

# Synthetic shear sonic log generation utilizing hybrid machine learning techniques



Jongkook Kim

Korea National Oil Corporation, 305 Jongga-ro, Ulsan, 44538, South Korea

## ARTICLE INFO

**Keywords:**

Synthetic log  
Data clustering  
Particle swarm optimization (PSO)  
Support vector machine (SVM)  
Deep neural network (DNN)  
Long short-term memory (LSTM)

## ABSTRACT

Compressional and shear sonic logs (DTC and DTS, respectively) are one of the effective means for determining petrophysical/geomechanical properties. However, the DTS log has limited availability mainly due to high acquisition costs. This study introduces a hybrid machine learning approach to generating synthetic DTS logs. Five wireline logs such as gamma ray (GR), density (RHOB), neutron porosity (NPHI), deep resistivity (Rt), and DTS logs are used as input data for three supervised-machine learning models including support vector machine for regression (SVR), deep neural network (DNN), and long short-term memory (LSTM). The hybrid machine learning model utilizes two additional techniques. First, as an unsupervised-learning approach, data clustering is integrated with general machine learning models for the purpose of improving model accuracy. All the machine learning models using the data-clustered approach show higher accuracies in predicting target (DTS) values, compared to non-clustered models. Second, particle swarm optimization (PSO) is combined with the models to determine optimal hyperparameters. The PSO algorithm proves time-effective, automated advantages as it gets feedback from previous computations so that is able to narrow down candidates for optimal hyperparameters. Compared to previous studies focusing on the performance comparison among machine learning algorithms, this study introduces an advanced approach to further improve the performance by integrating the unsupervised learning technique and PSO optimization with the general models. Based on this study result, we recommend the hybrid machine learning approach for improving the reliability and efficiency of synthetic log generation.

## 1. Introduction

Understanding the petrophysical and mechanical properties of the rock is crucial for successful oil and gas upstream projects ranging from exploration to production phase. Operators employ various exploration approaches such as coring, wireline logging, and seismic survey in order to figure out rock properties and their spatial distributions. Each approach has differences in spatial coverage, resolution, cost, and processing time. Coring provides direct measurements of rock properties by acquiring rock samples, but it requires a series of laboratory tests and additional time to obtain rock properties. Even though logging is one of the indirect approaches to obtaining rock properties, both the time and cost of dealing with wireline log data are more efficient compared to the coring process. In addition, core data only provide discrete information from a limited number of rock samples so which ultimately requires wireline log data to get continuous rock properties along the wellbore. For these reasons, operators normally calibrate petrophysical/mechanical logs derived from wireline log data to core data. Regarding the seismic survey, it has a larger areal coverage compared to core and

wireline log approaches, but it also requires wireline log data for rock property estimations in the inversion process. In general, standard wireline log tools measure natural gamma radiation, formation density, resistivities in different penetrations, and neutron porosity. These are employed in general formation evaluations providing petrophysical and lithological information (Alexeyev et al., 2017; Wang et al., 2019a; He et al., 2019). And as an additional approach, acoustic logging (i.e., detecting acoustic waves such as compressional- and shear sonic logs) is carried out to identify advanced petrophysical- and mechanical characteristics. The compressional- and shear sonic logs are essential in deriving not only mechanical properties including Young's Modulus, Poisson's Ratio, Lambda-rho, and Mu-rho (Ameen et al., 2009; Rasouli et al., 2011) but its application to quantitative seismic analysis (Bukar et al., 2019; He et al., 2019; Olaiyiola and Sanuade, 2021). In recent years, advanced applications utilizing sonic logs have reached to evaluating gas-hydrate systems, which are expected to provide additional geological implications in the exploration and exploitation followed. (Saumya et al., 2019; You et al., 2021). While the monopole logging tool has a limitation in measuring shear wave data from slow formations, the

E-mail address: [jk.kim2008@knoc.co.kr](mailto:jk.kim2008@knoc.co.kr).

