 0 \\ \ln(x_i) + 1 & if\ \lambda = 0 \\ -|(-x_i+1)2^{-\lambda} - 1|/(2-\lambda) & if\ \lambda \neq 0 \\ -\ln(-x_i+1) & if\ \lambda = 0 \end{cases} \quad (3)$$

in the Yeo-Johnson transformation, xi represents the value of the data point at the i-th position. Unlike the Box-Cox transformation, the Yeo-Johnson transformation is capable of handling negative values for xi, making it more versatile in its application [12].

2.2 Processing: Model Selection and Development

Once the data has undergone pre-processing and is ready for predictive modeling, the next step involves selecting an appropriate machine learning model through the learning process (model selection) and applying optimization techniques to ensure minimal bias and variance (model development). This stage consists of:

a. Hyperparameter Optimization
Hyperparameters are parameters that govern the learning process of a machine learning model when applied to a dataset. Selecting the correct hyperparameters for an algorithm significantly enhances its performance. Due to the wide range of possible variations, determining the optimal parameters requires specific methods. One such method is Grid Search Cross-Validation, where the model tests various parameter combinations and evaluates its performance using repeated test datasets to achieve optimal results [13].

b. Grid Search Cross-Validation
Grid Search, also referred to as exhaustive search, operates by examining every possible combination of hyperparameters. Each specified combination of hyperparameter values is tested on the model and evaluated for accuracy. To streamline the evaluation process, grid search is often combined with Cross-Validation techniques. Cross-Validation involves dividing the entire dataset into nnn equally sized groups, or folds, wherever possible. In this process, the model is trained on n-1 folds while the remaining fold is used as validation data. This process is repeated until each fold has been used as validation data exactly once. This technique, commonly known as K-Fold cross-validation, ensures a thorough and reliable evaluation of the model's performance [14, 15].

2.3 Model Evaluation

In this study, the magnitude of error is assessed using several evaluation models, including score metrics such as the R^2 (correlation coefficient) and error metrics like MAPE (Mean Absolute Percentage Error) and RMSE (Root Mean Squared Error). Using these three metrics together provides a more balanced assessment, ensuring that the model is not only accurate in relative terms but also maintains low absolute errors while effectively capturing relationships within the data [16]. The score metric involves R^2, which measures how well the predicted data points align with the actual data. When the predicted and actual data points lie perfectly on a diagonal linear line, the R^2 value indicates perfect accuracy, represented by a value of 1. The R^2 value ranges from 0 to 1 and is independent of the dataset size [17]. The formula for calculating R^2 is provided in Equation 4.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y-\bar{y})^2}{\sum_{i=1}^{n}(y_i-\bar{y}_i)^2} \quad (4)$$

where $\hat{y}i$ represents the predicted data point or the point on the regression line, yi is the actual data at the i-th point, and $\bar{y}$ is the mean of all actual values. For the MAPE calculation, the result is expressed as a percentage, calculated using the formula provided in Equation 5 [18].

$$MAPE = \sum_{i=1}^{n}\left|\frac{f_i-a_i}{a_i}\right| \times 100\% \quad (5)$$

where n is the total number of data points, fi represents the predicted value for the iii-th data point, and ai denotes the actual value for the i-th data point. MAPE calculations provide an easily interpretable error value. To validate the error calculation further, RMSE is used as an additional error metric, defined by the formula presented in Equation 6 [19].

$$RMSE = \sqrt{\sum \frac{(\hat{y}-y_i)^2}{n}} \quad (6)$$

3 Results and Discussion

This section presents the findings of the study and the interpretation of the results, focusing on the key steps and outcomes of the machine learning process, from data preparation to model evaluation. Each stage is discussed in detail to provide insights into the methods and their effectiveness.

3.1 Data Completeness

The pre-processing stage begins with assessing the completeness of the data. This process is carried out in Python using the Pandas library to display dataset values and the Missingno library for visualization. Ensuring data completeness is a crucial step in preparing for the machine learning process.

	DEPT	CALI	GR	SP	ILD	NPHI	RHOB	DT
0	259.610	NaN	NaN	6.6307	NaN	NaN	NaN	NaN
1	259.762	NaN	NaN	3.1519	NaN	NaN	NaN	NaN
2	259.914	NaN	NaN	-10.8877	NaN	NaN	NaN	NaN
3	260.067	NaN	NaN	-27.0548	NaN	NaN	NaN	NaN
4	260.219	NaN	NaN	-33.8752	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
18114	3020.183	8.0364	NaN	NaN	NaN	NaN	NaN	NaN
18115	3020.336	8.0379	NaN	NaN	NaN	NaN	NaN	NaN
18116	3020.488	8.0394	NaN	NaN	NaN	NaN	NaN	NaN
18117	3020.640	8.0409	NaN	NaN	NaN	NaN	NaN	NaN
18118	3020.793	8.0423	NaN	NaN	NaN	NaN	NaN	NaN

