Petroleum and Coal

Article **Open Access**

The Implementation of TPE-Bayesian Hyperparameter Optimization to Predict Shear Wave Velocity Using Machine Learning: Case Study From X Field in Malay Basin

Zaky Ahmad Riyadi, Maman Hermana Husen, Luluan Almana Lubis and Tsara Kamilah Ridwan

Department of Geoscience, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Malaysia

Received October 7, 2021; Accepted March 1, 2022

Abstract

Shear wave velocity is a fundamental parameter for geophysical, geomechanical, and petrophysical studies. To date, many wells are absent in shear wave velocity measurements due to the high cost and time-consuming to acquire. Many researchers have recently implemented a machine learning approach to estimate shear wave velocity because of its robustness in predicting a non-linear paradigm. However, many previous studies neglect the importance of optimizing machine learning's hyperparameter as many preferred to configure the hyperparameter manually, which can be less efficient, expensive to evaluate, and a time-consuming process. Optimizing the hyperparameters of machine learning is vital to obtain the maximum predictive potential. In this study, The Tree Parzen Estimator (TPE) Bayesian optimization algorithm was implemented to automatically fine-tuned the hyperparameters of Extreme Gradient Boosting (XGBoost), Random Forest (RF), and a Multi-Layered Perceptron Neural Network (MLPNN) algorithms. Subsequently, The effect of tuning hyperparameters on the performance of the technique is studied. Grid Search (GS) and Random Search (RS) algorithms are used to compare and evaluate the TPE-Bayesian optimization algorithm's performance. Common empirical relations for estimating shear wave velocity were also calculated to compare the performance between the empirical and machine learning approach. The results revealed that the TPE-Bayesian optimization algorithm managed to optimize the machine learning's hyperparameter and significantly improve the machine learning model's accuracy. Besides, the MLPNN algorithm optimized by the TPE-Bayesian optimization algorithm was able to outperform other presented methods. When computing power is limited, XGBoost with the implementation of TPE-Bayesian optimization is recommended.

Keywords: *Shear wave velocity estimation; Extreme Gradient Boosting; Random Forest; MLP Neural Network; Tree Parzen Estimator; Bayesian hyperparameter optimization.*

1. Introduction

Shear wave and compressional wave velocity are crucial for geophysical, geomechanical, and petrophysical studies. The application includes Amplitude Variation with Offset (AVO) [1], fluid type identification [2-3], pore type analysis [4], lithology identification [5-8], pore pressure prediction [9-11] and mechanical rock properties analysis [12-14]. However, many of the petro-physical logs are often missing shear wave velocity measurements. While compressional wave velocity is measured using sonic log tools directly, measuring shear wave velocity is more complicated. It requires measurement either from core sample analysis or Dipole Sonic Imager (DSI) tools. Core sample analysis is a time-consuming process and expensive. Furthermore, the DSI tools may not be applicable for all of the wells [15]. Therefore, a considerable amount of research has been made to develop shear wave velocity prediction using the available well logs measurements.

The two most common approach is based on empirical equation and artificial intelligence. Numerous studies have been made to predict shear wave velocity empirically. Pickett [5],

proposed an empirical equation to estimate a shear wave velocity based on the ratio of compressional and shear wave velocity for a specific lithology. Greenberg and Castagna [16], developed shear wave velocity empirical equations using the compressional wave velocity and the rock composition for water-saturated clean sands. Brocher [17], proposed a shear wave velocity empirical equation valid only when the compressional wave is between 1.5 to 7.5 km/sec. A comprehensive list of previous shear wave velocity empirical equations can be found in Oloruntobi and Butt [18]. Although empirical calculations are a more pronounced approach to predict shear wave velocity, empirical formulas are not accurate enough to meet the precision requirement for shear-wave prediction [19].

Generally, empirical equations are not globally applicable as it is established based on a specific formation. Therefore, it requires to be calibrated and validated with the field data. Furthermore, it may not be applicable for heterogeneous formations as it is highly dependent on the formation's lithology [18].

Over the last few years, the implementation of artificial intelligence systems has been studied extensively in petroleum industries and demonstrated as more superior to conventional approaches. Recent studies includes reservoir characterization analysis [20-21], reservoir parameter prediction [22-23], stress and fracture reservoir evaluation [24-25], reservoir facies models and identification [26-28], distribution of hydrocarbon reservoirs [29], reservoir uncertainty analysis [30] and as well as lithology identification [31-33]. Based on the previous studies, the artificial intelligence approach has proven to be a robust method in determining non-linear relations without a prior assumption of the procedure involved between parameters.

Several researchers have implemented artificial intelligence to predict shear wave velocity. Rezaee *et al.* [34], developed shear wave velocity prediction using fuzzy logic, neuro-fuzzy, and artificial neural network algorithm. In a study by Nourafkan [35], he integrates fuzzy logic algorithm with swarm intelligence (ant colony optimization). While Mehrgini [36], predicts the shear wave velocity using Elman neural network with the Levenberg-Marquardt algorithm. Based on the previous studies, the machine learning approach provides promising results in shear wave velocity prediction. However, many of the earlier studies neglect the importance of hyperparameter optimization.

Having optimized configuration of hyperparameters enables the machine learning algorithm to obtain the maximum predictive potential. However, finding the optimal configuration of hyperparameters is a challenge. Many earlier studies fine-tuned their machine learning's hyperparameters model based on the heuristic approach and fine-tuned it manually. This approach can be inefficient, time-consuming, and expensive to evaluate. Besides, machine learning algorithms such as extreme gradient boosting and the neural network have many hyperparameters required to be tuned. Therefore, manually evaluate different configurations of hyperparameter can be overwhelmed. Many researchers are employing an optimization algorithm to overcome the difficulty of optimizing machine learning's hyperparameters.

The most common approach to optimize a machine learning algorithm's hyperparameter is by implementing the Grid Search (GS) and Random Search (RS) algorithm. GS explores the optimal set of hyperparameters by trying different combinations of all the hyperparameters from a search space. Whereas RS randomly searches for a group of optimal hyper-parameters from a search space. However, both approaches have their drawbacks. GS suffers from the dimensionality constraint when a substantial configuration of hyperparameters is evaluated. At the same time, RS finds the optimal hyperparameter randomly where no intelligence is used. Over the past years, there has been a considerable amount of research for more intelligent hyperparameters optimization algorithm [37-40]. One of the most robust approaches is employing Tree Parzen Estimator (TPE) based on Bayesian optimization algorithm, which has proven to outperform other global optimization algorithms [41].

To the author's best knowledge, hyperparameter optimization algorithms have not been widely implemented, especially in shear wave velocity prediction. In this work, a workflow is proposed to predict shear wave velocity that implements TPE-Bayesian optimization algorithm to optimize the hyperparameter of presented machine learning models that includes; Extreme

Gradient Boosting (XGBoost), Random Forest (RF), and Multi-Layer Perceptron Neural Network (MLPNN) algorithms. The performances of the machine learning models are then compared. To evaluate the TPE-Bayesian optimization algorithm's performance, it is subsequently compared with the machine learning performance that utilized different hyperparameter algorithms such as GS and RS algorithm. In this study, petrophysical well log data were used extracted from three wells drilled in X field, Malay Basin. Empirical equations to estimate shear wave velocity, such as Pickett [5], Castagna [16], and Brocher [17], are used to compare the performance between machine learning and empirical relation approach. All of the models' prediction accuracy is evaluated based on Root Mean Squared Error (RMSE) and the coefficient of determination (R^2). This study's outcome is a proposed method for the researchers to better predict the shear wave velocity using a machine learning approach that utilized a hyperparameter optimization algorithm, where many of the previous studies may neglect its importance.

2. Study area and datasets analysis

In this work, the study area is from three adjacent wells located in X field, Malay Basin. Malay basin is situated offshore east of Peninsular Malaysia (Figure 1). It is one of the hydrocarbon-bearing basins in Southeast Asia. It comprises of gas-rich zone, flanked on either side and to the south by mixed oil/gas [42]. The Malay basin was formed during the early tertiary and underlined by the pre-tertiary basement of sedimentary, metamorphic, and igneous rocks [43]. It is the deepest continental extensional basin in the region and is located at the center of the Sundaland Cratonic core of South East Asia. The Malay basin strata are subdivided informally into Seismo-stratigraphic units and referred to as "Groups". Shown in Figure 2 is the Stratigraphy of the Malay basin from unit A (youngest) to M (oldest).

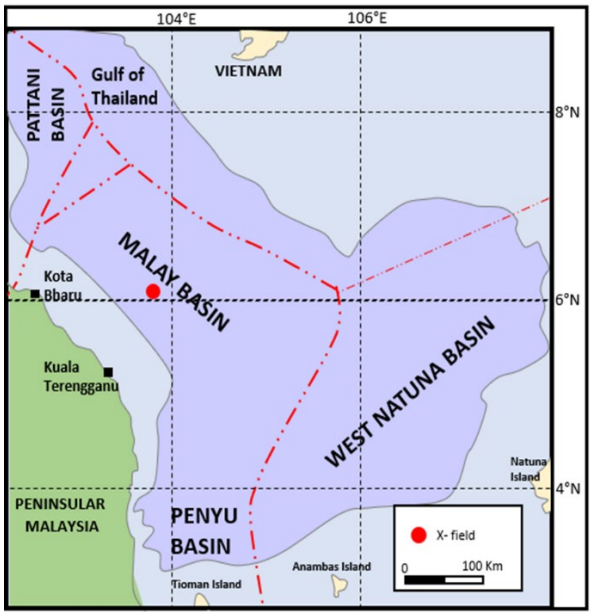

Figure 1. The geographic map location of the Malay Basin with its adjacent basin; Penyu, West Natuna, and Pattani Basin. Modified after [43]

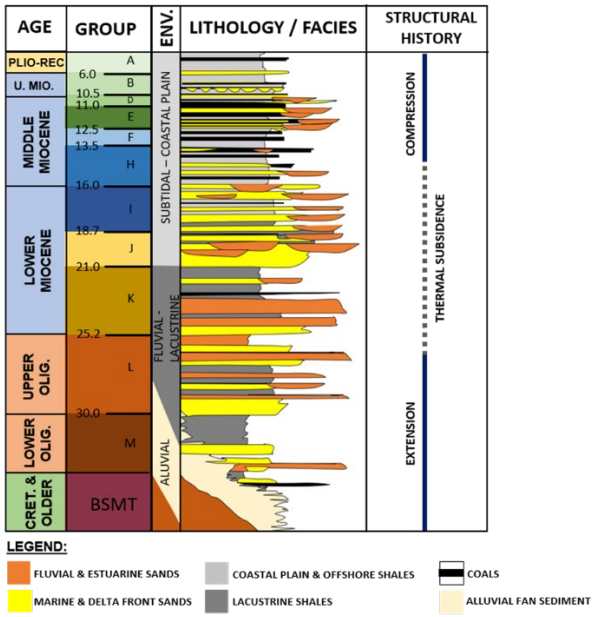

Figure 2. Generalized stratigraphy and structural history of the Malay Basin. Modified after [54]

The stratigraphic development of the Malay Basin is directly related to its structural evolution. In the late-Oligocene, it undergoes an extensional or syn-rift phase. In the early to middle Miocene, the thermal subsidence phase occurred, and subsequently, in the Late-Miocene, it undergoes a basin inversion [42]. Based on the rock core samples analysis from three wells, the primary lithology are consists of interbedding of Sandstone, Claystone, Siltstone, and thin layers of coal.