<https://doi.org/10.1016/j.aiig.2022.09.001>

Received 10 July 2022; Received in revised form 17 September 2022; Accepted 19 September 2022

Available online 28 September 2022

2666-5441/© 2022 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

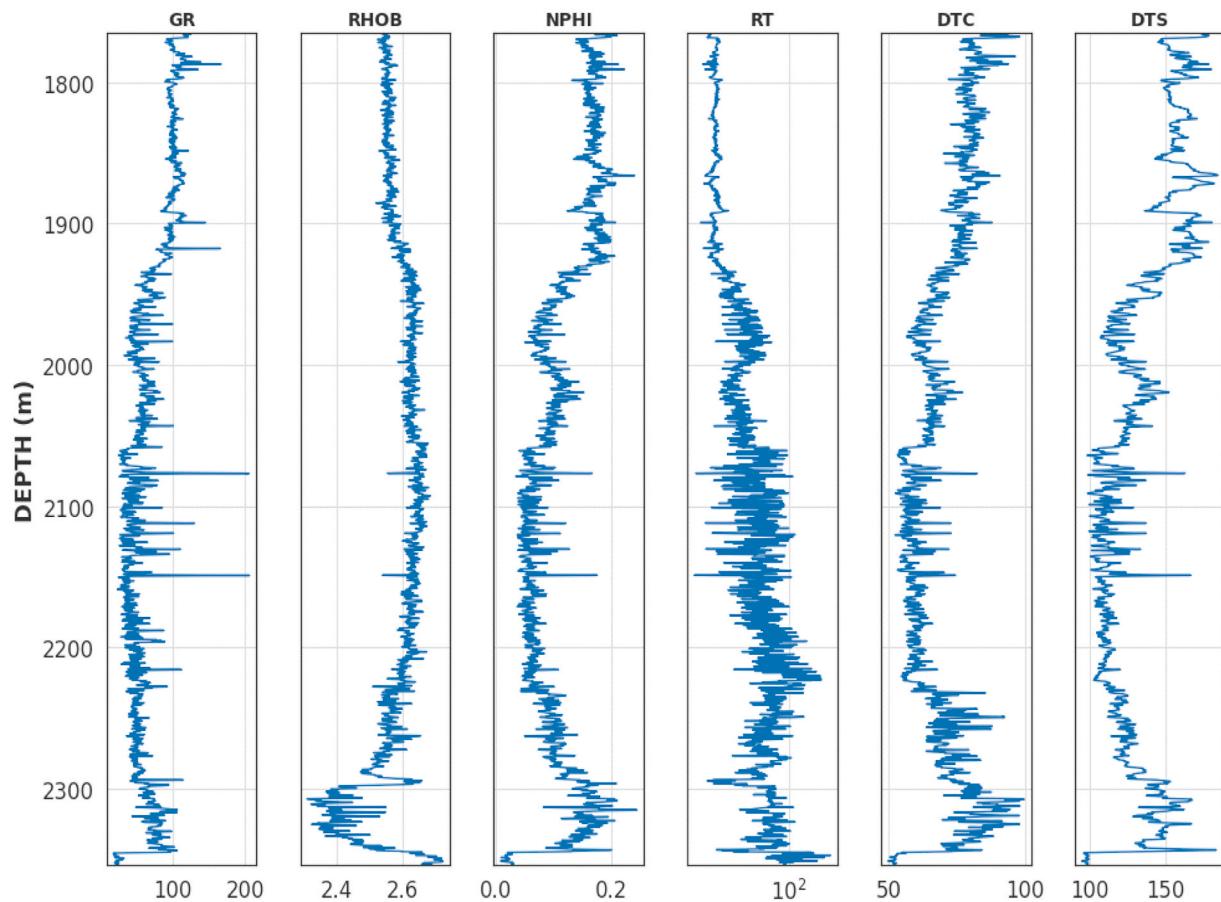
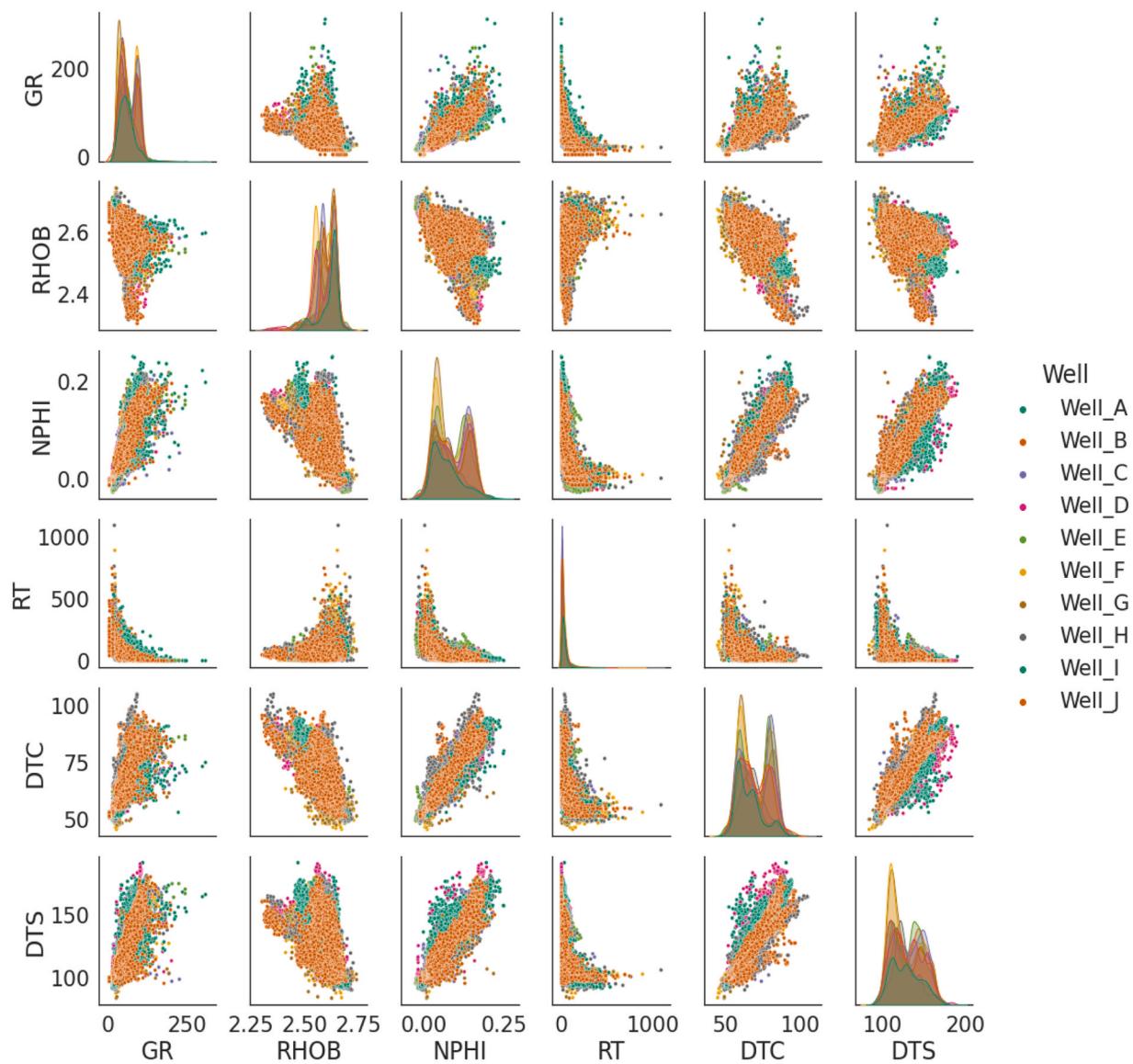