Fig. 2. Data Point for Well A-050 (Training Data)

As shown in Fig. 2., the completeness of the data for one of the training datasets, specifically well A-050, is uneven. Parameters such as CALI, GR, ILD, NPHI, RHOB, and DT lack data at the initial depth of the well, while the SP parameter shows missing data at the final depth. Visualizing the completeness of the data, as illustrated in Fig. 3., aids in interpreting the extent of missing values. The loss of data in well logs can occur due to equipment malfunctions during field measurements or the high costs associated with obtaining measurements, which often result in data being collected only in specific, targeted zones [20, 21].

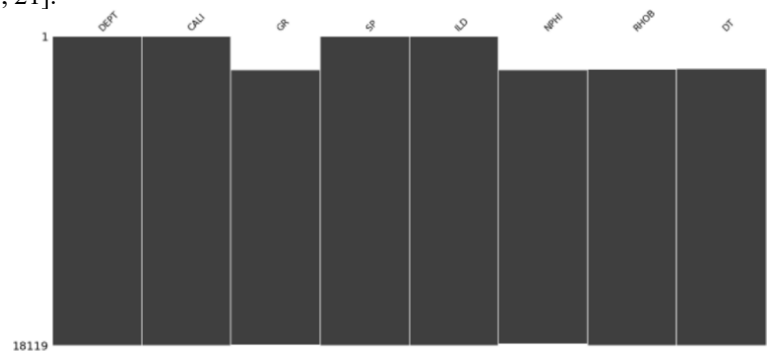

Fig. 3. Visualization of Data Completeness for Well A-050 (Training Dataset)

One solution to address missing values in datasets is the listwise deletion technique [22, 23]. This method eliminates rows containing NaN values, ensuring that the remaining dataset is entirely numeric and suitable for machine learning processes. The impact of applying the listwise deletion technique is shown in Fig. 4., where a reduction of 2119 data rows is observed. This reduction also applies to other datasets containing missing values, demonstrating the technique's effectiveness in handling incomplete data.

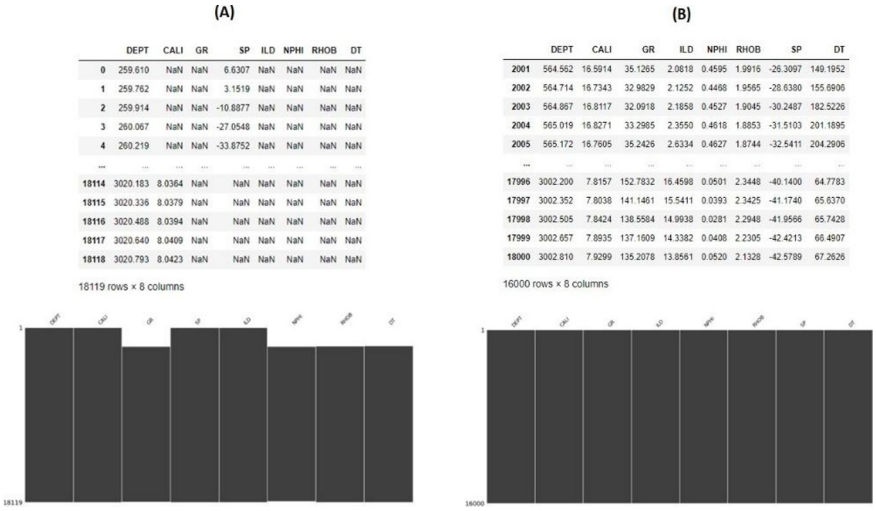

Fig.4. (A) Before and, (B) After Listwise Deletion (Well A-050)

3.2 Spearman Rank Correlation for Well Log Parameters

The next stage of pre-processing involves correlation analysis for each well log parameter. The Spearman rank correlation is used, adhering to the assumptions that the data is continuous, free of NaN values, and paired appropriately (see Fig. 6.). This analysis includes training wells with parameters such as CALI, GR, SP, ILD, NPHI, RHOB, and DT, aiming to evaluate the influence of these parameters on the DT parameter. The results of the analysis are presented as a heat map in Fig. 5., while the highest-ranked correlations between the DT parameter and other parameters are summarized in Table 2.