To keep the wells confidential, the wells are renamed into Well A, Well B, and Well C. There are 9758, 5779, and 1693 data points in Well A, Well B, and Well C, respectively. The petrophysical logs data that are utilized are Caliper (CAL), Bulk Density (RHOB), Gamma Ray (GR), Neutron Porosity (NPHI), Resistivity Deep (RD), Compressional Wave velocity (DTC), and Shear wave velocity (DTS) logs. Statistical data of these logs are presented in Table 1.

Table 1. Statistical analysis of the petrophysical logs in the studied well

Well Name	Statistical Index	Petrophysical well logs							
		Depth (m)	CAL (in)	RHOB (g/cc)	GR (API)	NPHI (v/v)	RD (Ohmm)	VP (km/s)	VS (km/s)
Well A	Mean	1892.59	13.39	2.46	100.34	0.27	2.83	2.73	1.31
	Min	1000.2	10.22	1.12	29.94	0.09	0.73	2.07	0.73
	Q1	1426.96	12.13	2.43	85.26	0.24	1.35	2.54	1.13
	Q2	1895.17	12.69	2.49	102.35	0.27	1.72	2.7	1.26
	Q3	2338.24	14.88	2.54	115.72	0.3	2.52	2.9	1.47
	Max	2709.98	19.74	2.77	237.17	0.57	308.25	3.75	2.09
Well B	Mean	1450.57	11.81	2.42	61.18	0.25	2.11	2.89	1.38
	Min	946.4	11.54	1.21	11.24	0.08	0.64	2.03	0.8
	Q1	1238.02	11.72	2.34	54.41	0.22	1.11	2.67	1.18
	Q2	1458.16	11.78	2.47	62.39	0.24	1.46	2.91	1.32
	Q3	1678.31	11.87	2.53	68.76	0.27	2.53	3.13	1.59
	Max	1907.59	14.04	2.68	100.52	0.64	30.67	3.55	2.02
Well C	Mean	1674.31	12.42	2.41	57.53	0.27	5.61	3	1.5
	Min	1545.05	12.25	1.18	8.44	0.06	0.87	1.99	0.68
	Q1	1609.52	12.39	2.37	50.37	0.21	1.45	2.93	1.39
	Q2	1673.98	12.41	2.52	61.56	0.25	1.87	3.06	1.5
	Q3	1739.51	12.42	2.56	67.48	0.27	2.88	3.18	1.71
	Max	1803.98	14.35	2.71	95.22	0.85	357.69	4.6	2.08

3. Methodology

Numerous data samples are mandatory to build an accurate machine learning model. Therefore, because the higher number of data samples are available in Well A and Well B, both wells were used for the training dataset. Whereas, Well C is used for testing datasets. An attribute of $\frac{V_p}{\sqrt{RHOB}}$ is also used as an additional input for machine learning and denoted as VPDEN.

The methodology of the study mainly consists of four steps procedure: (i) preprocessing the wireline log data, (ii) searching the optimal configuration hyperparameters using optimization algorithms, (iii) retraining the machine learning model using the optimized configuration of hyperparameters based on the optimization algorithms, and lastly, (iv) evaluate the performance. The workflow of the study is illustrated in Figure 3. Firstly, in data preprocessing, elimination, and standardization of the petrophysical logs, data are performed. Subsequently, Tree Parzen Estimator Bayesian optimization algorithm is utilized to fine-tune the machine learning's hyperparameters automatically based on the defined search space. Grid Search and Random Search algorithms are used to compare and evaluate the effectiveness of the TPE Bayesian algorithm's performance.

One hundred evaluations were set for the Bayesian optimization algorithm to search the optimized hyperparameters configuration. The list of hyperparameters and their range setting are shown in Table 2. Subsequently, using the hyperparameters configuration based on the optimization algorithm, the presented machine learnings are retrained to predict shear wave velocity from the test dataset. Finally, the accuracy of the three proposed machine learning algorithms with different hyperparameter optimization algorithms are evaluated using RMSE and R^2 score. The shear wave velocity empirical equation includes Pickett [5], Castagna *et al.* [16],

and Brocher [17] are calculated to compare and evaluate the overall performance of the machine learning method.

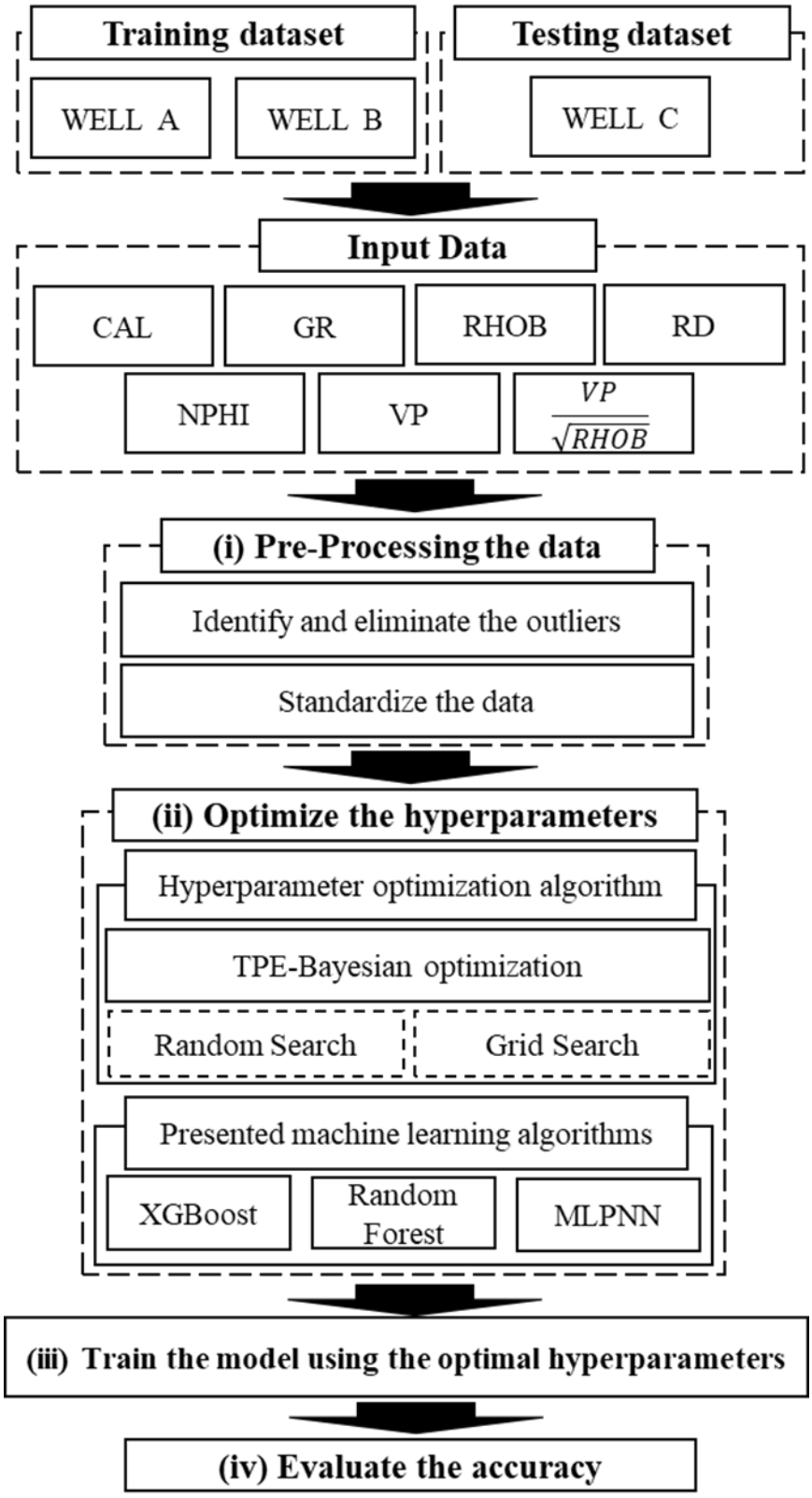

Figure 3. The workflow of the study

For interest, all the methodologies are performed using intel Xeon E5-2643 running at 3.30 GHz and with 48 GB of RAM. Furthermore, all of the methodologies are utilizing the library packages on python version 3.8.3. To perform the TPE-Bayesian optimization algorithm, the Hyperopt library was used. Subsequently, Grid Seach and Random Search algorithm are performed using the Sklearn library. To developed the machine learning models, the Tensorflow library was used to create the MLPNN model. Meanwhile, XGBoost and Random forest models are developed using the XGBoost and SKlearn library, respectively.

3.1. Eliminating the outliers

The quality of logging data depends on environmental conditions, poor instrument calibration, and human error [44-45]. Therefore, it is crucial to filter out some of these invalid measurements. For this study, a quality check is performed by identifying the bad hole based on caliper and bulk density correction logs. All of the bad hole measurements are then removed from the training and testing dataset.

3.2. Data standardization

Data standardization is commonly carried out before training any machine learning model, especially training a deep learning model. Many machine learning algorithms do not perform well when the numerical input attributes have different large scales [46]. Since each of the petrophysical logs data have a different scale or unit, standardization of the dataset is necessary. In this work, the standard score equation (equation 1) is used to standardize the scale of petrophysical logs. Where μ is mean, and σ is standard deviation.