Fig. 1. Wireline logs example (Well-A).

dipole sonic-logging tool is able to detect both compressional and shear wave data. However, the dipole sonic tool is more expensive compared to the monopole sonic tool so the availability of shear sonic is highly limited compared to compressional sonic log (Close et al., 2009; Anemangely et al., 2017). As one of the alternative ways, there have been attempts to generate synthetic shear sonic logs using empirical correlations between other wireline logs and shear sonic logs available from analogy data. Especially, based on the highest correlation between compressional sonic log and shear sonic logs, researchers have provided empirical equations at various lithological settings (Greenberg and Castagna, 1992; Castagna et al., 1985; Iverson and Walker, 1988; Brocher, 2005; Lee, 2006). However, this approach has two major limitations. First, the empirical equations have intrinsic uncertainties concerning broad applications because the correlation between compressional and shear sonic logs ( $V_p/V_s$  ratio in other words) is affected by multiple geological features such as mineralogy, lithology, porosity, fluid type, saturation, compaction, and overpressure (Zimmer et al., 2002; Mavko et al., 2009; Brantut and David, 2019; Chaikine and Ian, 2020). Second, since it commonly uses a simple relationship between compressional and shear sonic log, these approaches are not able to identify variations in the  $V_p/V_s$  ratio which is one of the useful techniques to identify fluid type in pore space. Recent studies have proved that using multiple logs is better at reducing error in synthetic shear sonic log generation compared to the single log-approach (Anemangely et al., 2017; Akhundi et al., 2014; Ali et al., 2021; You et al., 2021). Oloruntobi and Butt (2020) introduced new regression equations utilizing the empirical relationship of  $V_s$ ,  $V_p$ , and density logs, which are applicable to a broader range of lithological settings. In addition to that, the introduction of advanced machine learning techniques and deep learning algorithms led to improvements in synthetic log generations. Rezaee et al. (2007) tested three machine learning algorithms such as