Table 2. Interpretation of Spearman Rank Correlation Coefficient Values [7]

Rank	Parameter Logs	Correlation Coefficient	Value
1	ILD	-0.81	The correlation is very strong
2	RHOB	-0.64	Strong correlation
3	CALI	0.578	Strong correlation
4	NPHI	0.449	Strong correlation
5	SP	-0.033	The correlation is very weak
6	GR	-0.028	The correlation is very weak

As shown in Fig. 5. and Table 2, the results of the Spearman rank correlation analysis reveal that the most influential well log parameter for DT is ILD, with a correlation value of -0.81, indicating a very strong negative relationship with the DT parameter. This strong negative correlation arises from the working principles of the DT log, which measures the propagation of sound waves in a medium, and the resistivity log, which evaluates the propagation of electrical currents. When a medium has high resistivity, it often indicates high porosity, suggesting that the rock is brittle or not compact. This brittleness and porosity impede the propagation of sound waves, resulting in a negative correlation between the ILD and DT parameters. Specifically, higher resistivity values correspond to lower values of sound wave travel time. In contrast, the low correlation between the DT and GR parameters stems from the principle behind the GR log, which measures the presence of natural radioactive elements in rocks. This measurement is less effective at indicating the compactness or brittleness of rocks. Porous rocks, which typically exhibit low gamma ray values, may sometimes contain uranium-rich water, potassium feldspar, or other elements that result in unexpectedly high gamma ray values, thus reducing the correlation with DT [24, 25]. This correlation analysis is crucial for selecting parameters for machine learning training to predict target values. Parameters with higher correlations improve the model's ability to learn and predict accurately, while those with low correlations can introduce noise, reducing accuracy. This study also examines the impact of the selected training parameters on the predictive performance of the machine learning model.

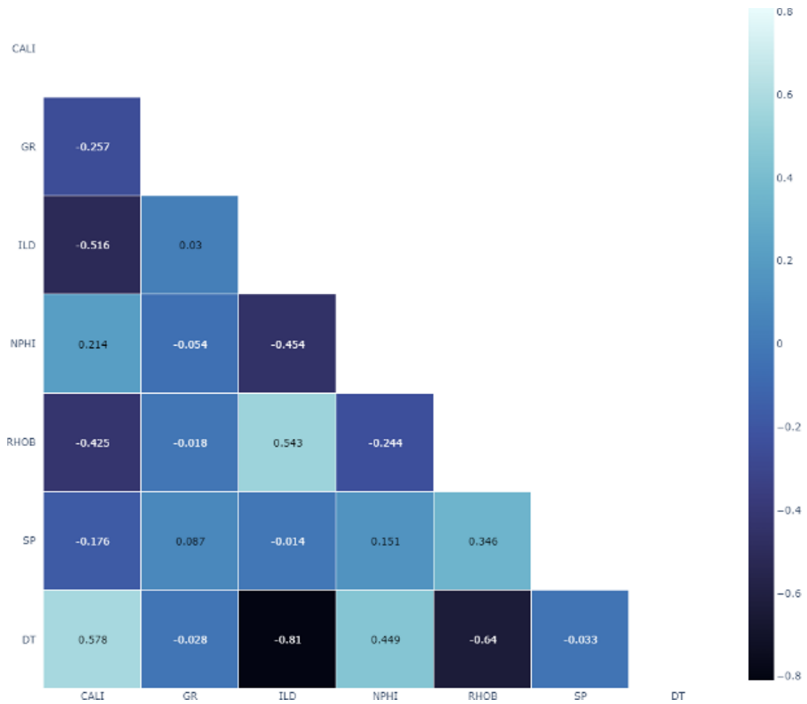

Fig.5. Spearman Rank Correlation for Training Data

3.3 Spearman Rank Correlation for Well Log Parameters

Machine learning performs better at predicting values when the data being analyzed follows a normal or Gaussian distribution [26]. This is because such data minimizes bias toward any particular category, as the mean, mode, and median share the same value [26]. However, data

collected directly from the field often exhibits abnormal distributions. An example of this can be seen in the distribution of well log data shown in Fig. 6. Some well log parameters display abnormal distributions or skewness toward one category due to unbalanced sample proportions. For instance, in the distribution of NPHI values, the majority of data points are concentrated at lower values. This skewness affects the machine learning model by biasing its predictions toward the dominant distribution of the NPHI parameter. Similarly, the GR, ILD, and NPHI parameters have distributions skewed to the right, with a tail extending to the left, indicating that most of their values are small. Conversely, the RHOB parameter has a distribution skewed to the left, with a tail to the right, suggesting that most RHOB values are large. Other parameters, such as SP, DT, and CALI, have distributions closer to normal. However, to improve these distributions further, feature scaling is required. In this study, feature scaling was applied by first performing a logarithmic transformation on the ILD parameter. Subsequently, the Yeo-Johnson transformation was used. The resulting distribution graphs are displayed in Fig. 7.

