RESEARCH ARTICLE • OPEN ACCESS

Shear Wave Travel Time Prediction using Well Log Filtering and Machine Learning

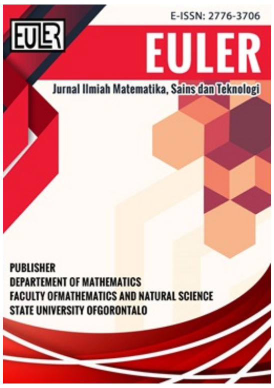
Indra Rivaldi Siregar, Adhiyatma Nugraha, Anwar Fitrianto, Erfiani, and L.M. Risman Dwi Jumansyah

Volume 12, Issue 2, Pages 206–212, Dec 2024

Received 29 September 2024, Revised 9 December 2024, Accepted 17 December 2024, Published 19 December 2024

To Cite this Article : I. R. Siregar, et al.,"Shear Wave Travel Time Prediction using Well Log Filtering and Machine Learning", *Euler J. Ilm. Mat. Sains dan Teknol.*, vol. 12, no. 2, pp. 206–212, 2024, *https://doi.org/10.37905/euler.v12i2.29021*

JOURNAL INFO • EULER : JURNAL ILMIAH MATEMATIKA, SAINS DAN TEKNOLOGI


E-ISSN: 2776-3706
EULER
Jurnal Ilmiah Matematika, Sains dan Teknologi
PUBLISHER
DEPARTEMENT OF MATHEMATICS
FACULTY OFMATHEMATICS AND NATURAL SCIENCE
STATE UNIVERSITY OFGORONTALO

Homepage	:	http://ejurnal.ung.ac.id/index.php/euler/index
Journal Abbreviation	:	Euler J. Ilm. Mat. Sains dan Teknol.
Frequency	:	Biannual (June and December)
Publication Language	:	English (preferable), Indonesia
DOI	:	https://doi.org/10.37905/euler
Online ISSN	:	2776-3706
Editor-in-Chief	:	Resmawan
Publisher	:	Department of Mathematics, Universitas Negeri Gorontalo
Country	:	Indonesia
OAI Address	:	http://ejurnal.ung.ac.id/index.php/euler/oai
Google Scholar ID	:	QF_r-_gAAAAJ
Email	:	euler@ung.ac.id

Euler : Jurnal Ilmiah Matematika, Sains dan Teknologi, Volume 12, Issue 2, Pages 206–212, Dec 2024 *https://doi.org/10.37905/euler.v12i2.29021*

Research Article

Check for updates

Shear Wave Travel Time Prediction using Well Log Filtering and Machine Learning

Indra Rivaldi Siregar[1,*], **Adhiyatma Nugraha**[1], **Anwar Fitrianto**[1], **Erfiani**[1], **and L.M. Risman Dwi Jumansyah**[1]

[1] *Study Program of Statistics and Data Science, IPB University, Bogor Indonesia*

ARTICLE HISTORY
Received 29 September 2024
Revised 9 December 2024
Accepted 17 December 2024
Published 19 December 2024

KEYWORDS
K-nearest neighbors
Machine learning
Random forest
Shear wave travel time
Smoothing
XGBoost

ABSTRACT. *Shear wave travel time (also known as Delta-T Shear and commonly abbreviated as DTS) is an important parameter in petroleum for exploration, production, and characterization of borehole stability. Direct measurement of DTS is often limited by high costs and a constraint of geography, making machine learning (ML) predictive approaches necessary. This study aims to explore the effectiveness of ML models in predicting DTS, emphasizing the importance of data preprocessing techniques to improve prediction accuracy. Preprocessing techniques include Yeo-Johnson transformation to handle non-normality, outlier elimination using z-score, and data smoothing using the Savitzky-Golay filter and median filter. Incorporating smoothing techniques can fill important gaps in some existing studies and may improve the performance of machine learning models in predicting DTS, particularly in situations with limited or noisy data. Four ML models were tested in this study, namely Linear Regression (LR), K-Nearest Neighbors (KNN), Extreme Gradient Boosting (XGBoost), and Random Forest (RF), with performance evaluation based on metrics RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and R2 (coefficient of determination). The results showed that the RF model produced the best performance with RMSE of 9.41, MAE of 6.35, and R2 of 0.90 in scenarios with Yeo-Johnson transformation, outlier elimination, and smoothing techniques using a median filter with a window size of 5.*

1. Introduction

The shear wave travel time (called Delta-T Shear and typically denoted by DTS) data from sedimentary rock layers is necessary to apply most mathematical models in petroleum engineering geomechanics [1–3]. Geomechanical models are essential for understanding rock properties, predicting reservoir behavior, and optimizing hydrocarbon extraction. The DTS in geomechanical models is directly linked to the mechanical properties of rocks, such as hardness and strength. DTS significantly improves the capacity to assess these properties, facilitating informed decisions in drilling, injection, and extraction. Additionally, DTS plays a pivotal role in fracture analysis and diagnostics, providing crucial insights for understanding fracture behavior, optimizing operations, and managing reservoirs effectively. The data required (including DTS) for geomechanical modeling is obtained from wellbore core samples. However, due to the high cost and time, most wells lack DTS data acquisition using advanced dipole sonic logs [1, 4]. As a result, the availability of DTS data for geomechanical modeling is limited, leading to the development of many estimation and extrapolation techniques to address this data gap. Therefore, reliable methods for estimating DTS are imperative, considering their critical impacts on decisions during drilling and production processes [5].