fuzzy logic, neuro-fuzzy, and back-propagation Artificial Neural Network (ANN) using five conventional wireline logs. All these approaches showed similarly high accuracies in the shear sonic log prediction. Maleki et al. (2014) and Bagheripour et al. (2015) compared empirical equations to ANN and Support Vector Regression (SVR). SVR approaches provided better performance in their studies. He et al. (2019) tested 6 different types of machine learning models such as Ordinary Least Squares (OLS), Partial Least Squares (PLS), Elastic Net (EN), Least Absolute Shrinkage and Selection Operator (LASSO), Multivariate Adaptive Regression Splines (MARS), and ANN. In this study, the ANN model outperformed the other five models. Olayiwola and Sanuade (2021) utilized ANN, adaptive neuro-fuzzy inference system (ANFIS), and least square support vector machine (LSSVM) to generate both compressional and shear sonic logs, which showed that LSSVM is the most accurate model. As a further advanced approach, Anemangely et al. (2019) made an attempt to utilize hybrid machine learning algorithms combining general LSSVM model and optimization algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Cuckoo Optimization. In recent years, Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) are being emerged with the advantage of high performance in sequential data. Zhang et al. (2018) proposed an LSTM model for synthetic log generation backed by higher accuracy than traditional ANN approaches. Chaikine and Ian (2020) introduced a hybrid convolutional-Recurrent Neural Network (c-RNN) that showed a better performance compared to the ANN model. Pham and Naeini, 2019 and Jeong et al. (2021) tested bidirectional LSTM (bi-LSTM) to take bidirectional anisotropy of sequential data into account. Convolutional Neural Network (CNN) is another recently proven machine learning algorithm providing a high accuracy in predicting shear wave velocity. Rajabi et al. (2022) compared machine learning model performances of the CNN-based model and hybrid



**Fig. 2.** Data distribution (Wireline logs of ten wells: Well-A to J). GR: Gamma ray, RHOB: Density, NPHI: Neutron porosity, Rt: Deep resistivity, DTC: Compressional sonic (slowness), DTS shear sonic (slowness). Unit for each wireline log is listed in [Table 1](#).

models that integrate a multi-hidden layer extreme learning model (MELM) and optimization algorithms such as particle swarm optimizer and genetic algorithm. These data-driven approaches have made great achievements in generating practical as well as reliable synthetic shear sonic logs. However, previous studies were more focused on comparing overall performances among prospective machine learning models to identify the superiority. For this reason, it has limitations in terms of maximizing the performance of the best model. First, in most studies, a single set of hyperparameters was provided, which were determined manually through repetitive runs. This implies that overall model performance could vary depending on data analysts' manual approaches. And, considering that possible combinations of these hyperparameters are too many to test all the cases, it is subtle to decide how many times the model has been run with different combinations to reach an optimal result. Second, most studies set machine learning models without considering the variations in geological characteristics (i.e., lithology, lithofacies, mineralogy) that affect the relationship between the shear sonic log and other logs. This is mainly because (1) the geological characterization requires additional time and cost for interpretation, (2) these approaches mainly depend on manual analysis that might cause unexpected error or complexity to the machine learning approaches

followed. In this study, we introduce a hybrid machine learning model generating shear sonic log for the purpose of enhancing model performances compared to the general approach. As an unsupervised geological characterization as well as an alternative way of geological characterization, data clustering was applied to test if it could help enhance the model performance. And PSO algorithm is combined with three representative machine learning models such as SVR, DNN (Deep Neural Network), and LSTM in order to find an optimal set of hyperparameters. These two approaches are expected to improve the performance of general machine models further as well as reduce manual intervention and time spending.

## 2. Material and method

### 2.1. Data availability and preprocessing

This study was carried out using six wireline logs such as Gamma ray (GR), Density (RHOB), Neutron porosity (NPHI), Deep resistivity (Rt), compressional sonic (DTC), and shear sonic (DTS) log available from ten vertical wells drilled in onshore Texas, US. In order to keep these wells' confidentiality, actual well names are renamed as Well-A ([Fig. 1](#)) to J for