Notable differences were observed in the data distribution before and after feature scaling. The most significant change is that the data distribution becomes closer to a Gaussian or normal distribution following the application of feature scaling. However, the NPHI log parameter exhibits a bimodal distribution, indicating two distinct modes within the dataset. This occurs due to the extreme differences between the high and low values in the NPHI measurements. For other log parameters, such as DT, SP, RHOB, ILD, GR, and CALI, the distribution shifts closer to normal. Although the distribution is not perfectly Gaussian, the reduced bias and skewness in the data enhance the machine learning model's ability to learn effectively from the dataset.

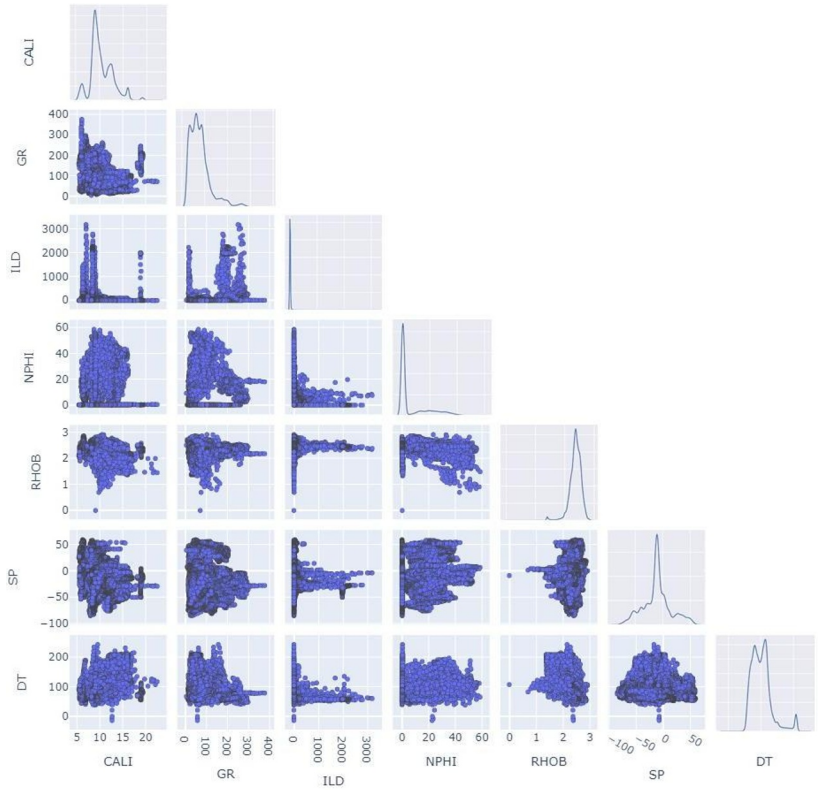

Fig.6. Data Distribution Plot Before Applying Feature Scaling

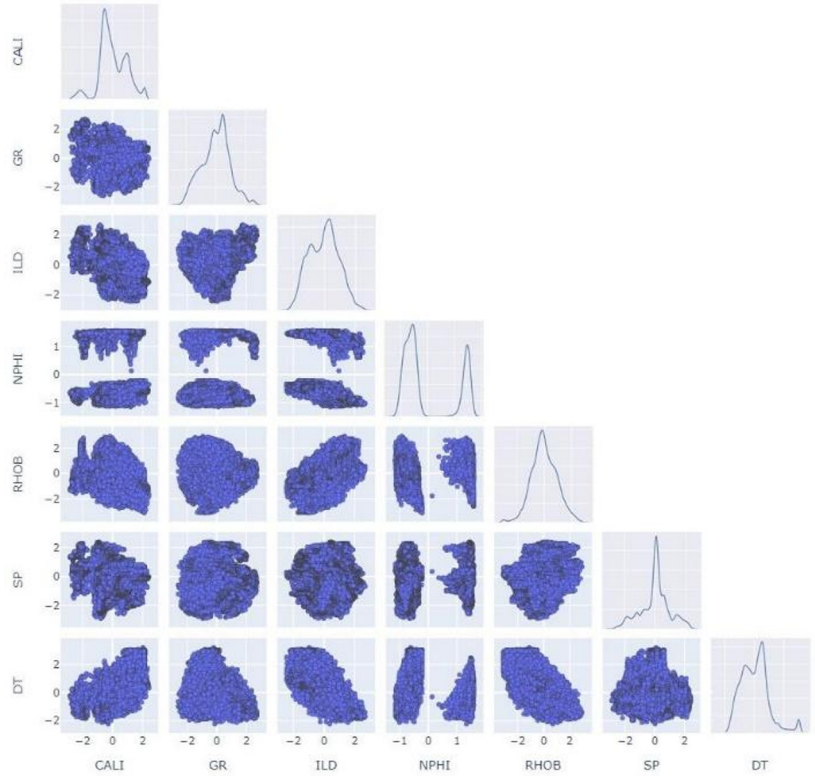

Fig.7. Data Distribution Plot After Applying Feature Scaling

3.4 Model Development

Model development aims to achieve high accuracy by optimizing hyperparameters, a process that requires considerable computational resources and time [27, 28]. Initially, the gradient boosting algorithm is trained using default parameters to establish a baseline performance. Subsequently, hyperparameter optimization is conducted to evaluate its impact on bias and variance in the regression plot compared to the default configuration [29]. The objective of this process is to obtain a regression plot with minimal bias and variance. Hyperparameter optimization is carried out using the Grid Search Cross-Validation method, which splits the dataset into three subsets: one for testing and two for validation. The optimized parameters include n_estimators (iterations), learning_rate, min_samples_leaf, min_samples_split, and max_depth. The results of the Grid Search Cross-Validation process are detailed in Table 3.

Table 3. Results of Parameter Optimization Using Grid Search Cross-Validation

No	Hyperparameter	Default	Grid Search
1	n_estimators	100	80
2	max_depth	6	3
3	min_samples_leaf	1	3
4	min_samples_split	2	2

No	Hyperparameter	Default	Grid Search
5	learning_rate	0.1	0.15

The grid search cross-validation process takes approximately 210.8 seconds (3.51 minutes) to identify and optimize the hyperparameters. In this study, the n_estimators hyperparameter, which defines the number of trees or iterations built by the model, is constrained to the grid range [80, 100]. The max_depth parameter, representing the depth of each tree, is limited to the grid values (2, 3). Additionally, the min_samples_leaf and min_samples_split hyperparameters, which determine the minimum number of leaves and sample splits, are set to grid ranges (2, 3) and (1, 2, 3), respectively. Finally, the learning_rate hyperparameter, which adjusts the contribution scale of each tree, is constrained to grid values (0.1, 0.15, 0.5, 1).