Mathematically, DTS data is directly transformed into shear wave velocity (Vs) through an inverse calculation. Conventional methods in predicting shear wave velocities (Vs), such as those proposed by Pickett [6], Castagna et al. [7], and Brocher [8], offer a fast and convenient way to estimate it. However, their applicability and accuracy are often limited due to variations in subsurface rock types and geographical differences, resulting in poor generalization [3]. These methods are empirical correlations for predicting Vs that rely on compressional velocity (Vp) and have accuracy limitations, mainly due to their dependence on lithology type and their field- or basin-specific nature. Their lack of generalizability and poor fit with real data limit their application across different fields.

Machine learning (ML) approaches can model such data without mathematical models like conventional methods. ML offers a promising alternative to traditional empirical methods in predicting shear wave velocity [9, 10]. As shown by the results of [1], the performance of deep and hybrid machine learning algorithms (R^2 between 0.97 and 0.98, and RMSE between 0.05 and 0.06) is superior to conventional methods such as Pickett, Carroll, Castagna, Eskandari, and Brocher (R^2 between 0.85 and 0.87, and RMSE between 0.13 and 0.21). By utilizing available well log data, such as density, porosity, and resistivity, ML techniques can make predictions in scenarios where direct measurements are impractical or too costly [5]. These methods excel at identifying patterns and complex relationships in the data, leading to more accurate and reliable predictions.

[*] Corresponding Author.

Email : *siregarindra@apps.ipb.ac.id* (I. R. Siregar)
Homepage : http://ejurnal.ung.ac.id/index.php/euler/index / E-ISSN : 2776-3706

However, the high performance of ML models depends significantly on proper data preprocessing before modeling [11, 12].This is particularly relevant given that well log data often contains signal noise or distortions [13–15]. We observe potential in handling data preprocessing techniques, such as smoothing, which could improve ML model performance in addition to outlier removal techniques.

Previous studies [1, 9, 16] focused on developing more advanced methods by combining multiple algorithms in machine learning or deep learning. Another approach [4] took a different path through feature engineering, utilizing nearby data points in depth as eigenvalues for machine learning to improve the accuracy of single-well predictions. Thus, we seek to enhance the performance of DTS predictions using an alternative approach involving smoothing techniques. Integrating smoothing techniques potentially enhance the performance of ML models in predicting shear wave travel time, especially in cases with limited or noisy data.

The smoothing techniques explored in this study include well-known methods such as the Savitzky–Golay filter [17, 18] and median filter [19]. Meanwhile, the ML methods we will apply include K-nearest neighbors (KNN), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). We selected these algorithms because previous works have demonstrated their strong predictive capabilities across various cases and datasets [20–22]. In this work, we also implement Linear Regression (LR) as a baseline model to determine whether the ML methods offer significant improvements. We expect these smoothing techniques will reduce distortions in well log data, thereby improving the model's performance in predicting DTS.

2. Methods

The data used in this study is sourced from Equinor [23], specifically from the Volve Field located in the Norwegian North Sea, approximately 200 kilometers west of Norway. We use well log data from four wells, where F-1 B, F-11 A, and F-11 T2 are used for model training, while Well 15_9-F-1 A serves as the blind well (testing data). The blind well is randomly chosen from the available wells. Each well dataset consists of six log types: gamma ray (GR), bulk density (RHOB), neutron porosity (NPHI), true formation resistivity (RT), compressional sonic (DTC), and shear sonic (DTS). To predict shear wave travel times (represented by DTS), all well log (except DTS) and depth are input features in this study. The steps of this study are illustrated in Figure 1.

We combine data from three wells (F-1B, F-11A, and F-11T2) into a single dataset to develop machine learning models for four algorithms. Before modeling, we explore the data, focusing on distribution analysis, outlier detection, and examining correlations between predictors and the target DTS. Based on the exploration results, we carry out five scenarios, each incorporating different data pre-processing procedures. Scenarios 3 - 5 include smoothing techniques for well log data. The details of these scenarios are displayed in Table 1.

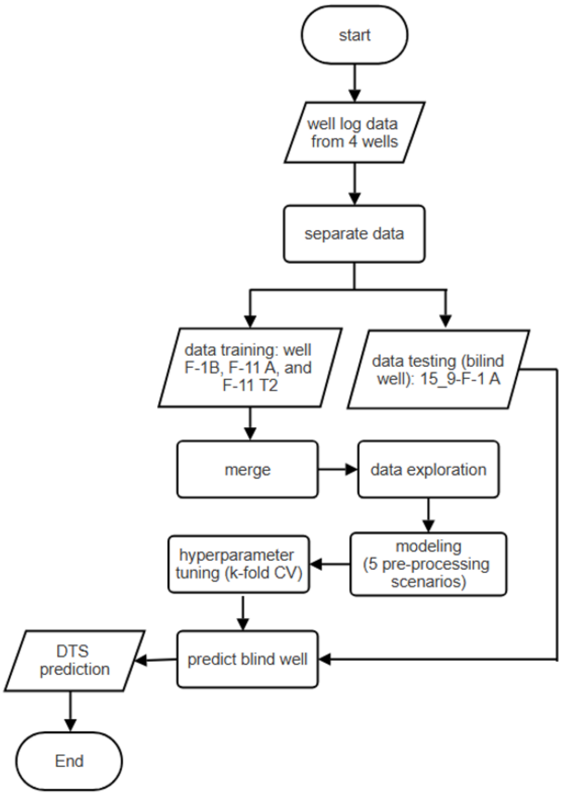

start
well log data from 4 wells
separate data
data training: well F-1B, F-11 A, and F-11 T2
data testing (bilind well): 15_9-F-1 A
merge
data exploration
modeling (5 pre-processing scenarios)
hyperparameter tuning (k-fold CV)
predict blind well
DTS prediction
End

