Prediction of Compressional Slowness from Conventional Well Log Data using the Gradient Boosting Algorithm

Prediction of Compressional Slowness from Conventional Well Log Data using the Gradient Boosting Algorithm

Widya Utama[1*], Eki Komara[1], Sherly Ardhya Garini[2], Nahari Rasif[1], Alif Nurdien Fitrah Insani[1], Omar Abdul Jabar[1], Yudi Rosandi[3], Abdul Hakam[4]

[1]Department of Geophysics Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[2]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[3]Department of Geophysics, Universitas Padjadjaran, Sumedang 45363, Indonesia
[4]Department of Civil Engineering, Universitas Andalas, Padang 25163, Indonesia

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

Abstract. Compressional slowness (DTCO) is the most basic parameter in geophysics, petrophysics, and geomechanics. These parameters can be obtained through the sonic log tool. However, equipment constraints, relatively new technology, and high cost of measurement make the parameters generated by sonic logs unavailable in old wells or wells being developed. Therefore, it is essential to predict sonic logs, especially in the case of compressional slowness prediction. Using machine learning, predictions can be generated by studying data on existing log wells. One of the algorithms that can produce predictions on continuous data, such as log values, is gradient boosting. MAPE and RMSE were used as evaluation metrics. The inputs used are gamma ray log data (GR), density (RHOB), porosity (NPHI), and shear slowness (DTSM). MAPE results show an error value of 12.28% with an RMSE of 10.74, indicating that the predictive model obtained has good results and performance. Using hyperparameter tuning in machine learning can reduce the error rate by 2.29% with faster processing times. In addition, it was found that the quantity of training wells can affect the resulting error value. The existence of this research can help a petrophysicist, geologist, and geophysicist characterize a reservoir with limited data. The use of this method also has the potential to be an alternative solution when sonic log measurements are expensive.

1. Introduction

Along with technological growth, machine learning is increasing. Machine learning applications can be employed across various industries, including the oil and gas sector. The factor that makes machine learning necessary to develop is that the capabilities of the technology can do activities that are impossible for humans to do. An example is predicting an event by utilizing the search for patterns from events that have occurred previously. The concept brought by machine learning, studying, and making predictions will make it easier for practitioners to achieve their goals. One of the cases in the oil and gas industry was the prediction of a petrophysical parameter to analyze the reservoir.

A well's hydrocarbon zone can be determined through petrophysical parameters. By measuring the sonic log, one of the factors used to determine reservoir quality in a field is porosity. The sonic log can describe the time and speed of compressed sound emitted by the device into the formation until the receiver captures it.

Among other petrophysical parameters, sonic logs are often used to perform petrophysical analysis and calculate well-to-seismic ties. Compressional slowness (DTCO) and shear slowness (DTSM) are fundamental seismic parameters useful in petrophysics. Both are measured via a sonic log. Sonic log transmits compression and shear waves through subsurface formations in the borehole environment to obtain matrix or fluid information [1]. Hence, this parameter becomes essential to have in every well. But sonic log data is sometimes not found in old wells. Data loss or the simple inability to take measurements at that moment both contribute to this. Sonic log tests are not only performed in old wells but also in newly created wells because they are one of the measurements that take a lot of time and money. [2]. Due to the complex subsurface stratigraphy, poor drilling conditions (washout), inadequate logging tools, relatively tricky, time-consuming, and poor subsurface correlation with offset wells, good sonic logs are scarce [3,4]. Machine learning technology may be able to solve this issue. Therefore, it is crucial to investigate and assess how machine learning is used in the oil and gas sector.

2. Methodology

Previous researchers have made predictions of log values using specific algorithms. Using the Extreme Gradient Boosting algorithm to generate density pseudo logs, for example [5], where the error results are found to be relatively low with an 5% error rate or less. Other studies used deep learning as a machine base to predict the values of density, porosity, and sonic logs with low errors [6]. As for other uses in machine learning, especially the gradient boosting algorithm, namely lithology prediction. The resulting accuracy value is 93.55% through various stages, including hyperparameter tuning and feature selection [7]. Other studies that have been done previously are also making use of supervised machine learning to forecast lithology, such as support vector machines, decision trees, multi-layer perceptrons, random forests, and extreme gradient boosting, which show accuracy values above 80% [8]. From the previous experimental results, this study tries to evaluate the accuracy and performance of gradient boosting in the case of determining the sonic log value or DTCO.