The optimal hyperparameter configuration results in a predictive model with a simpler tree structure compared to the default settings [30]. To visualize the effect of hyperparameter optimization on the model, the regression generated by gradient boosting is plotted for one of the training datasets. This allows for an evaluation of the bias and variance in the resulting model. A comparison of the regression results before and after hyperparameter optimization, using well A-051 as an example (with RHOB and DT as the x and y variables), is shown in Fig. 8.

Qualitatively, regression with default parameters demonstrates low bias but high variance, resulting in high accuracy on training data but low accuracy on test data a phenomenon commonly referred to as overfitting in machine learning. In contrast, hyperparameter optimization produces regression with both low bias and low variance, creating a model that is more flexible and better adapted to test data.

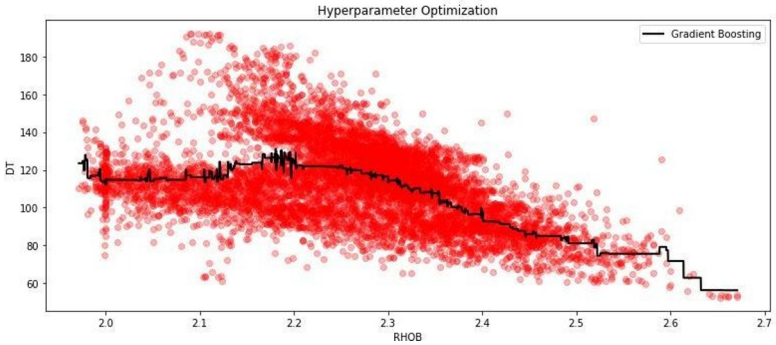

Fig.8. Results of the Regression Model Before and After Hyperparameter Optimization

3.5 Model Evaluation

The final step in assessing model performance and accuracy is the evaluation stage. This involves determining score metrics using R^2 and three error metric calculations: Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). The evaluation results are then compared between the default parameters and the optimized hyperparameters in the gradient boosting algorithm, as shown in Table 4. The prediction results for the DT log parameter are presented in Fig. 9.

Table 4 highlights the significance of hyperparameter optimization, demonstrating a 1.78% reduction in the error rate compared to the default parameters. Additionally, processing time differs between parameter settings, with the hyperparameter-optimized model requiring 25.63 seconds less than the default parameters. This reduced processing time

is attributed to the simplified tree structure of the hyperparameter-optimized model. The coefficient of determination (R^2) for predicting the DT log parameter also shows a notable improvement with hyperparameter optimization compared to the default settings. As illustrated in Fig. 9., the data distribution for the hyperparameter-optimized model exhibits a stronger correlation, closely aligning with the perfect relationship line, represented by the red diagonal line.