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

3.3. Hyperparameters optimization method

3.3.1. Hyperparameters

Hyperparameters are parameters that determine the learning process of a machine learning algorithm. It does not learn directly from the input and requires to be tuned before training the model. The problem of hyper-parameters optimization can be expressed mathematically [47] as trying to locate an input x^* in a domain X ($x^* \in X$) to an unknown objective function f ($f : X \to \mathbb{R}$) which minimize the value of this function over X such that:

$$x^* \in \underset{x \in X}{arg\ min}\{f(x)\} \quad (2)$$

In general, the complexity or regularization is controlled by machine learning algorithms' hyperparameters [48]. Therefore, tuning and finding the optimal configuration of hyperparameters is essential. Manually tuned the hyperparameters of machine learning is a challenge as it depends on the trial-and-error approach and experience [40]. In addition, retraining the machine learning model to evaluate the effect changes from a different set of hyperparameter configurations is very costly.

3.3.2 Bayesian hyperparameters optimization

The Bayesian optimization algorithm is commonly used for optimizing the hyperparameter of the machine learning models. It is a robust method for optimizing the objective functions that are expensive to evaluate [38,48]. The Bayesian optimization algorithm was able to outperform grid search and random search in the hyper-parameter optimization tuning. Previous studies show that Bayesian optimization requires a smaller iteration to converge a solution, and it is more generalized on the testing dataset than GS and RS algorithms [40]. Bayesian optimization algorithm builds a probabilistic surrogate model based on the target value's previous evaluation result to minimize the objective function. Like any other hyperparameter optimization algorithm, search space is required to be defined as a bounded boundary to restrict the objective function's search limit.

In the present study, the search space for all of the presented optimization algorithms is shown in Table 2. To evaluate the performance of the TPE-Bayesian optimization (TPE-BO) algorithm, it is compared with Grid Search (GS) and Random Search (RS). The GS and RS algorithm are become more expensive to evaluate when the search space is expansive. Therefore, to reduce the computational cost, the search space's increment was reduced while the hyperparameters configuration and range setting remain. Root Means Squared Error (RMSE) was used to evaluate the grid search and random search for every iteration. Furthermore, seven-fold cross-validations were used as a resampling technique to evaluate the machine learning models. To evaluate hyperparameter optimization's significance, we also compared with the machine learning models without utilizing any hyperparameter optimization.

Table 2. The tuned hyperparameters and their range setting for each of the proposed machine learning algorithms

Machine Learning Algorithm	Tuned Hyperparameters	TPE-BO algorithm			GS and RS algorithm		Default hyperparameters
		Range Setting	Step	Distribution	Range Setting	Step	Value
XGBoost	Max depth	1 – 10	1	Choice	1-10	2	6
	Learning rate	0.01 - 0.1	0.001	Quniform	0.01-0.1	10	0.01
	Colsample by level	0.1 – 1	-	Uniform	0.1-1	0.2	1
	N estimators	1000 - 3000	-	Uniform	1000-3000	500	0
	Reg alpha	1 – 10	1	Quniform	1-10	2	0
	subsample	0.1 - 1	0.1	Quniform	0.1-1	0.2	1
RF	Min samples split	0 – 1	-	Uniform	0-0.5	0.1	2
	Min samples leaf	0 – 0.5	-	Uniform	0-0.5	0.1	1
	N estimators	100 – 500	1	Choice	100-500	100	100
	Max features	sqrt, log2, 0.2, 0.5, 0.8	-	Choice	sqrt, log2, 0.2, 0.5, 0.8	-	'auto'
	Max depth	1-10	-	Uniform	1-10	2	None
MLPNN	Batch Size	16,32,64,128	-	Choice	64,128	-	128
	Optimizer function	Adam, RMSprop, SGD	-	Choice	Adam, RMSprop, SGD	-	Adam
	Learning rate	0.001 – 0.01	0.001	Quniform	0.001,0.01	-	0.01
	Activation function	Relu, Sigmoid	-	Choice	Relu, Sigmoid	-	Relu
	Number of units	3 – 30	-	5, 10, 15, 25, 30	5, 10, 15, 25, 30	-	3,12,35

3.3.3. Sequential Model-Based Optimization (SMBO)

Sequential Model-Based Optimization (SMBO) is the formalization of Bayesian optimization. Essentially, SMBO runs trials successively, and in every trial, it tries to look for better hyperparameters by implementing Bayesian reasoning and updating a surrogate (probability) model. The procedure and pseudocode of SMBO are shown in Figure 4 [48]. SMBO firstly mapped the hyperparameter's configuration λ into a Surrogate model S to the loss function L and history H, to record the hyperparameter's configuration and loss value. Then SMBO iterates the following steps:

1. Looks for the local optimal hyper-parameter setting based on the present model S_{t-1}.
2. Calculate the loss c under the settings λ^*.
3. Update H to store λ^* and the corresponding loss c.
4. Build a new model S_t correspond to the updated record H.
5. Iteration stop till it reaches the pre-determined maximum iteration number.
6. The SMBO outputs the global optimal hyper-parameter settings with the minimum c.

Algorithm 1: Sequential Model-Based Optimization (SMBO)

Initialization $S_0; H = \emptyset$
For t = 1 to T do
$\lambda^* = argmin S_{t-1}(\lambda)$
C = L(λ^*)
$H = H \cup (\lambda^*, c)$
Fit new model S_t according to updated H
end for
Return λ with minimum c in H

Figure 4. Pseudocode of Sequential Model-Based Optimization

In this study, Root Mean Squared Error (RMSE) was used as a metric to measure the objective function's loss score. The Tree Parzen Estimator was used as the surrogate model to evaluate the different configurations of hyperparameters. One hundred were set as the maximum iteration or evaluation for the TPE-Bayesian optimization to search the hyperparameters' optimized configuration based on the restricted search space (Table 3).

Table 3. The chosen hyperparameters values for XGBoost, RF, and MLPNN algorithm based on TPE Bayesian optimization

Machine Learning Algorithm	Hyperparameters	TPE-BO			GS	RS
XGBoost	Max depth	1			2	2
	Learning rate	0.057			0.01	0.01
	Colsample by level	0.477			0.6	0.4
	N estimators	1316.2			3000	1500
	subsample	0.7			0.2	0.4
	reg alpha	2			2	
RF	Min Sample Split	0.07			0.2	0.5
	Min Sample Leaf	0.028			0.01	0.02
	N Estimators	321			100	100
	Max Features	sqrt			0.5	0.5
	Max Depth	5.025			8	8
MLPNN	Batch Size	128			128	128
	Optimizer function	RMSprop			Adam	Adam
	Layers	1	2	3	1-3	1-3
	Learning rate	0.1	0.001	0.01	0.01	0.01
	Activation function	Sigmoid	ReLu	ReLu	Sigmoid	Sigmoid
	Number of units	12	11	5	15	15

3.3.4. Tree Parzen Estimator (TPE) Surrogate model

As aforementioned, SMBO builds a surrogate model based on the previous evaluation result. The Surrogate or probability model is a representation of the objective function. It is used for training different sets of hyperparameter settings to predict the performance of the learning algorithm based on a given hyperparameters setting and dataset. It is cheaper to evaluate

in a surrogate model compared to the objective function [40]. In the SMBO framework algorithm, there are three common variants of the surrogate model: Gaussian Process (GP), Tree Parzen Estimators (TPE), and Random Forest Regressions (RFR).

In this study, TPE was used as the surrogate model. TPE [40] is an unconventional approach to SMBO. While the GP model the predictive distribution over the objective function, TPE creates two hierarchical processes $\ell(\lambda)$ and $g(\lambda)$ acting as generative models for all domain variables. These processes model the domain variables when the objective function is in the range of defined threshold c^*:

$$p_s(c|\lambda) = \begin{cases} l(\lambda), & if\ C < C^*, \\ g(\lambda), & if\ C \geq C^*. \end{cases} \quad (3)$$

where $\ell(\lambda)$ is the density estimate built from the observations based on the value of loss function c, which is less than the threshold c^* in H and g(λ) is formed from the remaining observations. To determine λ^*, corresponding to the current model S_t for the t times loop, SMBO is using the acquisition function. The two most common types of acquisition functions are Expected Improvement (EI) and improvement (I). EI can be expressed as:

$$\mathrm{EI}(\lambda) = \int_{-\infty}^{C_{min}} \max\left(C_{min} - c(\lambda), 0\right) \cdot p_{S_t}(c|\lambda) dc \quad (4)$$

where, C_{min} and $c(\lambda)$ denote the current minimum loss in H and the loss under the hyper-parameter setting λ, respectively. The improvement I can be expressed as:

$$\mathrm{I}(\lambda) = \max(c_{min} - c(\lambda), 0) \quad (5)$$

A more detailed explanation of the parameter adjustment process of TPE is discussed by Bergstra *et al.* [40].

3.4. Shear wave velocity estimation using Intelligence Methods

3.4.1. Extreme gradient boosting (XGBoost)

In recent years, Extreme gradient boosting or XGBoost has become a popular machine learning algorithm because of its efficiency, rapidness, and scalability [49]. It is a novel version of gradient boosting and was proposed by Chen and Guestrin [49]. XGBoost is a supervised algorithm that is based on boosting, one of the ensemble learning methods. The main idea of boosting is to compile the base learner sequentially, and each tries to correct the predecessors to be a strong learner. The process of XGBoost is to fits an additive base learner that minimizes the loss function by letting the loss function decrease in the direction of its gradient [50]. Chen and Guestrin have improved the gradient boosting algorithm by implementing a regularization term into the objective function. Regularization is a technique used to prevent the model from overfitting by adding a penalty term to the cost function. Additionally, XGBoost performs second-order Taylor expansion to the objective function, allowing XGBoost to define the loss function more accurately. XGBoost objective function is defined in the following equation:

$$Obj^t = \sum_{k=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \Omega(f_t) \quad (6)$$

where, $\hat{y}_i^{(t)}$ and f_t denote given data samples, the prediction at the t^{th} and the structure of a decision tree (the base learner), respectively. $\sum_{k=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right)$ is the training loss function that describes who well the fit is with training data. The regularization $\Omega(f_t)$ for penalizing the complexity of the model is calculated using:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2 \quad (7)$$

where γ is the penalty coefficient, the minimum loss is needed to partition the leaf node further. λ is the regularization hyperparameter, and ω is the vector of scores in the leaves. The function of the model after the prediction at the t^{th} is calculated by adding the prediction at $(t\text{-}1)^{\text{th}}$ with a new decision tree. The objective function is subsequently updated to:

$$Obj^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (8)$$

where $\hat{y}_i^{(t-1)}$ denotes the (t-1)th iteration prediction function, and $f_t(x_i)$ denotes a new decision tree.