2.1. Gradient Boosting Algorithm (GBoost)

The gradient boosting algorithm is a combination of several weak learning algorithms (learners), such as a decision tree, which works by adding predictors sequentially into the predictive model and each corrects the predecessor predictive model, so that the weak algorithm becomes stronger [9,10]. The concept and tree structure of gradient boosting can be explained through a simple visualization shown in Figure 1 and 2.

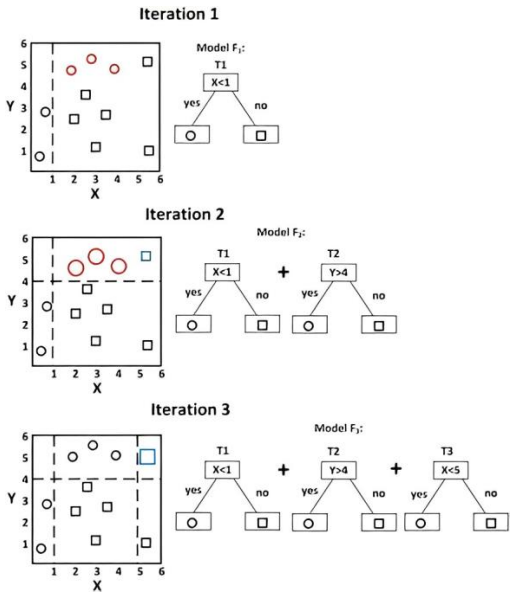

Figure 1. A simple visualization of gradient boosting algorithm [11].

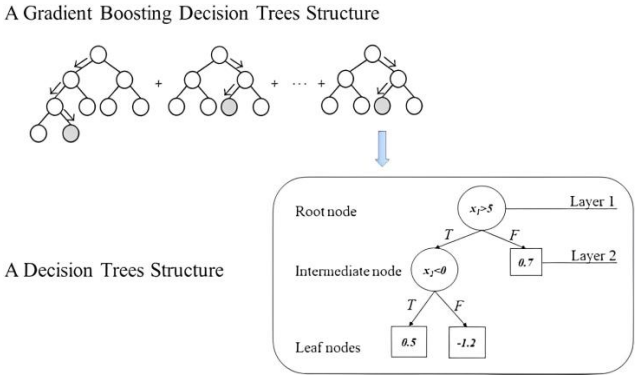

Figure 2. Gradient boosting tree structure and its parts [12].

The picture above shows the principle of the algorithm in classifying flat shapes.T1 is the simplest tree model for F1, and the symbol X with an axis less than 1 classifies it as a circle, and more than one is classified as a square. However, in the F1 model, there is still an error for classification, where the X axis is more than 1 in the actual data is a circle (marked in red). Therefore, this algorithm evaluates the prediction results by increasing the weight to incorrect ones. In the second iteration, this algorithm focuses more on the error of the 1st iteration. This way of working continues until the final iteration, in this case, the 3rd iteration, which shows the tree model T1, T2, and T3. The workings of this algorithm make it possible to generate accurate predictive values by utilizing learning from the results of errors in the previous iteration.

Table 1. Gradient Boosting algorithm procedure regression model.

Gradient Boosting	**Algorithm**
Input	Data $\{(x_i, y_i)\}_{i=1}^{n}$ and differentiable Loss Function $L(y_i, f(x))$
Step 1	Initialize model with a constant value: $f_0(x) = \underset{\gamma}{argmin}\, L(y_i, \gamma)$
Step 2	For m = 1 to M:
(A)	Compute $r_{im} = -[\frac{\delta L\,(y_i, F(x_i))}{\delta F(x_i)}]_{F(x)=F_{m-1}(x)}$ $\quad for\ i = 1, \dots, n$
(B)	Fir regression tree to the r_{im} values and create terminal regions R_{jm}, for $j = 1 \dots j_m$
(C)	For $j = 1 \dots j_m$ compute $\gamma_{jm} = \underset{\gamma}{argmin} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$
(D)	Update $F_m(x) = F_{m-1}(x) + u \sum_{j=1}^{jm} \gamma_{jm} I(x \in R_{jm})$
Output	$F_M(x)$

2.2. Metrics Evaluation

In assessing whether a result has high accuracy, it is necessary to evaluate the metric between the predicted value and the original. There are various ways to calculate metric evaluations, one of which is through a simple MAPE (Mean Absolute Percentage Error) calculation. Through MAPE, it can be seen how big the difference in the resulting values is in the form of a percentage. This equation is expressed by

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

Where n is the number of data, fi is the predicted value on the i-th data, ai is the actual data on the i-th data [13]. In order for the error calculation to be validated, RMSE is used as another error calculation, it is formulated as follows:

$$RMSE = \left(\sum_{i=1}^{n} (\frac{\hat{y} - y_i)^2}{n} \right)^{1/2}$$

Variable n is the amount of data, $\hat{y}i$ is the predicted value, and yi is the actual value. The RMSE value obtained later is in the form of the average error value generated by the predictive model [14].