Table 4. Results of Parameter Optimization Using Grid Search Cross-Validation

Model	MAPE (%)	RMSE	R^2	Time (second)
Default	12.24	15.56	0.659	34.99
Hyperparameter optimization	10.46	14.01	0.7238	9.36

When assessed qualitatively and holistically, the predicted results for the DT log parameters exhibit a high degree of similarity to the actual data. This is supported by a high R^2 value and relatively low error metrics. However, at certain depths, the gradient boosting algorithm model demonstrates lower accuracy in its predictions. For instance, at the initial depth of the well, between 830m and approximately 1200m, the predicted values deviate noticeably from the actual data. Similar discrepancies are observed at depths around 1590m and 2100m.

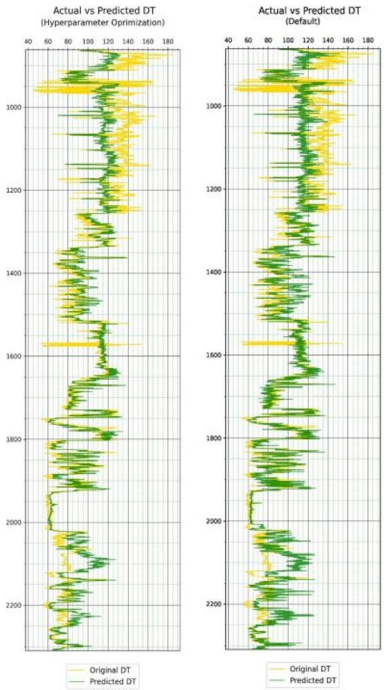

Fig.9. Comparison of Actual and Predicted DT: Hyperparameter-Optimized Model vs. Default Model

Prediction errors may arise due to several factors, including the quality of training data and an insufficient number of training samples. This issue stems from the handling of missing data using the listwise deletion technique, which can significantly reduce the dataset size. Therefore, an appropriate and optimal technique for handling missing data is required to maintain data integrity and ensure the model utilizes as much available information as

possible. The use of imputation methods, such as machine learning-based imputation [2], will be implemented in future research to minimize bias and enhance prediction accuracy.

Furthermore, suboptimal hyperparameter tuning may also affect prediction outcomes. As a result, future research will incorporate parameter sweeping techniques, such as Bayesian optimization [31] or random search [32] , to improve the model's predictive performance. Despite these discrepancies, at certain depths, particularly with the model optimized using hyperparameter tuning via the Grid Search Cross-Validation technique, the predictions align more closely with the actual data, supported by lower error values. This highlights the improved accuracy achieved through hyperparameter optimization.

3.6 Effect of Training Data Quantity and Features

This study also examines the impact of the quantity of training data and the features used in the machine learning process. The evaluation is based on score and error metrics. The effect of training data quantity on accuracy is presented in Fig. 10. and Table 5, while the impact of features is illustrated in Fig. 11. and Table 6.

In the comparison of accuracy across different amounts of training data, a general trend of increasing accuracy with more training data is observed. However, in some cases with smaller training datasets, the accuracy is higher than with larger datasets. This may occur if the data quality in certain wells is poor, and adding more data makes it more difficult for the machine learning model to identify patterns, leading to reduced accuracy. The highest error value is seen when only one well is used for training, resulting in an error of 21% and a coefficient of determination (R^2) of 0.351.

Table 5. Results of Parameter Optimization Using Grid Search Cross-Validation

Total Training Data	MAPE (%)	RMSE	R^2
14	10.46	14.01	0.7238
13	10.525	14.103	0.7214
12	10.701	14.117	0.7025
11	10.729	14.121	0.6986
10	10.877	14.123	0.6732
9	10.476	13.271	0.7012
8	10.467	13.215	0.7129
7	10.991	13.64	0.6578
6	11.612	14.161	0.6042
5	11.452	13.847	0.6264
4	11.783	14.375	0.5831
3	12.272	15.319	0.5022
2	16.423	19.63	0.4831
1	21.225	25.782	0.3512

Fig.10. Comparison of Error Values and Correlation (R^2) Across Different Training Data Quantities

Table 5. Results of Parameter Optimization Using Grid Search Cross-Validation

Number of Features	Features	MAPE (%)	RMSE	R^2
7	DT, ILD, RHOB, CALI, NPHI, SP, GR	10.46	14.01	0.7238
6	DT, ILD, RHOB, CALI, NPHI, SP	10.51	14.43	0.7068
5	DT, ILD, RHOB, CALI, NPHI	11.34	15.78	0.6493
4	DT, ILD, RHOB, CALI	11.47	16.37	0.6211
3	DT, ILD, RHOB	11.58	16.86	0.6032