Figure 1. Flowchart

Table 1. Pre-processing scenarios

Scenario	Pre-processing data			
	Feature transformation (Yeo-Johnson)	Handling outlier (z-score)	smoothing	normalization
1	-	-	-	✓
2	✓	✓	-	✓
3	✓	✓	Savitzky–Golay filter	✓
4	✓	✓	median filter w=3	✓
5	✓	✓	median filter w=5	✓

- *Handling Outlier:* An outlier refers to data points that deviate significantly from the general pattern of the dataset. Their presence can sometimes disrupt the performance of machine learning models, necessitating actions such as outlier removal. Thoughtful outlier removal can enhance model accuracy, reduce noise, and improve data quality and interpretability [24]. In this study, we remove the data points indicated as outliers using the z-score method [25, 26]. Technically, we calculate the z-score for all data points in each feature, and then any data points with an absolute z-score > 3 are removed.
- *Yeo-Johnson transformation*: Careless removal of outliers can lead to the loss of natural data patterns or the discarding of a substantial portion of data due to overly simplistic detection methods. Therefore, we first conduct a Yeo-Johnson transformation to avoid eliminating a significant portion of the data because of outlier detection. This transformation aims to reduce skewness and make data more Gaussian-like, supporting both positive and negative data [27], so fewer outliers need to be removed compared to a non-transformed

dataset.

- *Smoothing*: Well log data, the raw signals collected by instruments, typically includes noise and distortions. Therefore, smoothing techniques are applied to address these issues and assist machine learning in capturing the general patterns of the data while avoiding overfitting. Two popular smoothing methods, the Savitzky–Golay Filter and the Median Filter, are used in this study. Savitzky–Golay filter works by fitting successive polynomials to sliding windows of the data, preserving prominent features like peaks and reducing noise without significantly distorting the original signal [17, 18]. Meanwhile, the median filter works with a schema that replaces each data point with the median value within a sliding window [19].
- *Normalization*: The goal of min-max normalization is to adjust the data to a range of [0, 1], ensuring that all features have an equal influence on the model and enhancing its performance [28].

In machine learning regression, RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and R^2 (R-squared) are commonly used evaluation metrics to assess model performance [29]. RMSE measures the average squared differences between predicted and actual values, giving more weight to larger errors, and is sensitive to outliers. MAE, on the other hand, calculates the average of absolute differences, treating all errors equally, being less affected by outliers, and is easy to interpret. R^2 indicates how well the model explains the variance in the target variable, with a higher value signifying better model fit. So, using all these metrics can evaluate the models comprehensively.

In this study, we utilize k-fold cross-validation for hyperparameter tuning to ensure a more robust evaluation of the model [30]. By using this technique, we can mitigate the risk of overfitting and ensure that our hyperparameter tuning is based on how well the model generalizes to unseen data, rather than how it performs on a single training/validation split. In this study, we set k to 5, balancing computational efficiency with the need for accurate performance estimates.

3. Results and Discussion

3.1. Data Exploration

Shear wave travel time (DTS) represents the time required for a shear wave to travel a specific distance through a rock formation, commonly expressed in microseconds per foot (us/ft). DTS is measured using advanced dipole sonic logs (such as dipole sonic and multipole sonic), a well-logging technique employed to analyze the mechanical properties of subsurface rocks [31].

Figure 2 shows well log data from the three wells used to build the model. First, we drop the data for the depth that does not contain DTS. And then, we combine the data from these wells into a single dataset, resulting in 32,587 data points. Combining data from multiple wells is essential to capture a broader and more representative subsurface variation, such as lithology and mechanical properties. Relying on data from a single well may introduce bias and limit model accuracy. By merging data from three wells, we create a larger dataset, improving the model's ability to generalize and make more accurate predictions.