**Table 1**  
Data statistics.

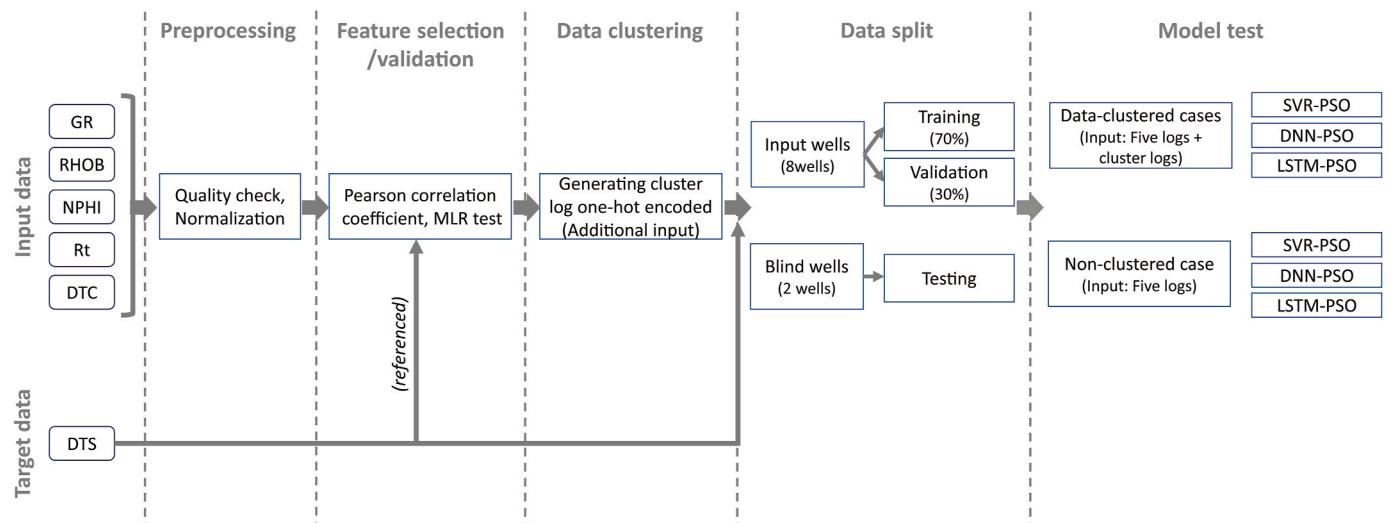
	GR	RHOB	NPHI	Rt	DTC	DTS
Unit	gAPI	g/cm <sup>3</sup>	m <sup>3</sup> /m <sup>3</sup>	ohm.m	μs/ft	μs/ft
Counts	37060	37060	37060	37060	37060	37060
Min	3.912	2.291	-0.021	1.447	40.274	73.156
Max	291.81	2.787	0.277	1092.413	99.794	202.666
Standard deviation	30.072	0.063	0.059	45.519	10.621	20.652

this study. The strata in the study area are sedimentary successions that mainly consist of clean limestone, dolomitic limestone, volcanic ash, marl, calcareous shale, and organic-rich clayey shale. The vertical resolution of the wireline log is 0.5 ft (15.24 cm) in common. Data quality

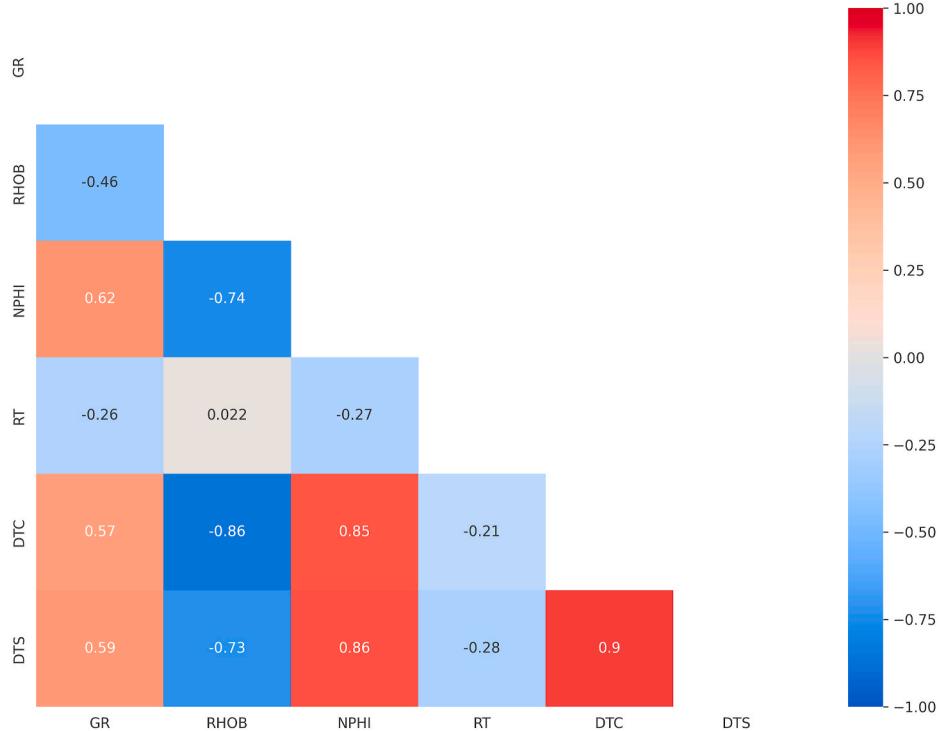
was assessed using the caliper log in order to identify bad borehole indications. From the original dataset (37,187 data points), 127 data points (0.3% of the original data) with abnormally high or low values in borehole-damaged intervals were eliminated in advance. In data distribution, overall wireline logs show multi-modal distribution (Fig. 2), which implies that the characteristics of the strata could be subdivided by petrophysical features. As the five input wireline logs (GR, RHOB, NPHI, Rt, and DTC) varies both in range and scale (Table 1), each wireline log was rescaled to fit a standard range (Min = 0 and Max = 1) by the following min-max scaling method (Eq. (1)).