XGBoost approximates using the 2nd-order Taylor expansion to optimize the objective in the general setting efficiently.

$$Obj^{(t)} \cong \sum_{i=1}^{n} [L(y_i, \hat{y}_i^{(t-1)} + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) + \Omega(f_t) \quad (9)$$

where $g_i = \partial_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)})$ and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)})$ are 1st and 2nd order gradient statistics of the loss function, respectively. For more thorough explanations of the XGBoost algorithm is discussed by Chen and Guestrin [49].

3.4.2. Random forest regression

Random Forest (RF) algorithm was introduced by Breiman [51]. It has been widely implemented to solve classification and regression problems because of its flexibility and fast machine learning algorithm. RF is an ensemble learning method that uses a bagging or bootstrap aggregation approach to compile the base learners. The general idea of bagging is to train the base learner independently and use the average estimation for the result [52]. It is shown in Figure 5, the workflow diagram of random forest regression. Firstly, the process of RF regression creates the number of different bootstrap samples by resampling the training dataset randomly with replacement. Hence, to increase the diversity of the decision trees (base learner) during training. Then RF construct numbers of regression trees (K) and take the average results $H(x)$. The final output result of the regression model is shown the following equation

$$H(x) = \frac{1}{K} \sum_{i=1}^{K} h_i(x) \quad (10)$$

where $H_i(x)$ is the output of the i-th regression tree (T_i)

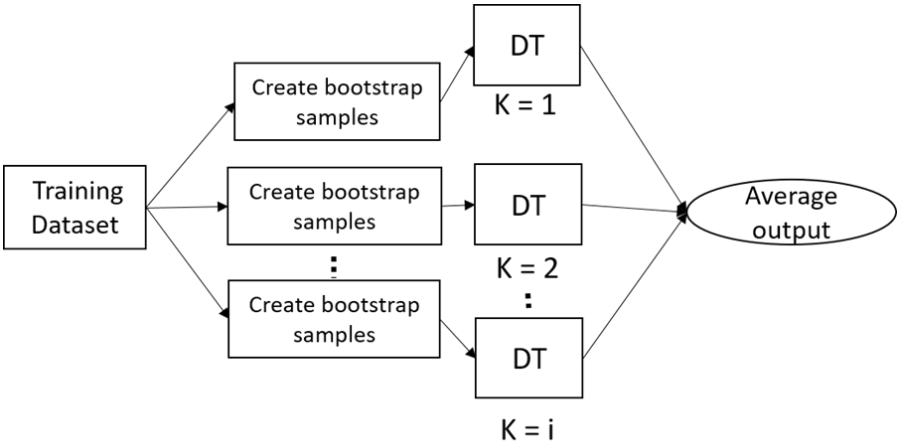

Figure 5. Workflow diagram of random forest regression

3.4.3. Multilayer Perceptron (MLP) Neural Network

Artificial Neural Network (ANN) has become more prevalent in recent years because of its ability to determine the non-linear relation without a prior assumption of the procedure involved between parameters. A traditional ANN architecture is known as a multilayer perceptron (MLP). MLP is one of the extensively used ANN architecture [53]. It is a feedforward neural network that consists of input, hidden, and output layers. Shown in Figure 6 is the illustration

of the MLP neural network. These layers are fully connected, and each layer is associated with the weight value. Multi-Layer Perceptron Neural Network (MLPNN) commonly employs the backpropagation or gradient descent technique to adjust and optimize the weight value based on the actual and predicted values' loss function. The process is repeated until it reached maximum iteration to obtain the minimum value of the loss function and the optimal weight values.

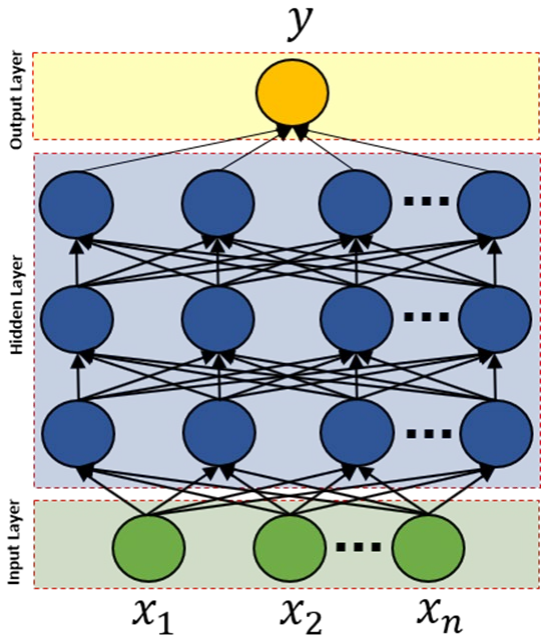

Figure 6. MLP architecture used in the study

3.5. Empirical calculation

In this work, three common shear wave velocity estimation empirical formula includes Pickett [5], Castagna *et al.* [16], and Brocher [17] are used to compare the performance of the machine learning model. The equations are shown in the following, where v_p and v_s is in km/s unit.

Pickett [5]:

$$v_s = \frac{v_p}{1.75} \quad (12)$$

Castagna *et al.* [16]:

$$v_s = 0.80416v_p - 0.85588 \quad (13)$$

Brocher [17]:

$$v_s = 0.7858 - 1.2344\, v_p + 0.7949v_p^2 - 0.1238\, v_p^3 + 0.0064v_p^4 \quad (14)$$

3.6. Performance metrics

To analyze the performance of the estimation, metrics such as Root Mean Squared Error (RMSE) and coefficient of determination (R^2) were employed. RMSE value that is close to 0 indicates that the error is prediction is low. RMSE is calculated by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (15)$$

R^2 value closer to 1 indicates that the predicted value is better for data fitting. The equation for R^2 is shown in the following.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \quad (16)$$

4. Results and discussion

In this section, the presented machine learning models: XGBoost, RF, and MLPNN, are compared systematically using three different hyperparameter optimization algorithms: Tree Parzen Estimator – Bayesian Optimization (TPE-BO), Grid Search (GS), and Random Search (RS), with respect to the accuracy and rapidity. Furthermore, to evaluate the machine learning method's overall performance, common empirical relations for shear wave velocity estimation such as Picket [5], Castagna *et al.* [16], and Brocher [17] were used. To illustrate the performance, we utilized the traditional regression performance metrics includes RMSE and R^2. Table 4 summarizes the presented model's performance and evaluation time based on the test dataset. Additionally, the total evaluation time, RMSE, R^2, and is ranked and illustrated in Figures 8, 9, and 10, respectively.

Table 4. Summarized the performance of Machine Learning method

Machine Learning algorithm	Optimization Algorithm	Evaluation Index		
		RMSE	R^2	Elapsed time (sec)
XGBoost	TPE-Bayeisan optimization	0.120	0.899	130.9
	Grid Search	0.108	0.83	16116.2
	Random Search	0.108	0.828	149.6
	-	0.135	0.729	0.6
RF	TPE-Bayeisan optimization	0.094	0.884	306.9
	Grid Search	0.113	0.81	5892
	Random Search	0.116	0.802	49.6
	-	0.136	0.728	6.59
MLPNN	TPE-Bayeisan optimization	0.076	0.918	1068.6
	Grid Search	0.119	0.789	6448.3
	Random Search	0.112	0.779	1879.0
	-	0.153	0.652	3.7

*** where XGBoost is Extreme Gradient Boosting Algorithm, RF is Random Forest algorithm, MLPNN is Multi-Layer Perceptron Neural Network, and TPE is Tree Parzen Estimator*

4.1. Hyper-parameters tuning performance

As aforementioned, in this study, the presented machine learning's hyperparameters are tuned using the TPE-BO algorithm. Subsequently, it was compared with the RS and GS algorithms to evaluate the TPE-BO algorithm's performance. The elapsed time for all machine learning models was recorded to evaluate and compare the presented models' rapidity.

The TPE-BO algorithm evaluates the optimal hyperparameter configuration based on the loss function of every objective function. In this study, the RMSE metric is used as the loss function to evaluate every objective function. The objective function lowest RMSE score is then chosen as the best model with the best configuration of hyperparameter.

Table 4 summarized the performance metrics for all of the presented methods. The result revealed that the three hyperparameter optimization algorithms significantly improve the machine learning algorithm's accuracy. Furthermore, the TPE-BO algorithm performed significantly better in all of the presented machine learning models compared to the GS and RS algorithms. Meanwhile, the GS algorithm shows a slightly better improvement than the RS algorithm.

The less effective performance by the GS and RS algorithms is expected as neither algorithm uses any correlation between different configurations of hyperparameters. Furthermore,

both algorithms are using a smaller search space than the TPE-BO algorithm. Despite their drawbacks, both algorithms performed better than without using any hyperparameter optimization algorithm at all. Manually tuned hyperparameter can take a long time as it requires a trials and error approach. Furthermore, it is expensive to evaluate as it requires retraining multiple trials.

The illustration from Figure 7 shows the loss score of every evaluation of the TPE-BO algorithm for each of the presented machine learning models. The diagram shows that the TPE-BO algorithm was able to find the lowest loss score of an objective function at earlier evaluation than the number of maximum evaluation. Once the optimal configuration of the hyperparameter is found, no further improvement will be acquired. Besides, It is also observed that the TPE-BO algorithm performed better in the MLPNN algorithm than RF and XGBoost algorithms. In the MLPNN algorithm, the lowest loss score of the objective function based on the TPE-BO algorithm is 0.0062. While in contrast, RF and XGBoost algorithm are 0.009 and 0.01, respectively. Based on the loss score for every TPE-BO evaluation, the MLPNN algorithm was better optimized in minimizing the objective function than the XGBoost and RF algorithm. It can be seen from the gradual decrease of the loss score for the subsequent evaluation. In contrast, in the RF algorithm, TPE-BO algorithm seems to have difficulties in minimizing the loss score.