3. Generation of Sonic Log Prediction

The computations are performed on a laptop with an AMD Ryzen 5 5600 (12 CPUs) 112 3.3 GHz and 8 GB of single channel RAM. The language used is Python with Scikit-learn 113 as library for making machine learning models.

3.1. Workflow of the Project

Figure 5 shows the workflow of this project. The green part is a pre-process, including checking the completeness of data and imputation, looking for the relationship of each feature, and performing feature scaling using the Yeo-Johnson transformation [15]. The red part is the model development stage, which includes separating training and test data, performing a grid search cross-validation, and tuning a hyperparameter. Then, the final stage is calculating the metric score.

3.2. Pre-processing

In the pre-processing stage, a completeness check is performed on the well-log data before the data is used to predict the target log parameters. Data will be checked and adjusted at this stage so the machine-learning process can run well. The pre-processing method used in this research is the listwise deletion method. The method has a working principle of deleting rows that contain empty values (NaN) in the dataset. So that the arranged data set is a complete data set without any empty data values or rows. That is empty. Can be seen in Figure 3 is a visualization of data completeness in well three before pre-processing and a visualization of data completeness after pre-processing in Figure 4. Tables 2 and 3 show the statistical values of well three before and after pre-processing.

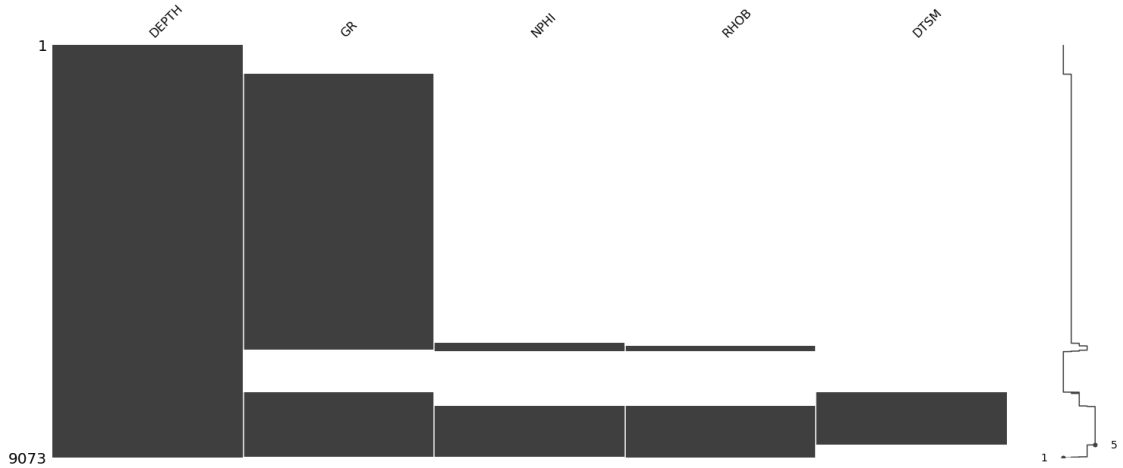

Figure 3. Visualization of completeness of data on well 3 before *pre-processing*.