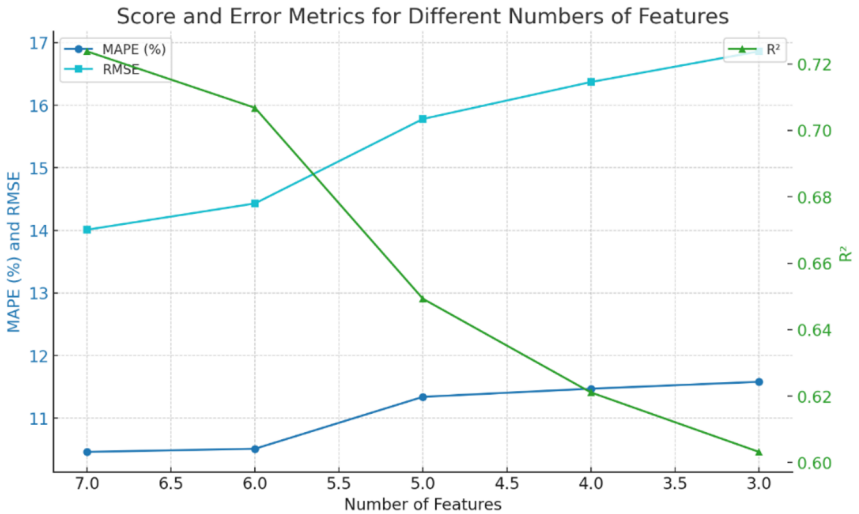

Fig.11. Comparison of Error Values and Correlation (R^2) or Different Numbers of Features

Conclusion

The prediction of sonic log values using the gradient boosting algorithm was successfully completed with relatively high accuracy. The pre-processing stages included listwise deletion, Spearman rank correlation, and feature scaling. The processing stage involved hyperparameter optimization through grid search cross-validation, yielding the following optimal values: results n_estimators=80, max_depth=3, min_samples_leaf=3, min_samples_split=2, and learning_rate=0.15. Hyperparameter selection played a crucial role in significantly reducing the error rate, with a 1.78% improvement compared to the default parameters. As a result, the model achieved a high accuracy level, with an error rate of 10.46% and an R^2 value of 0.72, compared to the default parameter's error rate of 12.24% and R^2 of 0.65. However, inaccuracies were observed at certain depths, specifically between 830m and approximately 1200m, as well as around 1590m and 2100m in the test data (well A-052). Increasing the amount of training data demonstrated a positive trend in accuracy, as did adding more features to the training data. However, in some cases, smaller training datasets yielded lower error rates compared to larger datasets. This anomaly is likely due to the varying quality of the data, which can significantly influence the predictive model's accuracy.

Acknowledgement

The authors gratefully acknowledge the financial support provided by Institut Teknologi Sepuluh Nopember through the Publication Writing and IPR Incentive Program (PPHKI) 2024.

References

1. Aminian, K., Ameri, S.: Application of artificial neural networks for reservoir characterization with limited data. J. Pet. Sci. Eng. **49**, 212 (2005).
2. Garini, S.A., Shiddiqi, A.M., Utama, W., Insani, A.N.F.: Filling-well : An effective technique to handle incomplete well-log data for lithology classification using machine learning. MethodsX. **14**, 103127 (2025).
3. Hair J, Anderson R, Babin B, Black W: Multivariate Data Analysis.pdf, (2010).
4. Hallam, A., Mukherjee, D., Chassagne, R.: Multivariate imputation via chained equations for elastic well log imputation and prediction. Appl. Comput. Geosci. **14**, 100083 (2022).
5. Myers, L., Sirois, M.J.: Spearman Correlation Coefficients, Differences between. Encycl. Stat. Sci. 1 (2005).
6. Wangwongchai, A., Waqas, M., Dechpichai, P., Hlaing, P.T., Ahmad, S., Humphries, U.W.: Imputation of missing daily rainfall data; A comparison between artificial intelligence and statistical techniques. MethodsX. **11**, 102459 (2023).
7. Leclezio, L., Jansen, A., Whittemore, V.H., De Vries, P.J.: Pilot validation of the tuberous sclerosis-associated neuropsychiatric disorders (TAND) checklist. Pediatr. Neurol. **52**, 16 (2015).
8. Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., Tu, X.M.: Log-transformation and its implications for data analysis. Shanghai Arch. Psychiatry. **26**, 105 (2014).