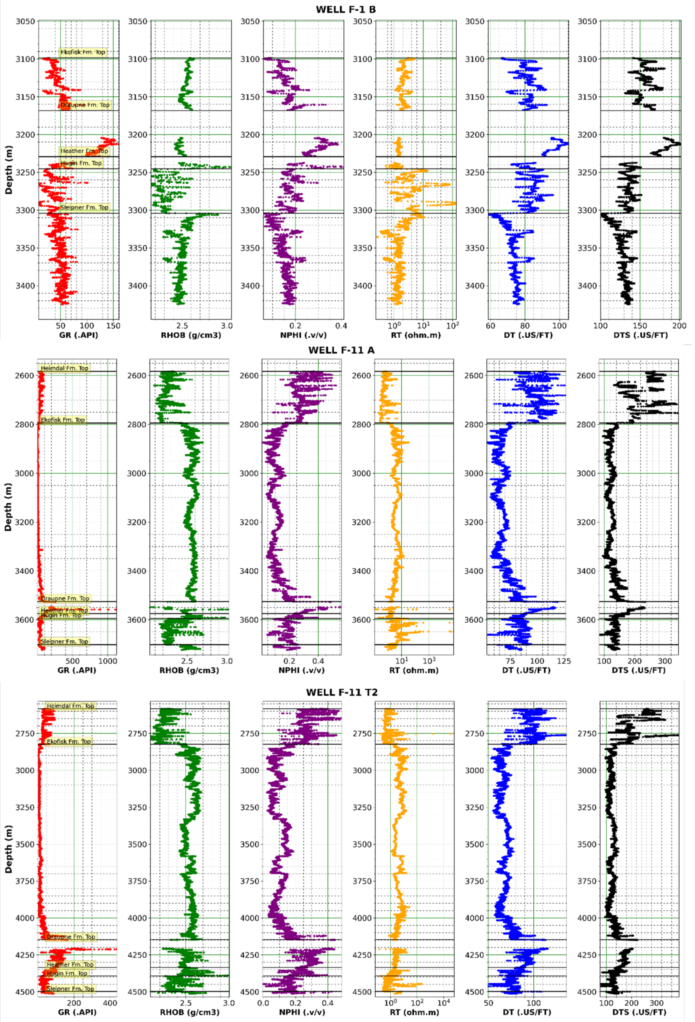


Figure 2. Well log data: Well F-1 B (top), Well F-11 A (middle), Well F-11 T2 (bottom)

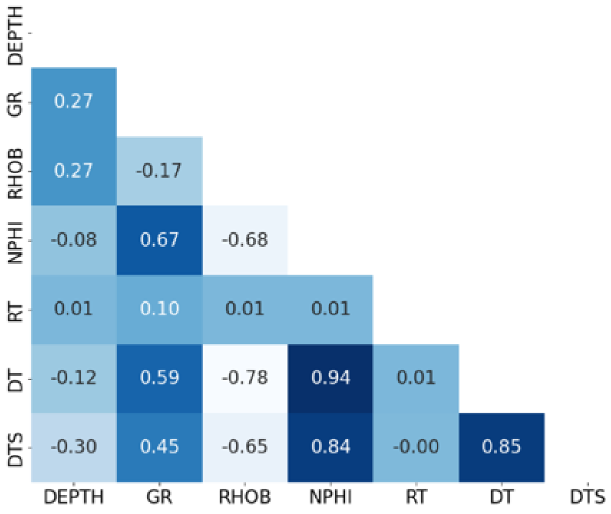


Figure 3. Pearson's correlation heatmap of all features (Combined wells F-1 B, F-11 A, and F-11 T2)

In the initial stage of data exploration, we utilize Pearson's correlation, presented in Figure 3, to assess the linear relationship between predictors and the target variable. The DT, NPHI, and RHOB logs exhibit an absolute Pearson's correlation ≥ 0.65 with DTS. This observation is further supported by the pair plot in

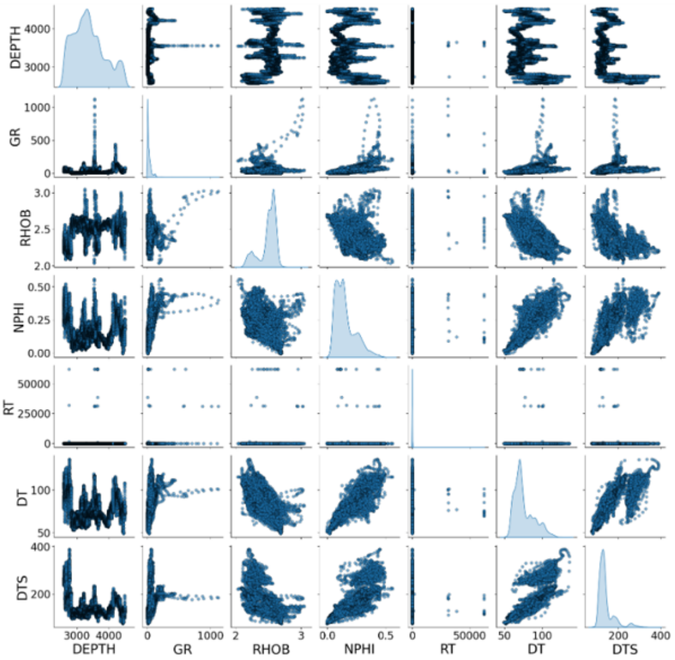


Figure 4. Pairplot of all features (Combined wells F-1 B, F-11 A, and F-11 T2)

Figure 4, which reveals strong linear correlations between these predictors and the DTS response, highlighting their potential to enhance model performance. These three predictors are particularly beneficial for improving model accuracy, especially in linear regression, which assumes a linear relationship between predictors and the target variable. Meanwhile, depth and GR exhibit relatively weak linear correlations with DTS. The RT log shows no linear correlation, with a value of 0. However, previous studies consistently include RT when predicting DTS [32–35], so we will retain it in our model. This case will still be accommodated, as all machine learning models used in this study, such as KNN, XGBoost, and RF, do not assume a linear relationship between features and the target. Therefore, these three models are likely to remain effective in modeling features and targets that may have a non-linear relationship. However, linear regression models might struggle to capture the non-linear relationship between the predictors and the target, and then will be represented by the weak predictive performance.