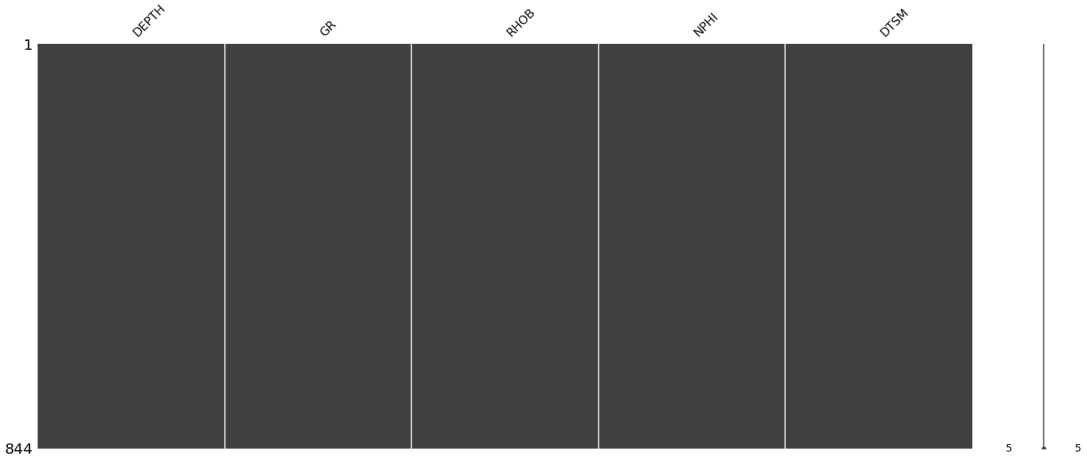

Figure 4. Visualization of completeness of data on well 3 after *pre-processing*

Table 2. Statistics of Well 3 (before pre-processing)

Description	Unit	Min	Max	Mean	STD	Count
GR	ºAPI	5.032	198.636	31. 624	31.016	7487
NPHI	g/cc	1.040	50.190	18.545	11.081	1290
RHOB	g/cc	1.442	3.035	2.534	0.160	1252
DTSM	us/ft	83.223	221.921	144.813	31.304	1156

Table 3. Statistics of Well 3 (After pre-processing)

Description	Unit	Min	Max	Mean	STD	Count
GR	ºAPI	7.134	198.636	89.101	52.871	844
NPHI	g/cc	1.300	50.190	20.536	10.549	844
RHOB	g/cc	1.442	3.036	2.564	0.173	844
DTSM	us/ft	82.223	221.921	134.370	28.940	844

3.3. Spearman Rank Correlation between Input and Output

Before using machine learning to predict sonic logs, analyzing the correlation between them and other log features is necessary. With this analysis, the fundamentals of physics will be comprehended, and feature selection will improve the accuracy of the machine learning itself. The Spearman Rank (rs) correlation method is utilized at this point, with a value ranging from -1 to +1. If rs is equal to 0, a variable does not correlate with other variables. Spearman Rank is used because of its working principle, which does not require normally distributed data and is not affected by outliers through the ranking process. [16]. Can be seen in Figure 6 shows how the relationship between the petrophysical parameters and the sonic log.