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where  $x_{max}$  and  $x_{min}$  are the maximum and minimum values of a feature,



**Fig. 3.** Workflow.



**Fig. 4.** Pearson correlation coefficient matrix.

**Table 2**  
Coefficient of determinations ( $R^2$ ) in multi-linear regression.

Features	Mean Absolute Error (MAE)			$R^2$		
	Train	Validation	Test	Train	Validation	Test
DTC	5.322	5.360	4.565	0.791	0.791	0.851
DTC, NPHI	4.720	4.732	4.561	0.847	0.849	0.856
DTC, NPHI, RHOB	4.647	4.656	4.560	0.851	0.849	0.858
DTC, NPHI, RHOB, GR	4.475	4.493	4.558	0.860	0.861	0.862
DTC, NPHI, RHOB, GR, RT	4.451	4.468	4.531	0.862	0.862	0.862

respectively, and  $x_{\text{norm}}$  is the normalized value.

## 2.2. Workflow

Fig. 3 illustrates the workflow implemented in the study. As described in section 1 (introduction), this study aims to test hybrid machine learning models that generate synthetic DTS log (Target) using five wireline logs (Input data: GR, RHOB, NPHI, Rt, and DTC). After preprocessing including quality check and input data normalization, feature selection and data clustering were implemented. Details of the two procedures are described in the following sections (Section 2.3 and 2.4). Eight wells (Well-A to H) out of ten available wells were used for training and validation. Data samples (30297 samples) of the eight wells were mixed and randomly split into training (70%, 21207 samples) and validation data (30%, 9090 samples). These datasets were used for two hybrid machine learning models such as the SVR-PSO and DNN-PSO. However, for the LSTM-PSO model, the training (70%) and validation data (30%) were split as keeping the depth order without mixing dataset because the LSTM model deals with sequential data. Two wells (Well-I and J, 6763 samples) were used for testing as blind wells. In order to test if the cluster information helps improve model performance, two types of the case study were conducted. The first is to test models using

clustering data. Cluster logs generated by the data clustering process (i.e., each data point is given a cluster group number) were utilized as extra input logs in addition to the five wireline logs. The cluster log was one-hot encoded to avoid unexpected impacts by the magnitude of cluster numbers. For this reason, two cluster logs were generated in the case of defining two clusters, and three cluster logs for the three-cluster case. The second case is to test models using non-clustered data (i.e., five wireline logs without cluster logs), which represents general approaches employed in previous studies. The machine learning models were implemented using Python (version 3.7.12) with TensorFlow library (version 2.7.0) on the Google Colab platform that utilizes Intel Xeon 2.30 GHz CPU, NVIDIA Tesla T4 16 GB GPU operated under Linux platform (Linux-5.4.144+-x86\_64-with-Ubuntu-18.04-bionic).

## 2.3. Feature selection

Using optimal features is recommended since it not only increases process speed by reducing data volume but improves the accuracy of machine learning models (Anemangely et al., 2019). In this study, two approaches were employed. First, the Pearson correlation coefficient matrix (Fig. 4) was used to figure out the correlation between the individual input feature and target data. DTC showed the highest correlation (0.90) to the target (DTS). And NPHI (0.86), RHOB (-0.73), and GR (0.59) were followed with good to fair correlations. Rt was the least (-0.28). Based on the Pearson correlation, the order of input candidates was identified (i.e., DTC > NPHI > RHOB > GR > Rt). Second, a multi-linear regression (MLR) model was employed to compare prediction accuracies for different numbers of the feature. The MLR model is relatively simple and easy to code as well as quick to process compared to advanced machine learning models. For this reason, the MLR model was used as a sort of tester to determine an optimal combination of features. Five cases (from one to five input logs) were set, and features for each case were selected using the rank of the Pearson correlation coefficient. The five-feature case showed the best accuracies both for the mean absolute error (MAE) and determination of coefficient ( $R^2$ ).