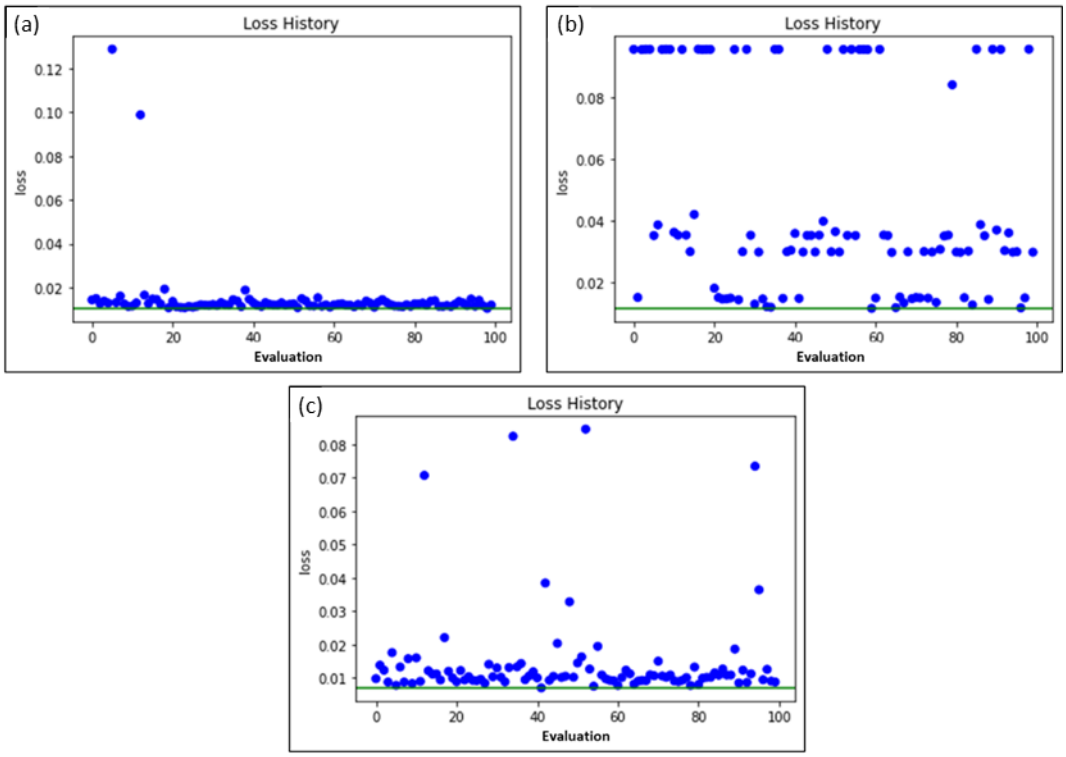

Figure 7. The result of each iteration and loss score based on A) XGBoost, B) RF, and C) MLPNN

Shown in Table 4 are the evaluation metrics score and the elapsed time to complete the maximum evaluation. Figure 8 ranked the elapsed time to complete the evaluation of every machine learning model. The result revealed that GS and RS algorithms are generally slower to evaluate despite the smaller search space than in the TPE-BO algorithm.

The RS algorithm is generally faster to evaluate compared to GS and TPE-BO algorithm. In RS, it took 149.6, 49.6, and 1879.0 seconds to search and optimize the hyperparameter of the XGBoost, RF, and MLPNN algorithm, respectively. GS took the slowest time, where it took 16.116, 5.892, and 6448.3 seconds to optimize the hyperparameter of XGBoost, RF, and MLPNN algorithms, respectively. Both GS and RS algorithms did not use any intelligence in selecting different hyperparameter configurations. The GS algorithm took the longest to evaluate as it requires to explores every hyperparameter by trying different combinations of all

the hyperparameters from a search space. In comparison, the RS algorithm searches the configuration of hyperparameters based on the number of iteration that has been set and randomly selects the configuration of hyperparameter from the defined search space. Hence, the quickest to evaluate.

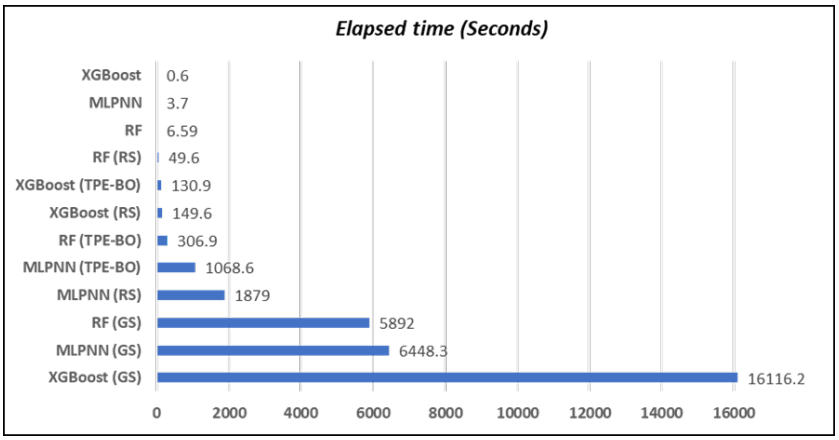

Figure 8. Ranked all of the models based on the time taken to complete

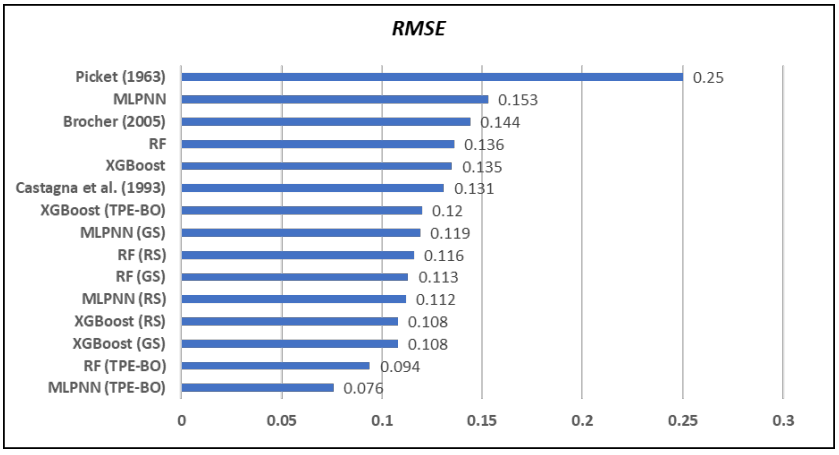

Figure 9. Ranked the performance of RMSE score for all of the presented models. A lower RMSE score indicates a better correlation

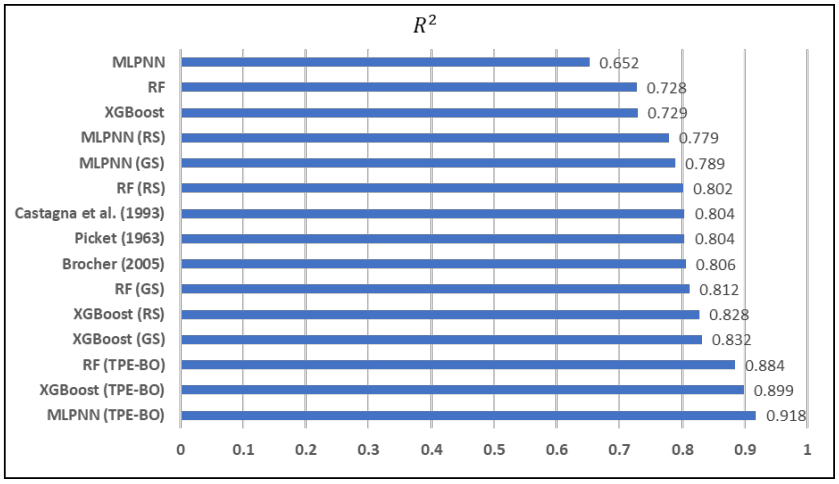

Figure 10. Ranked the performance of R^2 score for all of the presented models. Higher R^2 score indicate better correlation

Observing the evaluation time taken for the TPE-BO algorithm, the XGBoost algorithm is the fastest to evaluate (130.87 Sec), while the MLPNN algorithm took the longest (1068.6 Sec). XGBoost algorithm is known as the improved algorithm of the gradient boosting algorithm. It mainly designs for efficiency and performance. Therefore, it was able to optimize faster compared to RF and MLPNN algorithm. In contrast, the MLPNN algorithm is more complex to compute and requires higher computing power.

Optimizing the hyperparameter of a machine learning algorithm improved the accuracy of the machine learning models. Ideally, one should be able to list all of the hyperparameters with their maximum range values. Thereby ensuring the hyperparameter optimization algorithm can obtain the best values for every configuration of hyperparameter. However, this is not the case. Based on the observation, the three hyperparameters optimization algorithm's performance is affected by the search space's size, especially GS and RS. However, comparing the three optimization algorithms, the TPE-BO algorithm has shown to be less affected by the search space's size as it can search in a more expansive search space while still maintaining accuracy and evaluation time.

The most crucial step to obtain the best configuration of hyperparameters when utilizing the TPE Bayesian optimization algorithm is to configure the hyperparameter's search space. Machine learning algorithm such as XGBoost and MLPNN algorithm has many hyperparameters that would significantly affect the performance. Therefore, to optimize the hyperparameter's search space, it is essential to have the machine learning algorithm's domain knowledge and background. Furthermore, conducting multiple evaluations of TPE-Bayesian optimization is necessary to conceptualize the optimized search space. It is accomplished by observing the optimized hyperparameter's configuration based on multiple evaluations. Subsequently, gradually reduce the range setting of the hyperparameter based on the result of the previous evaluations. Therefore, it allows the TPE-BO algorithm to evaluate in a smaller search space and explore in a more optimized range setting for each hyperparameter.

4.2. Overall performance measures

The summary of the RMSE and R^2 score for the testing dataset of the machine learning models is shown in Table 4. Meanwhile in Table 5 shows the RMSE and R^2 score based on the empirical relation approaches. Figures 11, 12, 13, and 14 compare every presented model with the measured shear wave velocity.

Table 5. Summarized the performance of empirical relations

Empirical equations	Evaluation Index	
	RMSE	R^2
Picket [5]	0.25	0.804
Castagna *et al.* [16]	0.131	0.804
Brocher [17]	0.144	0.806

In general, all of the presented methods closely follow the measured shear wave velocity trend. Based on the RMSE and R^2, the MLPNN optimized by the TPE-BO algorithm (RMSE= 0.076 & R^2= 0.918) is shown as the best performance compared to the other approaches. Furthermore, the overall top-order performance is followed by XGBoost with TPE-BO (RMSE= 0.119 & R^2= 0.899), RF with TPE-BO (RMSE= 0.094 & R^2= 0.884), and RF with GS (RMSE= 0.113 & R^2= 0.81) algorithm.

Comparing the three machine learning algorithm's performance, optimized by the TPE-BO algorithm, the XGBoost algorithm could better predict the lower and higher extreme values (shown in Figure 11) where lower frequency data samples points are available. In contrast, the RF algorithm performed better at the higher frequency data samples (shown in Figure 12). The overall best performance appears to be the MLPNN algorithm. It was able to nearly follow the trend of the measured shear wave velocity throughout the depth intervals.