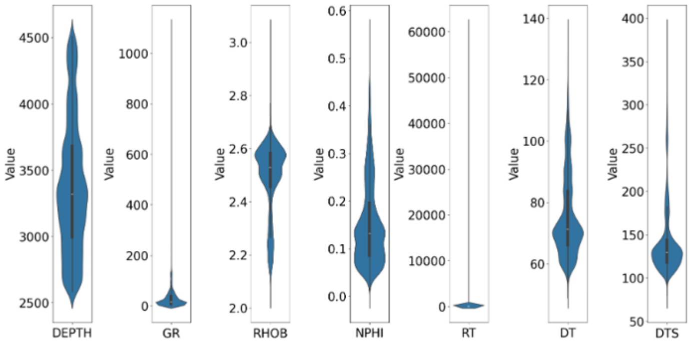


Figure 5. Violin plot of all features (Combined wells F-1 B, F-11 A, and F-11 T2)

From the violin plot in Figure 5, it is evident that all features (except depth) are relatively right skewed and tend to contain outliers. We first applied the Yeo-Johnson transformation to reduce skewness, making their distributions approximately normal. For RT and GR log data, which have many values close to or equal to 0, this transformation is more suitable than a standard logarithmic transformation. We prioritize applying the transformation before removing detected outliers to minimize data loss and preserve the natural information of the dataset. This strategy preserves the dataset's comprehensiveness and upholds its original integrity. We apply this transformation to scenarios 2 through 5.

Subsequently, we apply outlier removal for all features detected using the z-score method in scenarios 2 through 5. Failure to handle these outliers may lead to suboptimal model performance, particularly for LR and KNN models, which are sensitive to outliers [36, 37]. After outlier removal, the dataset contains 31,164 data points, a reduction of 4.37%. If we compare it with an approach that directly removes outliers without transformation, the reduction is 5.15%.

3.2. *Modeling*

We have experimented with four models: LR, KNN, XGBoost, and RF. No hyperparameter tuning is necessary for LR as it is a parametric method with a direct analytical solution. However, hyperparameter tuning is crucial in optimizing model performance for machine learning models (e.g., KNN, XGBoost, and RF) [38]. Hyperparameter tuning involves selecting the optimal set of hyperparameters that govern the algorithm's learning process. We have explored many combinations of hyperparameters for each model using grid search to significantly influence the accuracy, generalization, and efficiency [30], as shown in Table 2.

Table 2. Hyperparameter combinations

Algorithm	Hyperparameters
KNN	k: [1, 2, 3, …, 30] p: [Manhattan, Euclidean]
XGBoost	min_child_weight: [1, 3, 5, 7, 15, 30] max_depth: [3, 5, 7, 12, 15, 30] gamma: [0, 0.1, 0.2, …, 1] n_estimators: [10, 100, 200, 300, 500] learning_rate: [0.01, 0.05, 0.1, 0.3, 0.5, 0.9]
RF	n_estimators: [100, 200, …, 2000] max_depth: [10, 20, 30, …, 110] min_samples_split: [2, 4, 6, 8, 10] min_samples_leaf: [1, 3, 5, 7, 10] max_features: [auto, sqrt, log2]

For the KNN model, we adjust two main hyperparameters: the number of neighbors (k) and the distance metric (p). Choosing the appropriate distance metric depends on the dataset, and previous studies show that the right combination of k and distance metric can greatly improve the performance of KNN [20]. Key hyperparameters in XGBoost require careful adjustment to ensure optimal model performance, such as gamma, min_child_weight, max_depth, n_estimators, and learning_rate [39]. In the RF model, we tune the n_estimators, max_depth, min_samples_split, min_samples_leaf, and max_features [21].

Table 3 summarizes the model evaluation results across five different preprocessing scenarios for four models: LR, KNN, XG-

Table 3. Model evaluation (blind well 15_9-F-1 A)

Scenario	LR			KNN			XGBoost			RF		
	RMSE	MAE	R²	RMSE	MAE	R²	RMSE	MAE	R²	RMSE	MAE	R²
1	**13.13**	9.10	**0.83**	17.853	9.26	0.74	18.38	9.44	0.73	14.93	**8.19**	0.81
2	10.94	8.81	0.81	15.60	10.13	0.77	11.98	8.03	0.85	**9.76**	**6.61**	**0.89**
3	10.94	8.81	0.81	17.93	12.85	0.72	12.52	8.04	0.83	**9.75**	**6.62**	**0.89**
4	10.90	8.80	0.81	15.80	11.49	0.74	12.47	8.36	0.84	**9.59**	**6.44**	**0.89**
5	10.80	8.67	0.81	12.11	7.94	0.85	12.18	8.52	0.85	**9.41**	**6.35**	**0.90**

**Note:The values in bold represent the optimal RMSE, MAE, and R^2 for a given model in each scenario

Boost, and RF. The evaluation metrics used include RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and R^2 (coefficient of determination), with the optimal values for each scenario highlighted in bold.