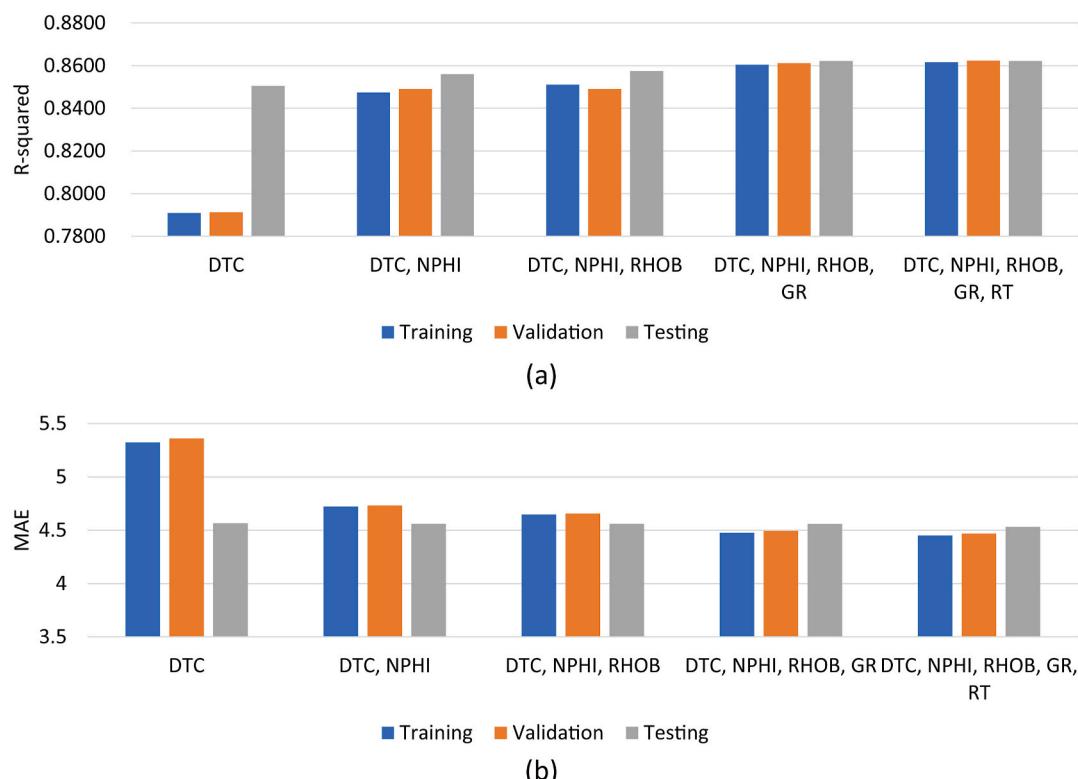
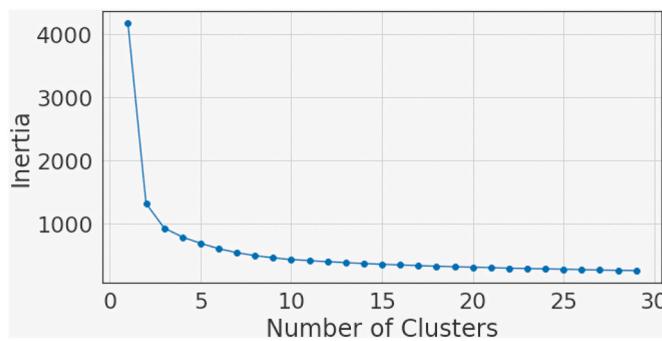


Fig. 5. MLR model test result for feature selection: (a) Mean absolute error, (b) Coefficient of determination ( $R^2$ ).



**Fig. 6.** Inertia values with increasing clusters.

(Table 2 and Fig. 5). Based on the result, these five input logs were determined to use for the data clustering and model tests followed. However, despite the fast and intuitive advantages, both the Pearson's coefficient and the multilinear regression have a limitation in figuring out non-linear relationships among multiple features. Based on the data distribution (Fig. 2) and previous research (Rajabi et al., 2022) imply that shear wave velocity tends to have complicated relationships that are not simply explained by the linear regression-based approaches. That is also one of the critical motivations to test multiple machine learning algorithms dealing with non-linear relationships among input features for the prediction of the DTS in this study.