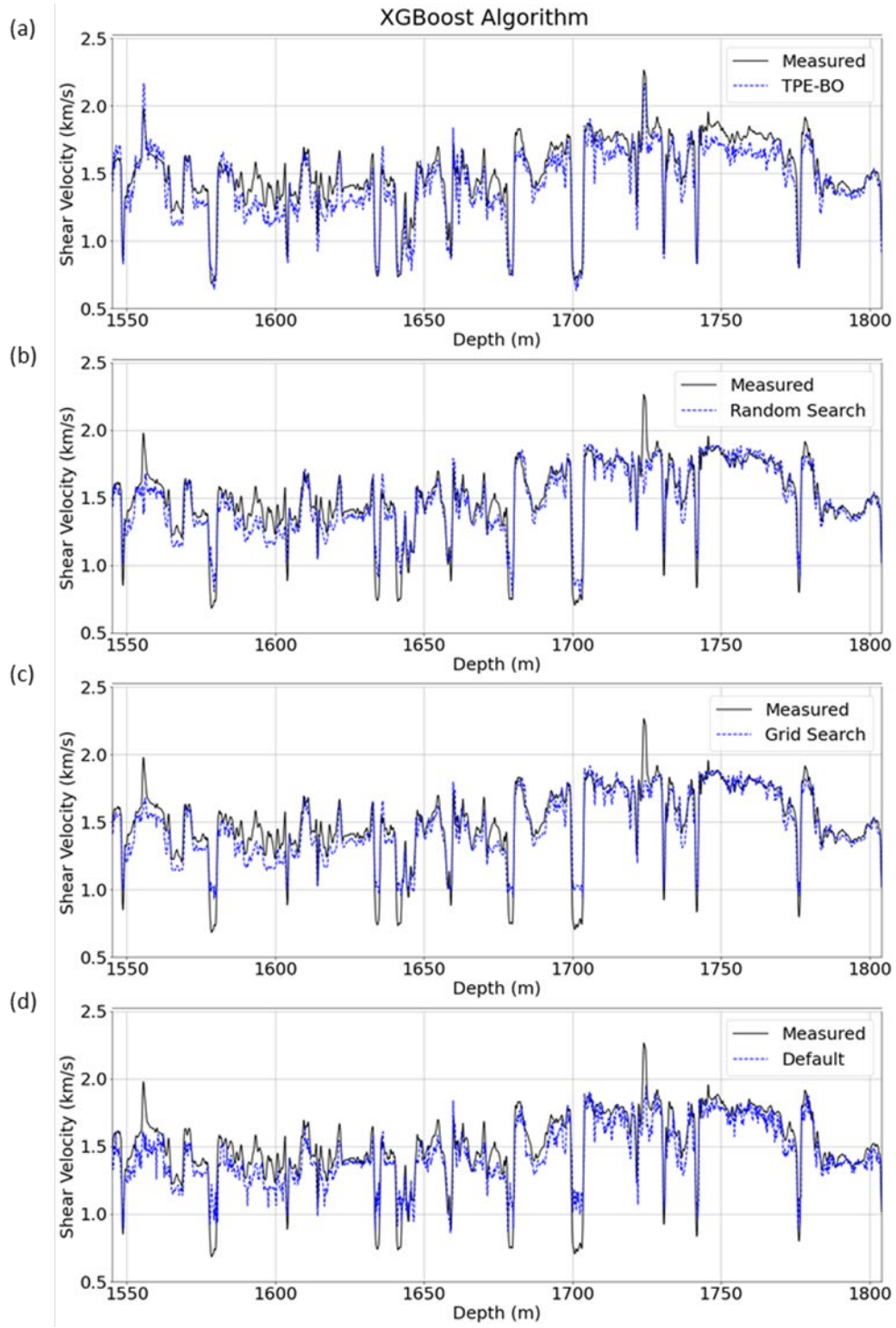

Figure 11. Comparison of XGBoost models with the measured shear wave velocity. (a) Measured Shear velocity vs. XGBoost with the implementation of TPE-BO algorithm, (b) Measured Shear velocity vs. XGBoost with the implementation of Random Search algorithm, (c) Measured Shear velocity vs. XGBoost with the implementation of Grid Search algorithm, and (d) Measured Shear velocity vs. XGBoost without hyperparameter optimization

All of the RMSE scores for all presented models are ranked and shown in Figure 9. It illustrated that the Castagna *et al.* [16] equation performed better than the machine learning methods that did not utilize hyperparameter optimization algorithm. Furthermore, based on the

value of R^2, all three empirical relation approaches performed better than the RF and MLPNN algorithms that utilized the RS algorithm, the MLPNN algorithm that utilized the GS algorithm, and all machine learning algorithms that did not utilize any optimization algorithm.

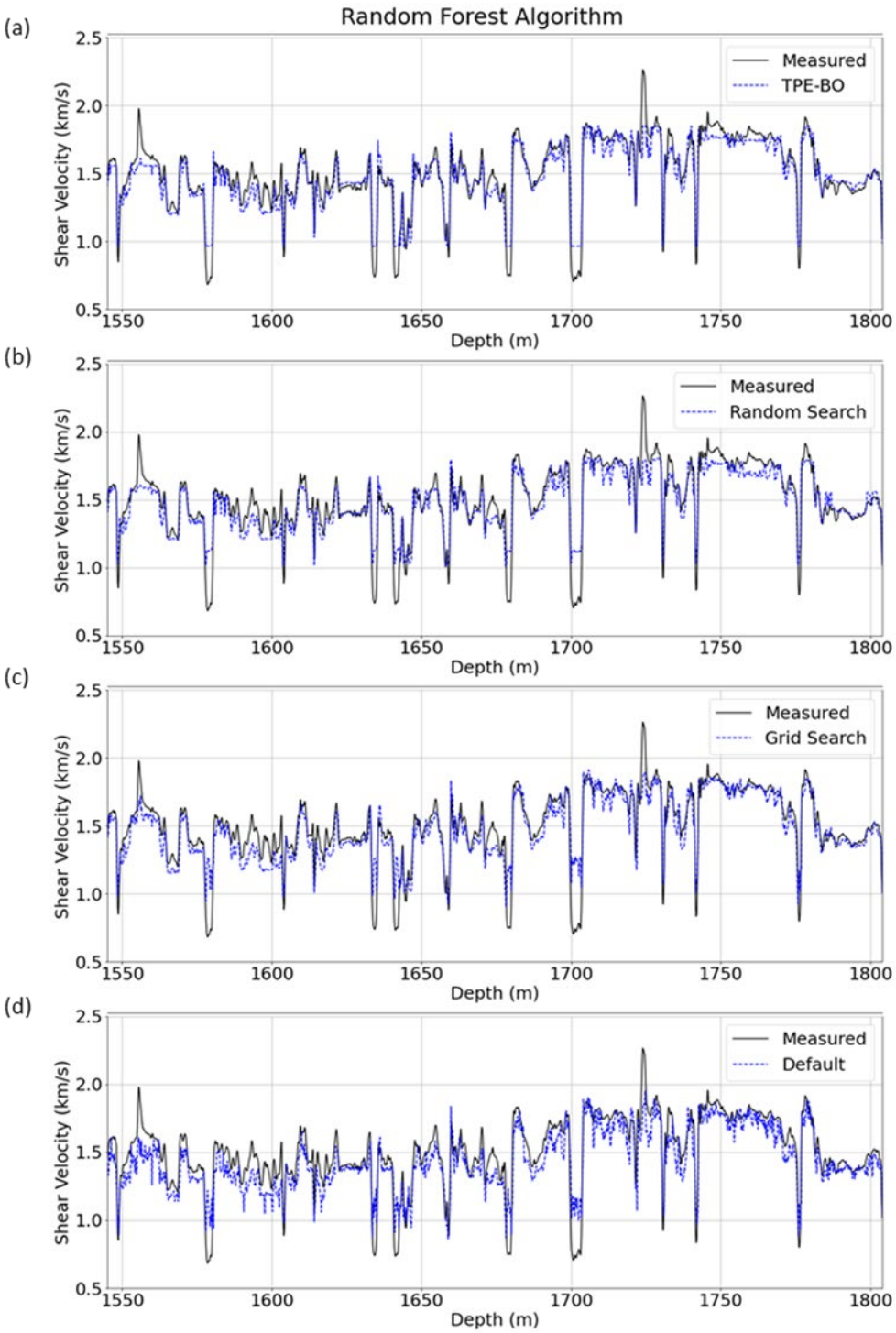

Figure 12: Comparison of Random Forest models with the measured shear wave velocity. (a) Measured Shear velocity vs. Random Forest with the implementation of TPE-BO algorithm, (b) Measured Shear velocity vs. Random Forest with the implementation of Random Search algorithm, (c) Measured Shear velocity vs. Random Forest with the implementation of Grid Search algorithm, and (d) Measured Shear velocity vs. Random Forest without hyperparameter optimization