1. *Scenario 1*: We only apply a normalization for this initial scenario. Surprisingly, we observe the best performance of the LR model, with an RMSE of 13.13 and R^2 of 0.83, indicating that this model gets the best benefits by only normalization preprocessing. RF also performs well in terms of MAE, with 8.19, but it has a much higher RMSE.
2. *Scenario 2*: In this scenario, we add the Yeo-Johnson transformation to handle non-normality and apply z-score to remove outliers while retaining normalization. RF outperforms the other models, with an RMSE of 9.76, MAE of 6.61, and R^2 of 0.89, showing improved model performance after handling outliers. XGBoost also benefits significantly, with RMSE reducing to 11.98, MAE of 8.03, and R^2 increasing to 0.85. Overall, for all models, we see that the error rate in this scenario decreases and R^2 increases compared to scenario 1. Therefore, the handling outlier step mathematically improves our models in this case.
3. *Scenario 3*: This scenario mostly has the same steps as scenario 2, but we also add the Savitzky-Golay filter in the preprocessing step before modeling. As shown in the table, applying this filter to the RF model slightly improves its performance (RMSE 9.75, R^2 0.89), while other models, such as XGBoost, also benefit from this additional step. However, KNN still struggles with high RMSE and MAE, showing that it may be more noise-sensitive than RF or XGBoost models. Meanwhile, there are no improvements for LR compared to scenario 2.
4. *Scenario 4*: As in scenario 3, we also conduct a smoothing technique for the well log data using a median filter with a window size of 3 for this scenario. Compared to scenarios 2 and 3, the median filter yields similar improvements for RF, with an RMSE of 9.59 and MAE of 6.44, but fewer benefits for XGBoost (the MAE and R2 are better in scenario 2). The performance of the KNN model remains relatively unchanged, indicating that smoothing may not have been enough for the model to generalize well. Based on RMSE, LR outperforms XGBoost, even though this scenario does not affect the LR model compared to scenarios 2 and 3.
5. *Scenario 5*: In the final scenario, we also implement the median filter with a window size 5. As a result, the RF model achieves its best performance in this scenario, with an RMSE of 9.41, MAE of 6.35, and R^2 of 0.90, the highest of all scenarios. The performance of XGBoost also improves (RMSE 12.18, R^2 0.85), although it does not surpass RF.

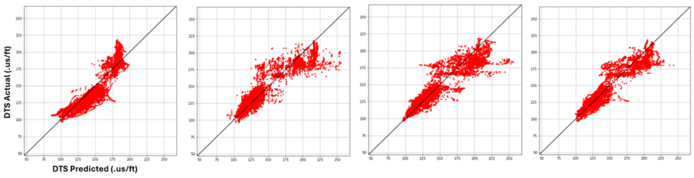


Figure 6. Cross plot between DTS prediction and DTS actual in blind well using scenario 5 (left to right: LR, KNN, XGBoost, RF)

The cross plots between predicted and actual DTS provided clear visual insights into model performance (Figure 6). The closer the points were to the 1:1 line, the more accurate the predictions. The RF model (far right) demonstrates the best alignment with the 1:1 line, particularly in scenario 5, which employs median filter (w=5). It is also supported by the model performance metrics from the table, where RF achieves the lowest RMSE and MAE, along with the highest R^2. The high R^2 indicates that the model captures most of the variability in the DTS values reflected in the cross plot by a tighter cluster of points around the diagonal. For other models, the deviation from the 1:1 line is more noticeable. Although LR has a relatively high R^2 (0.81 in scenario 5), its predictions are less accurate due to a noticeable spread of points, especially at higher DTS values. KNN's cross plot reveals significant dispersion, reflecting its poor performance metrics (RMSE of 12.11 and MAE of 7.94 in Scenario 5). XGBoost performs better than KNN but still exhibits scatter, particularly in the mid-range of DTS values.

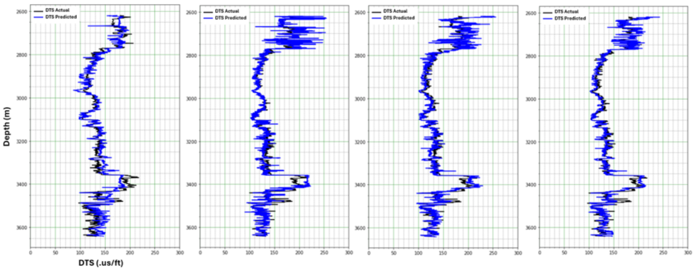


Figure 7. DTS prediction in blind well using scenario 5 (left to right: LR, KNN, XGBoost, RF)

A similar trend is also observed in the predicted DTS well log plots (Figure 7). RF consistently tracks the original DTS values closely, especially in the depth ranges where other models begin to deviate. It is particularly evident around the 2600 - 2800 m depth, where LR and KNN exhibit noticeable divergence from the actual DTS values, leading to larger prediction errors. LR successfully captures the general trend of the DTS log but has difficulty with more detailed variations, especially at greater depths (e.g., below 3200 m), where predicting DTS values becomes more challenging. In contrast, KNN generates much noisier predictions, with significant deviations from the actual DTS values throughout all depth intervals, leading to its overall weaker performance (RMSE of 12.11 in scenario 5). XGBoost delivers smoother predictions than KNN, though it still falls short of RF, particularly in areas with rapid DTS variations. Its RMSE of 12.18 in scenario 5 highlights these minor inconsistencies. In general, the visual agreement in the well log predictions confirms the quantitative findings from the performance table, with RF standing out as the top-performing model across the various preprocessing scenarios. It shows the effectiveness of more advanced preprocessing, such as feature transformation, outlier handling, and smoothing, particularly for the RF model. We also conclude the best smoothing technique in this case is a median filter with a window size of 5.

4. Conclusion

Based on the results above, we conclude that the Random Forest (RF) model demonstrates the best performance in almost all scenarios, except for scenario 1. Scenario 5, which included Yeo-Johnson transformation, outlier removal, smoothing with median filter (w=5), and normalization, emerged as the best scenario with the lowest RMSE and MAE values of 9.41 and 6.35, respectively, as well as the highest R^2 of 0.90. We observed that outlier removal and smoothing had a significant impact on the overall model performance. The KNN model exhibited greater sensitivity to noise, resulting in higher RMSE and MAE values compared to the other models across all scenarios. These findings highlight the importance of data preprocessing, especially the smoothing technique, in improving model performance, particularly for machine learning models like Random Forest. The success of Scenario 5 reinforces the idea that careful smoothing and transformation can significantly enhance predictive accuracy, which is crucial for real-world applications where noise and outliers are common.