9. Döppel, F.A., Votsmeier, M.: Efficient neural network models of chemical kinetics using a latent asinh rate transformation. React. Chem. Eng. **8**, 2620 (2023).
10. Cai, J., Xu, X.: Bayesian analysis of mixture models with Yeo-Johnson transformation. Commun. Stat. - Theory Methods. **53**, 6600 (2024).
11. Yang, L., Nguyen-Thoi, T., Tran, T.T.: Predicting the friction angle of clays using a multi-layer perceptron neural network enhanced by yeo-johnson transformation and coral reefs optimization. J. Rock Mech. Geotech. Eng. **16**, 3982 (2024).
12. Box, G.E.P., Cox, D.R.: An Analysis of Transformations. J. R. Stat. Soc. Ser. B Stat. Methodol. **26**, 211 (1964).
13. Probst, P., Boulesteix, A.L., Bischl, B.: Tunability: Importance of hyperparameters of machine learning algorithms. J. Mach. Learn. Res. **20**, 1 (2019).
14. Phung, V.H., Rhee, E.J.: A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. Appl. Sci. **9**, (2019).
15. Zhao, Y., Zhang, W., Liu, X.: Grid search with a weighted error function: Hyper-parameter optimization for financial time series forecasting. Appl. Soft Comput. **154**, 111362 (2024).
16. Mystakidis, A., Koukaras, P., Tsalikidis, N., Ioannidis, D., Tjortjis, C.: Energy Forecasting: A Comprehensive Review of Techniques and Technologies. Energies. **17**, 1 (2024).
17. Varma, B.V., Prasad, E. V., Singha, S.: Study on predicting compressive strength of concrete using supervised machine learning techniques. Asian J. Civ. Eng. **24**, 2549 (2023).
18. Jierula, A., Wang, S., Oh, T.M., Wang, P.: Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. Appl. Sci. **11**, 1 (2021).
19. Kumar Dubey, A., Kumar, A., García-Díaz, V., Kumar Sharma, A., Kanhaiya, K.: Study and analysis of SARIMA and LSTM in forecasting time series data. Sustain. Energy Technol. Assessments. **47**, 101474 (2021).
20. Solanki, P., Baldaniya, D., Jogani, D., Chaudhary, B., Shah, M., Kshirsagar, A.: Artificial intelligence: New age of transformation in petroleum upstream. Pet. Res. **7**, 106 (2022).
21. Koroteev, D., Tekic, Z.: Artificial intelligence in oil and gas upstream: Trends, challenges, and scenarios for the future. Energy AI. **3**, 100041 (2021).
22. Alruhaymi, A.Z., Kim, C.J.: Study on the Missing Data Mechanisms and Imputation Methods. Open J. Stat. **11**, 477 (2021).
23. Woods, A.D., Gerasimova, D., Van Dusen, B., Nissen, J., Bainter, S., Uzdavines, A., Davis-Kean, P.E., Halvorson, M., King, K.M., Logan, J.A.R., Xu, M., Vasilev, M.R., Clay, J.M., Moreau, D., Joyal-Desmarais, K., Cruz, R.A., Brown, D.M.Y., Schmidt, K., Elsherif, M.M.: Best practices for addressing missing data through multiple imputation. Infant Child Dev. **33**, 1 (2024).
24. Skjeldal, M.E.: Machine Learning techniques for Prediction of Rock Properties from Reservoir Well Logs, (2021).

25. Gitau, G.: Assessment of Elemental Content and Associated Radiological Exposure in Sand Beds of Tiva River, Kitui County, (2023).

26. Lee, J., Cho, Y.: National-scale electricity peak load forecasting: Traditional, machine learning, or hybrid model? Energy. **239**, 122366 (2022).

27. Wu, J., Chen, S.P., Liu, X.Y.: Efficient hyperparameter optimization through model-based reinforcement learning. Neurocomputing. **409**, 381 (2020).

28. Yang, L., Shami, A.: On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing. **415**, 295 (2020).

29. Belete, D.M., Huchaiah, M.D.: Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. Int. J. Comput. Appl. **44**, 875 (2022).

30. Cho, H.U., Nam, Y., Choi, E.J., Choi, Y.J., Kim, H., Bae, S., Moon, J.W.: Comparative analysis of the optimized ANN, SVM, and tree ensemble models using Bayesian optimization for predicting GSHP COP. J. Build. Eng. **44**, 103411 (2021).

31. Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H., Deng, S.H.: Hyperparameter optimization for machine learning models based on Bayesian optimization. J. Electron. Sci. Technol. **17**, 26 (2019).

32. Mantovani, R.G., Rossi, A.L.D., Vanschoren, J., Bischl, B., De Carvalho, A.C.P.L.F.: Effectiveness of Random Search in SVM hyper-parameter tuning. Proc. Int. Jt. Conf. Neural Networks. **2015-Septe**, 1 (2015).