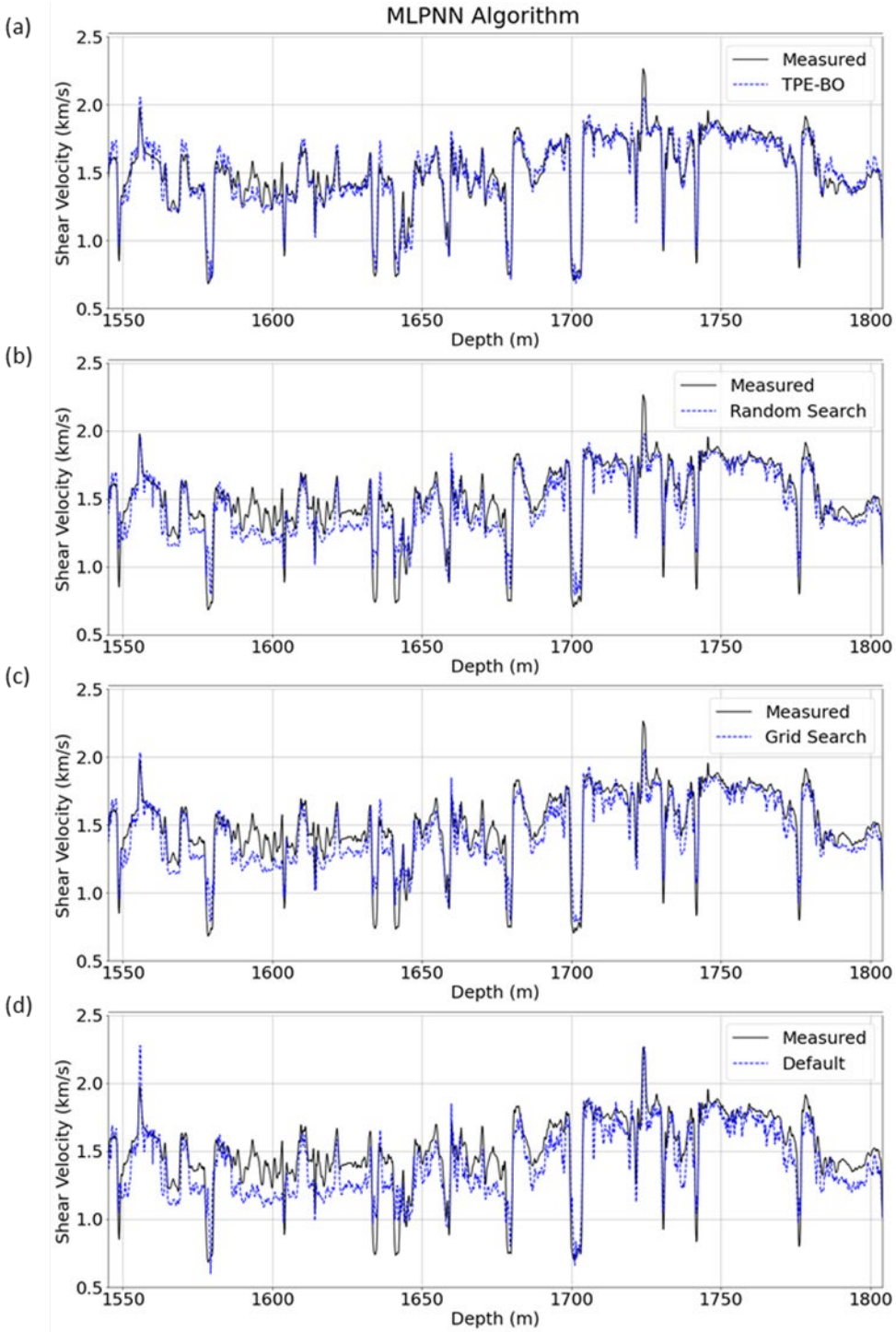

Figure 13. Comparison of MLPNN models with the measured shear wave velocity. (a) Measured Shear velocity vs. MLPNN with the implementation of TPE-BO algorithm, (b) Measured Shear velocity vs. MLPNN with the implementation of Random Search algorithm, (c) Measured Shear velocity vs. MLPNN with the implementation of Grid Search algorithm, and (d) Measured Shear velocity vs. MLPNN without hyperparameter optimization

Comparing the three empirical relations used in this study, all empirical relations followed the measured shear wave velocity trend. Furthermore, Castagna [16] and Brocher [17] shows the best performance where (RMSE= 0.131, R^2= 0.804) and (RMSE= 0.144, R^2= 0.806), respectively. Picket [5] equation demonstrate to be the lowest performance (RMSE= 0.250, R^2= 0.804)

All of the empirical equations use compressional velocity to estimate shear wave velocity as it is closely related. Although empirical equations were able to follow the measured shear wave velocity trend closely, there are many intervals that the empirical equation underestimates or overestimates the measured shear wave velocity, as shown in Figure 14. Furthermore, the equation by Pickett [5] mostly overestimates the actual shear wave velocity. In contrast, Brocher's equation [17] was able to closely follow the actual shear wave velocity in most depth intervals. The empirical equations used in the study are solely dependent on the compressional velocity and are based on a specific field study. Therefore, adjustment of the constant value is necessary to obtain better performance.

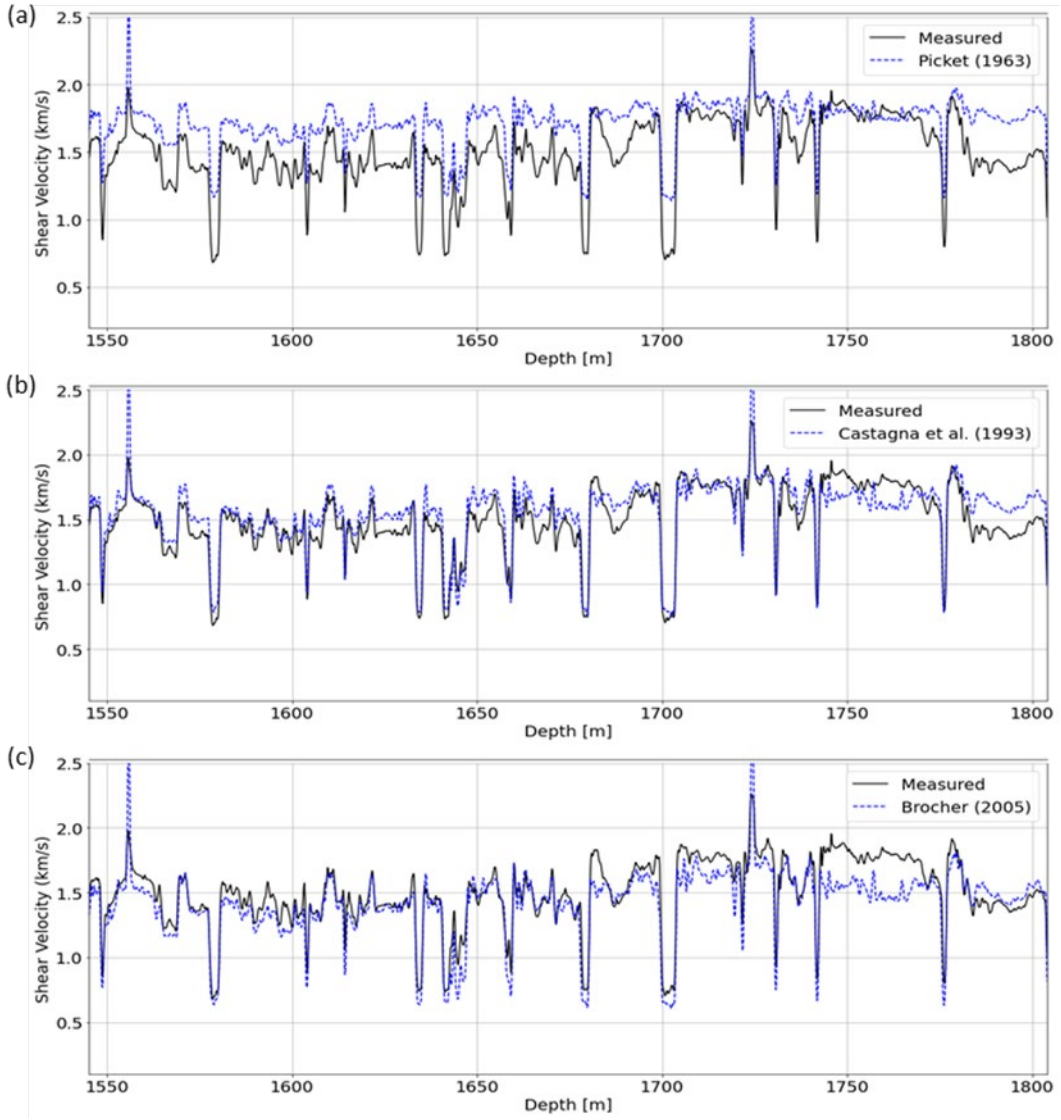

Figure 14: The comparison of estimated and measured shear wave velocity using empirical calculation methods. (a) Measured Shear velocity vs. Picket [5], (b) Measured Shear velocity vs. Castagna *et al.* [16], and (c) Measured Shear velocity vs. Brocher [17]

Overall, this study shows the importance of optimizing the machine learning algorithm's hyperparameter as it significantly affects the prediction accuracy. Implementing a hyperparameter optimization algorithm is necessary because it can increase the machine learning model's accuracy and eases the researchers from manually tuning the hyperparameter. An intelligent optimization algorithm such as the TPE-BO algorithm is recommended because it has proven to obtain the hyperparameter's best configurations.

5. Conclusion

The present study signifies the importance of utilizing the appropriate implementation of the hyperparameter optimization algorithm. Many previous authors neglected the importance of proper hyperparameter optimization and preferred to manually fine-tune the hyperparameter configuration in predicting shear wave velocity, which can be inefficient, time-consuming, and expensive to evaluate. The result revealed that ML methods that utilized the optimized configuration of hyperparameters are performing better than the empirical equation approach.

Furthermore, the comparisons between different optimization algorithms show the TPE-BO algorithm's superiority over GS and RS algorithm. The TPE-BO algorithm significantly improved the accuracy of all presented machine learning models. The MLPNN algorithm, with the implementation of the TPE-BO algorithm, demonstrate to be the best prediction model. Furthermore, the predictive performance and the overall order of prediction are followed by the XGBoost and RF algorithm optimized by the TPE-BO algorithm. Comparing the XGBoost and RF model optimized by the TPE-BO algorithm, the XGBoost model performed slightly better than the RF model. At some intervals, the RF model is showing more overfitting than the XGBoost model. When computing power is limited, the XGBoost algorithm with the implementation of the TPE-BO algorithm can be used as an alternative approach as it is more efficient than the MLPNN algorithm.

Acknowledgments

The authors are thankful to PETRONAS Malaysia for providing the data for this study. We would like to express our appreciation to the Centre of Seismic Imaging (CSI) and Universiti Teknologi PETRONAS. Massive gratitude to UTP fundamental research grant with cost center 015LCO-224 for granting this research.

Conflicts of Interest: *The authors declare no conflict of interest.*

Reference

[1] Buland A, and Omre H. Bayesian linearized AVO inversion. Geophysics, 2003; 68(1): 185–198.
[2] BH Russell, K Hedlin, FJ Hilterman, and LR. Lines Fluid-property discrimination with AVO: A Biot-Gassmann perspective. Geophysics, 2003; 68(1): 29.
[3] GM. Hamada Reservoir fluids identification using Vp/Vs ratio? Oil Gas Sci. Technol., 2004; 59(6): 649–654.
[4] GP Eberli, GT Baechle, FS Anselmetti, and ML. Incze Factors controlling elastic properties in carbonate sediments and rocks. Lead. Edge, 2003; 22(7): 654–660.
[5] GR. Pickett Acoustic Character Logs and Their Applications in Formation Evaluation. J. Pet. Technol., 1963; 15 (06): 659–667. doi:10.2118/452-pa.
[6] Hermana M, Ghosh DP, Sum CW, and Salim AMA. Implementation of New Seismic Attributes to Improve Reservoir Properties Prediction Using Probability Neural Network. no. January, 2016, doi: 10.2523/iptc-18698-ms.
[7] RL Eastwood and JP. Castagna Basis for interpretation of Vp/Vs ratios in complex lithologies. Paper presented at the SPWLA 24th Annual Logging Symposium, Calgary, Alberta, June 1983. Paper Number: SPWLA-1983-G.
[8] Miller SLM, and Stewart RR. The relationship between elastic-wave velocities and density in sedimentary rocks: A proposal. Crewes Res. Rep., 1991; 260–273.
[9] Zhang J. Pore pressure prediction from well logs: Methods, modifications, and new approaches. Earth-Science Rev., 2011; 108(1): 50–63
[10] Eaton BA. The equation for geopressure prediction from well logs. Paper presented at the Fall Meeting of the Society of Petroleum Engineers of AIME, Dallas, Texas, September 1975. Paper Number: SPE-5544-MS, https://doi.org/10.2118/5544-MS1975.
[11] Saleh S, Williams K, and Rizvi A. Predicting subsalt pore pressure with Vp/Vs Paper presented at the Offshore Technology Conference, Houston, Texas, USA, May 2013, Paper Number: OTC-24157-MS, https://doi.org/10.4043/24157-MS.
[12] Kumar J. The effect of poisson's ratio on rock properties. Proc. - SPE Annu. Tech. Conf. Exhib., 1976, doi: 10.2523/6094-ms.
[13] Zoback MD. Reservoir Geomechanics. Cambridge University Press, 2007.