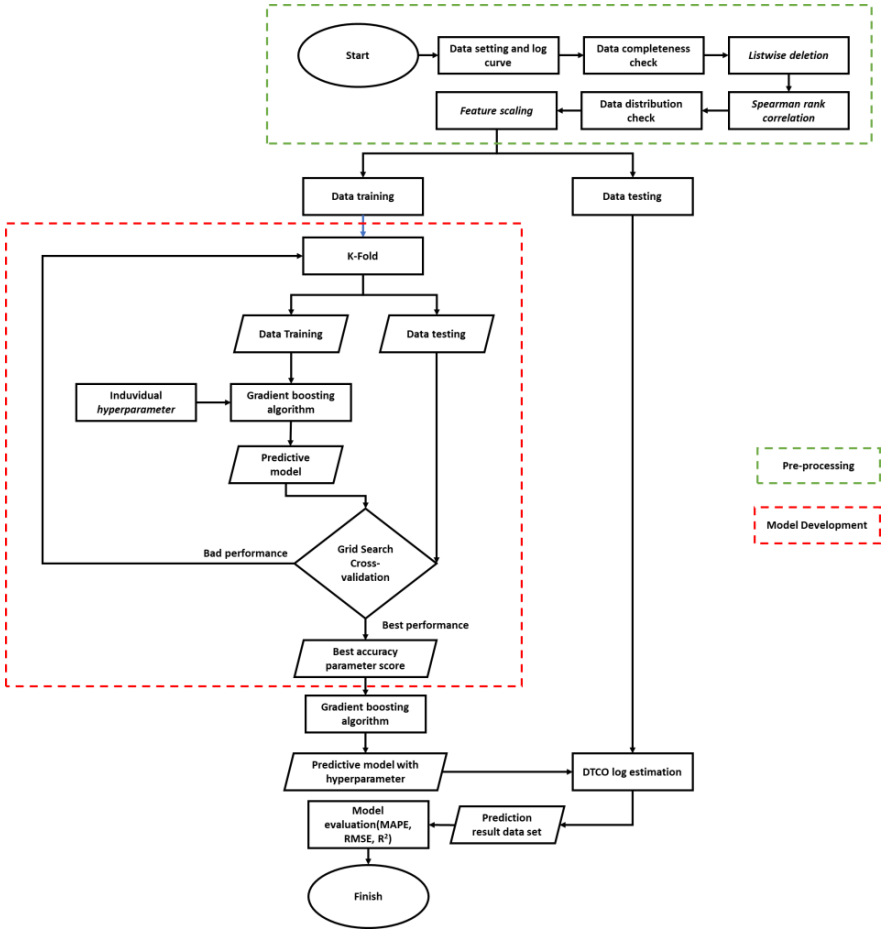

Figure 5. Workflow for the DTCO log prediction project

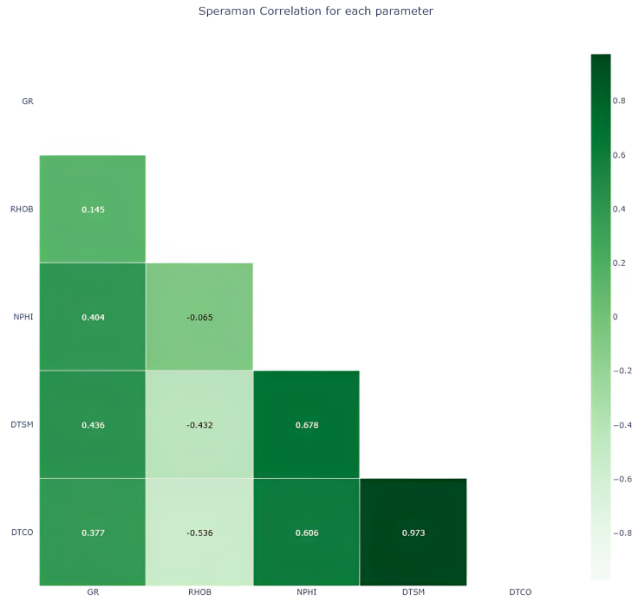

Figure 6. Relationship between input (GR, RHOB, NPHI, DTSM) and output (DTCO). (GR: gamma ray; RHOB: density; NPHI: porosity, DTSM: shear slowness; DTCO: Compressional slowness)

DTSM has a high correlation with DTCO, and this happens because both are body wave types with different directions of propagation in seismic waves. Suppose DTCO is a compressional wave whose particle motion is in the direction of the wave propagation. In that case, DTSM is a transverse wave whose particles are directed perpendicular to the wave's propagation direction. Thus, DTSM has a strong correlation with DTCO. Other parameters, such as RHOB and NPHI, correlate reasonably strongly with DTCO. The reason is that the density of rock will affect the speed of propagation of the waves produced. The higher the density of a rock, the faster the propagation. Therefore, the correlation with RHOB shows a negative value, and NPHI shows a positive value.

3.4. Training and testing

The distribution of training and test data is shown in Figure 7. Machine learning can run well if data from other log parameters are available. There are two stages in implementing machine learning. The first stage is training, which uses data from other wells (Wells 1, 2, and 4) to be studied by the algorithm. After the data has been provided and the predictive model has been formed, the second step is to test the predictive model on well 3 with the same input from other wells. The final result is a DTCO log prediction for well 3.

3.5. Hyperparameter Tuning

In machine learning, the parameters used to control the performance of an algorithm are called hyperparameters. Hyperparameters show the model's structure. The choice and values of a machine learning model's hyperparameters significantly impact how well it performs [17]. The number of hyperparameters depends on the complexity of the algorithm. However, in this study, four essential hyperparameters for the gradient boosting algorithm are shown in Table 4. The table also shows the best hyperparameter search results using the grid search method. Grid search, also known as exhaustive search because of the way it works and one of the most intuitive approaches for performing hyperparameter optimization [18], looks at every combination of existing hyperparameters. Each combination of hyperparameter values specified on the grid will be tested on a model and evaluated for the accuracy of the model [19]. Therefore, to facilitate the evaluation of each hyperparameter tested, grid search is assisted by a cross-validation (CV) technique, which divides all data samples into n-sample groups of the same size if possible. So that later, the model will study a number of n-1 sample groups, and the remaining sample groups will be interpreted as validation data. This technique is also known as K-Fold cross-validation. This study divided the data into 3 sample groups or folds.

Table 4. Grid search results for hyperparameters. The grid used is 3*3*3*3*3 (243) grid.