Author Contributions. **Indra Rivaldi Siregar:** Conceptualization, methodology, formal analysis, visualization. **Adhiyatma Nugraha:** Resources, data curation, software. **Anwar Fitrianto:** Writing—review and editing, validation. **Erfiani:** Writing—review and editing, supervision. **L.M. Risman Dwi Jumansyah:** Project administration, writing—review and editing. All authors discussed the results and contributed to the final manuscript.

Acknowledgement. The authors would like to thank Equinor for providing the data. The authors are also grateful the editors and reviewers who have supported us in improving this manuscript.

Funding. This research received no external funding.

Conflict of interest. The authors declare that there are no conflicts of interest related to this article.

Data availability. Data source: Equinor. https://www.equinor.com/energy/volve-data-sharing

References

[1] M. Rajabi *et al.*, "Predicting shear wave velocity from conventional well logs with deep and hybrid machine learning algorithms," *J Pet Explor Prod Technol*, vol. 13, no. 1, pp. 19–42, Jan. 2023, doi: 10.1007/s13202-022-01531-z.

[2] N. Mohamadian, H. Ghorbani, D. A. Wood, M. Mehrad, S. Davoodi, S. Rashidi, A. Soleimanian, and A. K. Shahvand., "A geomechanical approach to casing collapse prediction in oil and gas wells aided by machine learning," *J Pet Sci Eng*, vol. 196, p. 107811, Jan. 2021, doi: 10.1016/j.petrol.2020.107811.

[3] S. Parvizi, R. Kharrat, M. R. Asef, B. Jahangiry, and A. Hashemi, "Prediction of the Shear Wave Velocity from Compressional Wave Velocity for Gachsaran Formation," *Acta Geophysica*, vol. 63, no. 5, pp. 1231–1243, Oct. 2015, doi: 10.1515/acgeo-2015-0048.

[4] S. Liu, Y. Zhao, and Z. Wang, "Artificial Intelligence Method for Shear Wave Travel Time Prediction considering Reservoir Geological Continuity," *Math Probl Eng*, vol. 2021, pp. 1–18, Mar. 2021, doi: 10.1155/2021/5520428.

[5] M. Dehghani, S. Jahani, and A. Ranjbar, "Comparing the performance of machine learning methods in estimating the shear wave transit time in one of the reservoirs in southwest of Iran," *Sci Rep*, vol. 14, no. 1, p. 4744, Feb. 2024, doi: 10.1038/s41598-024-55535-2.

[6] G. R. Pickett, "Acoustic Character Logs and Their Applications in Formation Evaluation," *Journal of Petroleum Technology*, vol. 15, no. 06, pp. 659–667, Jun. 1963, doi: 10.2118/452-PA.

[7] J. P. Castagna, M. L. Batzle, and R. L. Eastwood, "Relationships between compressional-wave and shear-wave velocities in clastic silicate rocks," *GEOPHYSICS*, vol. 50, no. 4, pp. 571–581, Apr. 1985, doi: 10.1190/1.1441933.

[8] T. M. Brocher, "Empirical Relations between Elastic Wavespeeds and Density in the Earth's Crust," *Bulletin of the Seismological Society of America*, vol. 95, no. 6, pp. 2081–2092, Dec. 2005, doi: 10.1785/0120050077.

[9] X. Fu, Y. Wei, Y. Su, and H. Hu, "Shear Wave Velocity Prediction Based on the Long Short-Term Memory Network with Attention Mechanism," *Applied Sciences*, vol. 14, no. 6, p. 2489, Mar. 2024, doi: 10.3390/app14062489.

[10] J. Liu, Z. Gui, G. Gao, Y. Li, Q. Wei, and Y. Liu, "Predicting Shear Wave Velocity Using a Convolutional Neural Network and Dual-Constraint Calculation for Anisotropic Parameters Incorporating Compressional and Shear Wave Velocities," *Processes*, vol. 11, no. 8, p. 2356, Aug. 2023, doi: 10.3390/pr11082356.

[11] E. Alshdaifat, D. Alshdaifat, A. Alsarhan, F. Hussein, and S. M. F. S. El-Salhi, "The Effect of Preprocessing Techniques, Applied to Numeric Features, on Classification Algorithms' Performance," *Data (Basel)*, vol. 6, no. 2, p. 11, Jan. 2021, doi: 10.3390/data6020011.

[12] T. Johnson, A. J. Liu, S. Raza, and A. McGuire, "A Comparison of Modeling Preprocessing Techniques," Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.12042.

[13] F. Branisa, "FILTERING OF WELL-LOG CURVES," *GEOPHYSICS*, vol. 39, no. 4, pp. 545–549, Aug. 1974, doi: 10.1190/1.1440447.

[14] M. J. Duchesne and P. Gaillot, "Did you smooth your well logs the right way for seismic interpretation?," *Journal of Geophysics and Engineering*, vol. 8, no. 4, pp. 514–523, Dec. 2011, doi: 10.1088/1742-2132/8/4/004.

[15] S. Soltani, M. Kordestani, and P. Karim Aghaee, "New estimation methodologies for well logging problems via a combination of fuzzy Kalman filter and different smoothers," *J Pet Sci Eng*, vol. 145, pp. 704–710, Sep. 2016, doi: 10.1016/j.petrol.2016.06.032.