[14] Asoodeh M, and Bagheripour P. ACE stimulated neural network for shear wave velocity determination from well logs. J. Appl. Geophys., 2014; 107: 102–107.
[15] Mehrgini B, Izadi H, and Memarian H. Shear wave velocity prediction using Elman artificial neural network. Carbonates and Evaporites. Carbonates and Evaporites, 2019; 34: 1281–1291.
[16] Greenberg ML, and Castagna JP. Shear-wave velocity estimation in porous rocks: theoretical formulation, preliminary verification and applications. Geophys. Prospect., 1992; 40(2): 195–209.
[17] Brocher TM. Empirical relations between elastic wavespeeds and density in the Earth's crust. Bull. Seismol. Soc. Am., 2005; 95(6,): 2081–2092.
[18] Oloruntobi O, and Butt S. The shear-wave velocity prediction for sedimentary rocks. J. Nat. Gas Sci. Eng., 2020; 76: 103084, 2020.
[19] Yang Y, Yin X, Gao G, Gui Z, and Zhao B. Shear-wave velocity estimation for calciferous sandy shale formation. Journal of Geophysics and Engineering, 2019; 16(1): 105–115, 2019.
[20] Korjani M, Popa A, Grijalva E, Cassidy S, and Ershaghi I. A New Approach to Reservoir Characterization Using Deep Learning Neural Networks. SPE Western Regional Meeting. Society of Petroleum Engineers, Anchorage, Alaska, USA, p. 15, 2016, doi: 10.2118/180359-MS.
[21] Geng Z, Chen G, Han Y, and Wei Q. Oil Reservoir Classification Based on Convolutional Neural Network. 2018. Chinese Automation Congress (CAC), DOI: 10.1109/CAC.2018.8623320.
[22] Duan YX, Li GT, and Sun QF. Research on convolutional neural network for reservoir parameter prediction. Tongxin Xuebao/Journal Commun., 2016; 37(Z1): 1-9.
[23] Li Y, Li G, and Zheng Y. Reservoir parameter prediction using optimized seismic attributes based on gamma test. ICSCA '19: Proceedings of the 2019 8th International Conference on Software and Computer Applications, February 2019; 403–407. doi: 10.1145/3316615.3316652.
[24] Wu Z, Mo X, Zhou H, Liu L, and Li J. Classification of Reservoir Fracture Development Level by Convolution Neural Network Algorithm. in 2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2018; 243–250, doi: 10.1109/FSKD.2018.8687232.
[25] Zhai W, Li J, Xi Y, Liu G, Yang H, Jiang H, Zhou Y. The fundamental work to the pre-frac evaluation of deep shale reservoir: Evaluating reservoir heterogeneity based on principal components analysis and artificial neural network. 2019, Paper presented at the International Petroleum Technology Conference, Beijing, China, March 2019, Paper Number: IPTC-19299-MS. doi: 10.2523/iptc-19299-ms.
[26] TF Zhang, P Tilke, E Dupont, LC Zhu, L Liang, and W. Bailey Generating geologically realistic 3D reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. Pet. Sci., 2019; 16: 541–549. doi: 10.1007/s12182-019-0328-4.
[27] Valentín MB, Bom CR, Coelho JM, Correia MD, de Albuquerque MP, de Albuquerque MPFaria , EL. A deep residual convolutional neural network for automatic lithological facies identification in Brazilian pre-salt oilfield wellbore image logs. J. Pet. Sci. Eng., 2019, 179: 474-503.
[28] He J, La Croix AD, Wang J, Ding W, and Underschultz JR. Using neural networks and the Markov Chain approach for facies analysis and prediction from well logs in the Precipice Sandstone and Evergreen Formation, Surat Basin, Australia. Mar. Pet. Geol., 2019, 101:410-427.
[29] Fu Ch, Lin NT, Zhang D, Wen B, Wei Qq, Zhang K. Predicting distribution of hydrocarbon reservoirs with seismic data based on learning of the small-sample convolution neural network. Acta Geophys. Sin., 2018; 61: 4110–4125.
[30] Ani M, Oluyemi G, Petrovski A, Sina RG. Reservoir uncertainty analysis: The trends from probability to algorithms and machine learning. SPE Intelligent Energy International Conference and Exhibition 2016 - Aberdeen, United Kingdom, doi.org/10.2118/181049-MS.
[31] Zhang P, Sun JM, Jiang Y, and Gao JS. Deep Learning Method for Lithology Identification from Borehole Images. Conference Proceedings, 79th EAGE Conference and Exhibition 2017, June 2017, 2017:1 - 5
[32] Zhu L, Li H, Yang Z, Li C, and Ao Y. Intelligent logging lithological interpretation with convolution neural networks. Petrophysics, 2018; (06): 799–810.
[33] Peyret A-P, Ambía J, Torres-Verdín C, and Strobel J. Automatic Interpretation of Well Logs with Lithology-Specific Deep-Learning Methods. SPWLA 60th Annual Logging Symposium. Society of Petrophysicists and Well-Log Analysts, The Woodlands, Texas, USA, p. 20, 2019.

[34] Rezaee MR, Ilkhchi AK, and Barabadi A. Prediction of shear wave velocity from petrophysical data utilizing intelligent systems : An example from a sandstone reservoir of Carnarvon Basin, Australia.2007; 55: 201–212, 2007.
[35] Nourafkan A, and Kadkhodaie-Ilkhchi A. Shear wave velocity estimation from conventional well log data by using a hybrid ant colony–fuzzy inference system: A case study from Cheshmeh–Khosh oilfield. J. Pet. Sci. Eng., 2015; 127: 459–468.
[36] Mehrgini B, Izadi H, and Memarian H. Shear wave velocity prediction using Elman artificial neural network. Carbonates and Evaporites, 2019; 34(4):1281–1291.
[37] Hutter F, Hoos HH, and Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. in International conference on learning and intelligent optimization, 2011, pp. 507–523.
[38] Snoek J, Larochelle H, and Adams RP. Practical bayesian optimization of machine learning algorithms. in Advances in neural information processing systems, 2012: 2951–2959.
[39] Snoek J, Rippel O, Swersky K, Kiros R, Satish N, Sundaram N, Patwary M, Prabhat M, Adam R. Scalable Bayesian Optimization Using Deep Neural Networks. in Proceedings of the 32nd International Conference on Machine Learning, 2015; 37: 2171–2180.
[40] Bergstra J, Bardenet R, Bengio Y, and Kégl B. Algorithms for Hyper-Parameter Optimization. in Proceedings of the 24th International Conference on Neural Information Processing Systems, 2011: 2546–2554.
[41] Jones DR. A taxonomy of global optimization methods based on response surfaces. J. Glob. Optim.,2001; 21(4): 345–383.
[42] Madon M, Yang J-S, Abolins P, Hassan RA, Yakzan AM, and Zainal SB. Petroleum systems of the Northern Malay Basin. Bull. Geol. Soc. Malaysia, 2006; 49:125-134
[43] Madon M, Abolins P, and Ahmad MB. Petroleum Systems of the North Malay Basin. p. 665, 1999.
[44] Anemangely M, Ramezanzadeh A., Amiri H, and Hoseinpour S. Machine learning technique for the prediction of shear wave velocity using petrophysical logs. J. Pet. Sci. Eng., 2019; 174: 306–327.
[45] Yasin Q, Du Q, Kashif M, and Ismail A. State-of-the-Art in shear velocity determination from Geophysical well log data: Verifiable, accurate and touch-stone of all empirical models. in International Geophysical Conference, Qingdao, China, 17-20 April 2017, 2017, pp. 86–90.
[46] Géron A, and Safari OMC. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd Edition. O'Reilly Media, Incorporated, 2019.
[47] Osborne MA. Bayesian Gaussian processes for sequential prediction, optimisation and quadrature. Oxford University, UK, 2010.
[48] Xia Y, Liu C, Li YY., and Liu N. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. Expert Syst. Appl., 2017; 78: 225–241.
[49] Chen T, and Guestrin, C. Xgboost: A scalable tree boosting system. in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
[50] Guo J, Yang L, Bie R, Yu J, Gao Y, Shen Y, Kos A. An XGBoost-based physical fitness evaluation model using advanced feature selection and Bayesian hyper-parameter optimization for wearable running monitoring. Comput. Networks, 2019; 151: 166–180.
[51] Breiman L. Random forests. Mach. Learn., 2001; 45(1): 5–32.
[52] Chen W, Li Y, Tsangaratos P, Shahabi H, Ilia I, Xue W, Bian H. Groundwater spring potential mapping using artificial intelligence approach based on kernel logistic regression, random forest, and alternating decision tree models. Appl. Sci., 2020; 10(2): 425, 2020.
[53] D Li, F Huang, L.Yan, Z Cao, J Chen, and Z. Ye Landslide Susceptibility Prediction Using Particle-Swarm-Optimized Multilayer Perceptron: Comparisons with Multilayer-Perceptron-Only, BP Neural Network, and Information Value Models. Appl. Sci., 2019; 9(18): 3664, 2019.
[54] EPIC (Esso-PETRONAS Integrated Collaborative Study), "EPIC diagram," Reg. study Malay Basinfinal portfolios, vol. Unpublishes, no. Esso Production Malaysia Inc, 1994.

To whom correspondence should be addressed: Zaky Ahmad Riyadi, Department of Geoscience, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Malaysia, E-mail: zaky_20473@utp.edu.my