No	Hyperparameter	Range	Default	Best Hyperparameter
1	n_estimators	50, 80, 100	100	80
2	max_depth	3, 4, 5	6	3
3	min_samples_leaf	1, 2, 3	1	3
4	min_samples_split	2, 5, 10	2	2
5	learning_rate	0.1, 0.15, 0.2	0.1	0.15

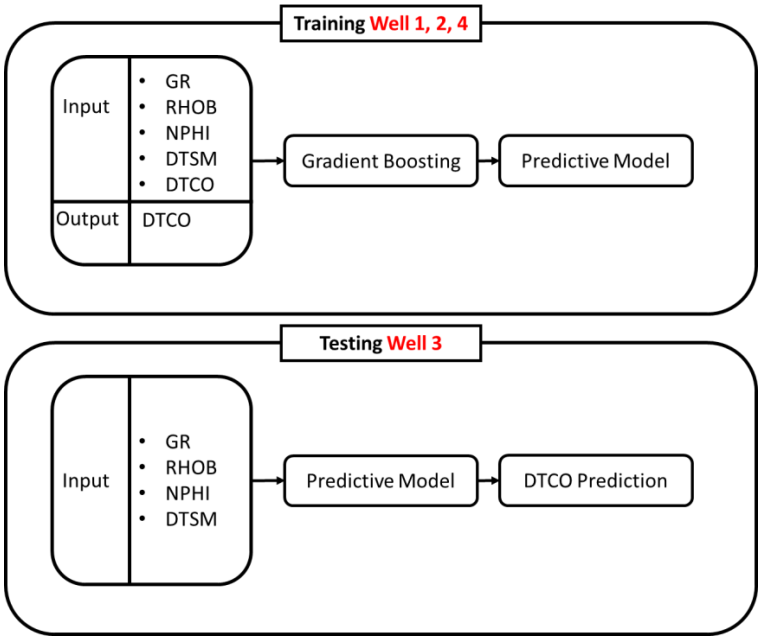

Figure 7. Splitting training and testing data set to predict DTCO logs.

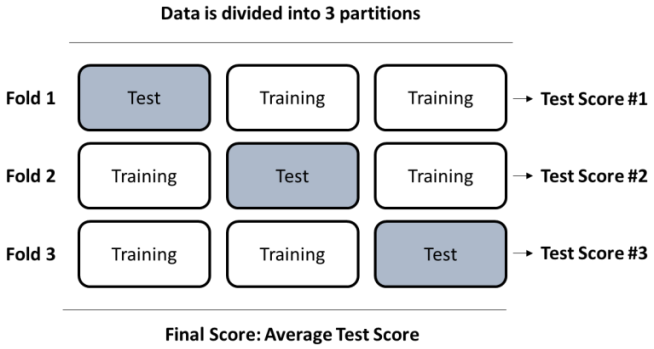

Figure 8. Visualization of the grid search CV concept with 3 folds [20].

4. Result and Discussion

4.1. Machine Learning Result

The predictions generated for well 3 can be seen in Figure 9. The prediction results qualitatively have a fairly good value with several well depths at a depth of around ± 4625 – 4700 m and 4900 – 4950 m having a fairly high error rate. This error may occur because the amount of data and features used as training data is relatively small when compared to previous research references. The second possibility of the emergence of large errors at a certain depth is also caused by the quality of the training data used. However, at some other depths, it shows similarities to the original DTCO log data. This is also reinforced by the results of the low MAPE and RMSE metric scores, which are shown in Table 5.