[16] J. Wang, J. Cao, and S. Yuan, "Shear wave velocity prediction based on adaptive particle swarm optimization optimized recurrent neural network," *J Pet Sci Eng*, vol. 194, p. 107466, Nov. 2020, doi: 10.1016/j.petrol.2020.107466.

[17] H. Azami, K. Mohammadi, and B. Bozorgtabar, "An Improved Signal Segmentation Using Moving Average and Savitzky-Golay Filter," *Journal of Signal and Information Processing*, vol. 03, no. 01, pp. 39–44, 2012, doi: 10.4236/jsip.2012.31006.

[18] S. R. Moosavi, J. Qajar, and M. Riazi, "A comparison of methods for denoising of well test pressure data," *J Pet Explor Prod Technol*, vol. 8, no. 4, pp. 1519–1534, Dec. 2018, doi: 10.1007/s13202-017-0427-y.

[19] D. C. Stone, "Application of median filtering to noisy data," *Can J Chem*, vol. 73, pp. 1573–1581, 1997.

[20] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, Jun. 2016, doi: 10.1016/j.neucom.2015.08.112.

[21] E. Scornet, "Tuning parameters in random forests," *ESAIM Proc Surv*, vol. 60, pp. 144–162, 2017, doi: 10.1051/proc/201760144.

[22] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[23] Equinor, "Disclosing all Volve data." Accessed: Sep. 19, 2024. [Online]. Avail-

able: https://www.equinor.com/energy/volve-data-sharing.

[24] C. C. Aggarwal, "An Introduction to Outlier Analysis," in *Outlier Analysis*, Cham: Springer International Publishing, 2017, pp. 1–34. doi: 10.1007/978-3-319-47578-3_1.

[25] P. Venkataanusha, Ch. Anuradha, Dr. P. S. R. Chandra Murty, and Dr. S. K. Chebrolu, "Detecting Outliers in High Dimensional Data Sets Using Z-Score Methodology," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 48–53, Nov. 2019, doi: 10.35940/ijitee.A3910.119119.

[26] V. Aggarwal, V. Gupta, P. Singh, K. Sharma, and N. Sharma, "Detection of Spatial Outlier by Using Improved Z-Score Test," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, Apr. 2019, pp. 788–790. doi: 10.1109/ICOEI.2019.8862582.

[27] L. Sun *et al.*, "Ensemble stacking rockburst prediction model based on Yeo–Johnson, K-means SMOTE, and optimal rockburst feature dimension determination," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-19669-5.

[28] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl Soft Comput*, vol. 97, Dec. 2020, doi: 10.1016/j.asoc.2019.105524.

[29] A. Jierula, S. Wang, T.-M. OH, and P. Wang, "Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data," *Applied Sciences*, vol. 11, no. 5, p. 2314, Mar. 2021, doi: 10.3390/app11052314.

[30] S. M. Malakouti, M. B. Menhaj, and 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*, vol. 15, p. 100664, Aug. 2023, doi: 10.1016/j.clet.2023.100664.

[31] D. Onalo, S. Adedigba, O. Oloruntobi, F. Khan, L. A. James, and S. Butt, "Data-driven model for shear wave transit time prediction for formation evaluation," *J Pet Explor Prod Technol*, vol. 10, no. 4, pp. 1429–1447, Apr. 2020, doi: 10.1007/s13202-020-00843-2.

[32] Y. Zhang, H.-R. Zhong, Z.-Y. Wu, H. Zhou, and Q.-Y. Ma, "Improvement of petrophysical workflow for shear wave velocity prediction based on machine learning methods for complex carbonate reservoirs," *J Pet Sci Eng*, vol. 192, p. 107234, Sep. 2020, doi: 10.1016/j.petrol.2020.107234.

[33] J. Wang, J. Cao, and S. Yuan, "Shear wave velocity prediction based on adaptive particle swarm optimization optimized recurrent neural network," *J Pet Sci Eng*, vol. 194, p. 107466, Nov. 2020, doi: 10.1016/j.petrol.2020.107466.

[34] J. Liu, Z. Gui, G. Gao, Y. Li, Q. Wei, and Y. Liu, "Predicting Shear Wave Velocity Using a Convolutional Neural Network and Dual-Constraint Calculation for Anisotropic Parameters Incorporating Compressional and Shear Wave Velocities," *Processes*, vol. 11, no. 8, Aug. 2023, doi: 10.3390/pr11082356.

[35] S. Gomaa, J. S. Shahat, T. M. Aboul-Fotouh, and S. Khaled, "Neural Network Model for Predicting Shear Wave Velocity Using Well Logging Data," *Arab J Sci Eng*, Jun. 2024, doi: 10.1007/s13369-024-09150-y.

[36] L. O. Tedeschi and M. L. Galyean, "A practical method to account for outliers in simple linear regression using the median of slopes," *Sci Agric*, vol. 81, 2024, doi: 10.1590/1678-992x-2022-0209.

[37] F. A. Tyas, M. Nurayuni, and H. Rakhmawati, "Optimasi Algoritma K-Nearest Neighbors Berdasarkan Perbandingan Analisis Outlier (Berbasis Jarak, Kepadatan, LOF)," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 13, no. 2, pp. 108–115, May 2024, doi: 10.22146/jnteti.v13i2.9579.

[38] R. Hossain and D. Timmer, "Machine Learning Model Optimization with Hyper Parameter Tuning Approach," 2021.

[39] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.