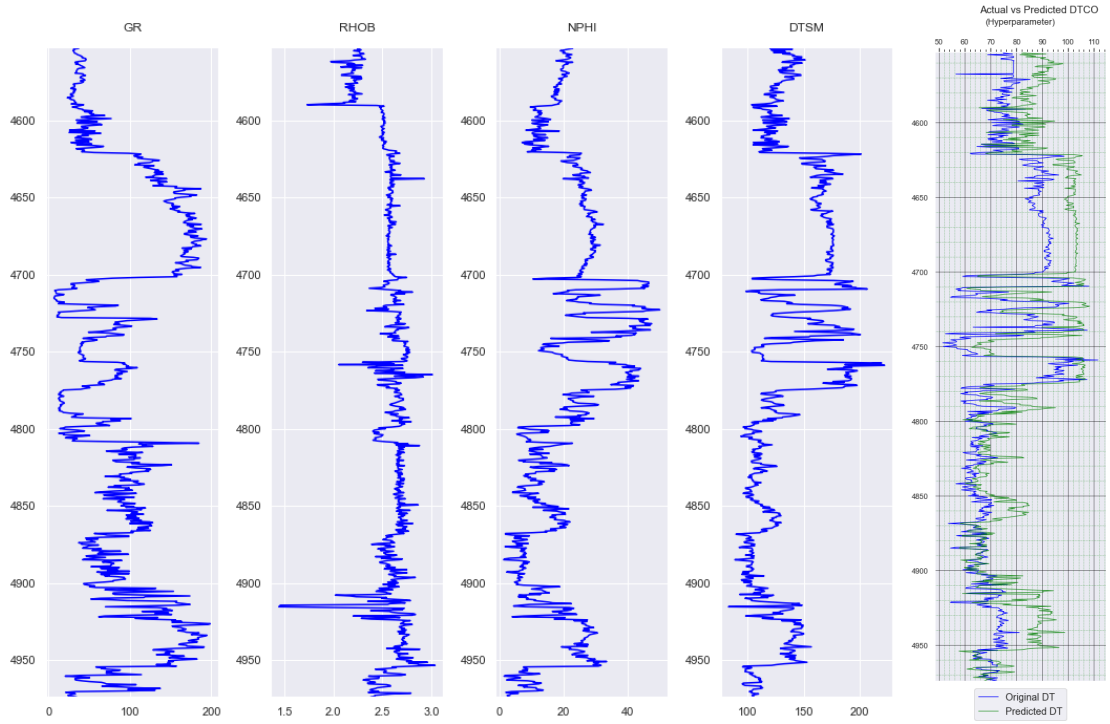

Figure 9. The log's parameters of well 3 and results of the comparison between the DTCO predictions with the original data in well 3.

4.2. The Importance of Hyperparameter Tuning

Table 5. Metric score result for DTCO log prediction on well 3.

No.	Model	MAPE (%)	RMSE	Processing Time (sec)
1	Default	14.57	12.82	0.348
2	Hyperparameter Tuning	12.28	10.74	0.233

Table 5 shows the prediction results obtained have a level of accuracy with an error value of 12.28% with an RMSE of 10.74. This uses machine learning with a gradient-boosting algorithm to produce a reasonably accurate log prediction value. Hyperparameter tuning is important in machine learning since it has been shown to dramatically lower the error value, which is 2.29%. Additionally, using the optimal hyperparameter values can speed up computation. This is because the hyperparameter optimization shows a lower hyperparameter value than the default configuration. Thus, the tree structure formed will also be more superficial.

4.3. The Importance of Data Quantity

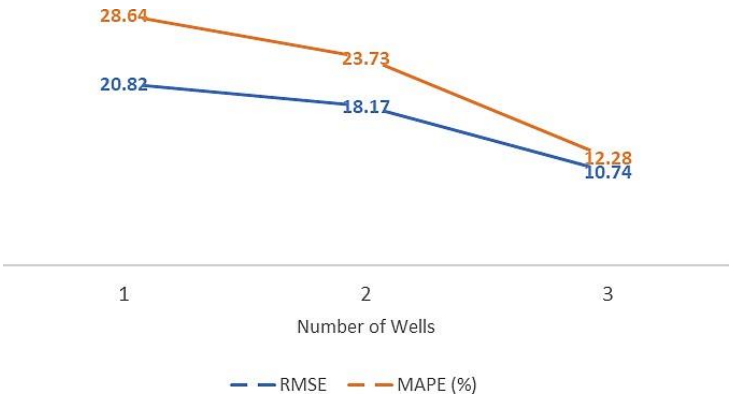

Figure 10. Metric score results from the use of training data for gradient boosting. (RMSE: Root Mean Squared Error; MAPE: Mean Absolute Error Percentage).

Machine learning performance is highly dependent on the quantity of training data used. Figure 10 shows the metric scores for RMSE and MAPE. The horizontal axis shows the amount of training data used, and the numbers on the colored lines are the metric scores. The error value in one amount of training data produces a prediction with a MAPE score of 28.64%. This figure continues to decrease as the quantity of training data increases. Therefore, increasing the amount of training data can improve machine learning's accuracy, especially if huge data is available for DTCO predictions.

5. Conclusion

This paper uses a gradient boosting algorithm to predict compression slowness (DTCO) based on four conventional well log data, 3 of which are training wells. MAPE and RMSE evaluate the error value for the resulting prediction. The existence of this machine learning method can replace empirical methods effectively and accurately without ideal conditions through learning log data around the test well. The conclusions of this study are:

1. The prediction results made by the gradient boosting algorithm have a high similarity with the original DTCO log value. The score metric shows an error rate of 12.28% with an RMSE of 10.74. Therefore, the predictive model has good results and stable performance.
2. It was discovered that hyperparameter tuning is crucial throughtout the machine learning stage. There is a decrease of 2.29% for the resulting error value. In addition, the quantity of data also affects the accuracy of the predictions generated. The more data used for training wells, the better the level of accuracy.

Acknowledgments:
Acknowledgments: The author expresses his gratitude for the financial support by the Directorate of Research and Community Service (DRPM) – Institut Teknologi Sepuluh Nopember (ITS) - Ministry of Education, Culture, Research, and Technology under Indonesian Collaborative Research (Scheme A) - ITS Fund in 2022 with the master contract number 1399/PKS/ITS/2022, on 12 May 2022.

References

[1] H Liu et al 2020 Well Logging Based Lithology Identification Model Establishment Under Data Drift: A Transfer Learning Method. *Sensors*. **20(**13).
[2] V Suleymanov, H Gamal, G Glatz, S Elkatatny, and A Abdulraheem 2021 Real-Time Prediction for Sonic Slowness Logs from Surface Drilling Data Using Machine Learning Techniques.
[3] D Onalo, S Adedigba, F Khan, L A James, and S Butt, 2018 Data driven model for sonic well log prediction. *J. Pet. Sci. Eng.* **170**: 1022–1037.
[4] T Olayiwola and O A Sanuade 2021 A data-driven approach to predict compressional and shear wave velocities in reservoir rocks. *Petroleum*,**7**(2): 199-208.
[5] R Zhong, R Johnson, and Z Chen 2020 Generating pseudo density log from drilling and logging-while-drilling data using extreme gradient boosting (XGBoost). *Int. J. Coal Geol*. **20**.
[6] R Kanfar, O Shaikh, M Yousefzadeh, and T Mukerji 2020 Real-Time Well Log Prediction From Drilling Data Using Deep Learning.
[7] Y Zou, Y Chen, and H Deng 2021 Gradient Boosting Decision Tree for Lithology Identification with Well Logs: A Case Study of Zhaoxian Gold Deposit, Shandong Peninsula, China. *Nat. Resour. Res.* **30.**
[8] T Kumar, N K Seelam, and G S Rao 2022 Lithology prediction from well log data using machine learning techniques: A case study from Talcher coalfield, Eastern India. *J. Appl. Geophys*. **199**.
[9] N Aziz, E Akhir, A P D I Aziz, J Jaafar, M H Hasan, and A Abas 2020 A Study on Gradient Boosting Algorithms for Development of AI Monitoring and Prediction Systems.
[10] C Bentéjac, A Csörgő, and G Martínez-Muñoz 2021 A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev*.**54**(3): 1937-1967.
[11] Z Zhang et al 2018 Exploring the clinical features of narcolepsy type 1 versus narcolepsy type 2 from European Narcolepsy Network database with machine learning. *Sci. Rep*.**8**(1): 1-12.
[12] Y Zhang, X Beudaert, J Argandoña, S Ratchev, and J Munoa 2020 A CPPS based on GBDT for predicting failure events in milling. *Int. J. Adv. Manuf. Technol*.**111**: 1-17.
[13] A Jierula, S Wang, T M OH, and P Wang 2021 Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data. *Applied Sciences*.**11**(5).
[14] A Kumar Dubey, A Kumar, V García-Díaz, A Kumar Sharma, and K Kanhaiya 2020 Study and analysis of SARIMA and LSTM in forecasting time series data. *Sustain. Energy Technol. Assessments*.**47**.
[15] M Riani, A C Atkinson, and A Corbellini 2022 Automatic robust Box–Cox and extended Yeo–Johnson transformations in regression. *Stat. Methods Appt*.
[16] P Schober, C Boer, and L A Schwarte 2018 Correlation Coefficients: Appropriate Use and Interpretation. *Anesth. Analg*. **126**(5).
[17] E Elgeldawi, A Sayed, A R Galal, and A M Zaki 2021 Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics*.**8**(4).
[18] B H Shekar and G Dagnew 2019 Grid Search-Based Hyperparameter Tuning and Classification of Microarray Cancer Data. *in 2019 Second International Conference on Advanced Computational and Communication Paradigms* (ICACCP). pp 1–8.
[19] P Probst, A L Boulesteix, and B Bischl 2019 Tunability: Importance of hyperparameters of machine learning algorithms. *J. 290 Mach. Learn. Res*. **20**: 1-32.
[20] V H Phung and E J Rhee 2019 A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. *Applied Sciences*.**9**(21).