#### 2.4. Data clustering

Clustering is the data analysis dividing a set of data into subgroups which are called clusters that data samples in the same cluster have similar characteristics to each other. K-means clustering is one of the widely-used, unsupervised clustering approaches with the advantages of simplicity and fast calculation (He et al., 2019; Kang and Choe, 2020;

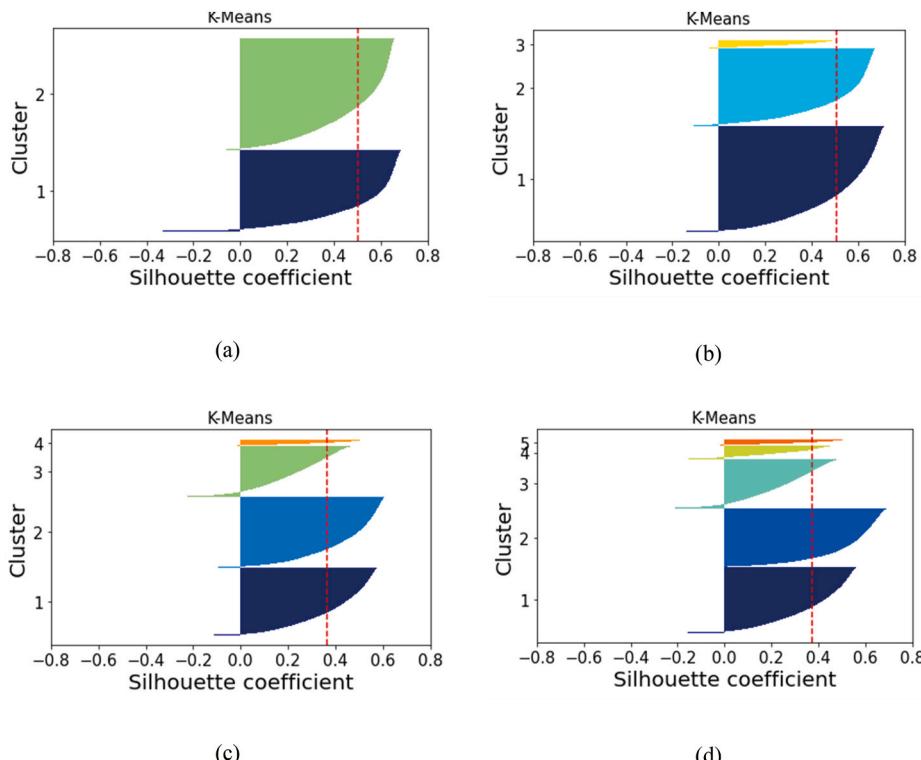
Al-Mudhafar and Bondarenko, 2015). K-means algorithm divides a dataset into clusters by minimizing the cluster inertia described as Eq. (2). It requires the number of clusters manually determined as one of the predefined criteria. As an initial state, it randomly selects points for initial clustering, which is necessarily not optimal at the beginning. Each data sample is allocated to the nearest centroid for the initial clustering. Then, the initial centroids are relocated to the center of each cluster so that updates centroids. This relocating process is iterated until it reaches predefined criteria such as a number of iterations or a value of the distortion function.

$$\text{Cluster inertia} = \sum_{k=1}^K \sum_{m=1}^M \|x_m^{(k)} - c_k\|^2 \quad (2)$$

where  $x^{(k)}$  and  $c$  are the data in  $k$ -th cluster and the centroid of the cluster, respectively.  $M$  is the number of data samples in each cluster.

In terms of finding optimal cluster numbers, two approaches such as elbow method and silhouette method were used for this study. The elbow method is to identify the number of clusters where the inertia shows a significant decrease as increasing number of clusters. In general, as the number of centroid increases, the inertial decreases since data samples are divided into smaller groups and ultimately samples in each cluster get closer to each other. However, too many clusters add complexity that could lower the model performance at the same time. For this reason, it is normally recommended to select optimal cluster numbers at the elbow point where decreasing inertia is no longer worth the additional cost (Ketchen and Shook, 1996). In this study, the elbow method was employed to narrow down candidates for optimal cluster models. Fig. 6 shows the trend of inertia with increasing number of clusters. The inertia significantly decreases at two to five clusters while the decreasing rate gradually lowers over higher cluster numbers. Based on the result of the elbow method, two to five cluster-cases were selected to compare in the silhouette method.

The elbow method and the silhouette method are complementary approaches to identifying optimal cluster numbers. The elbow method

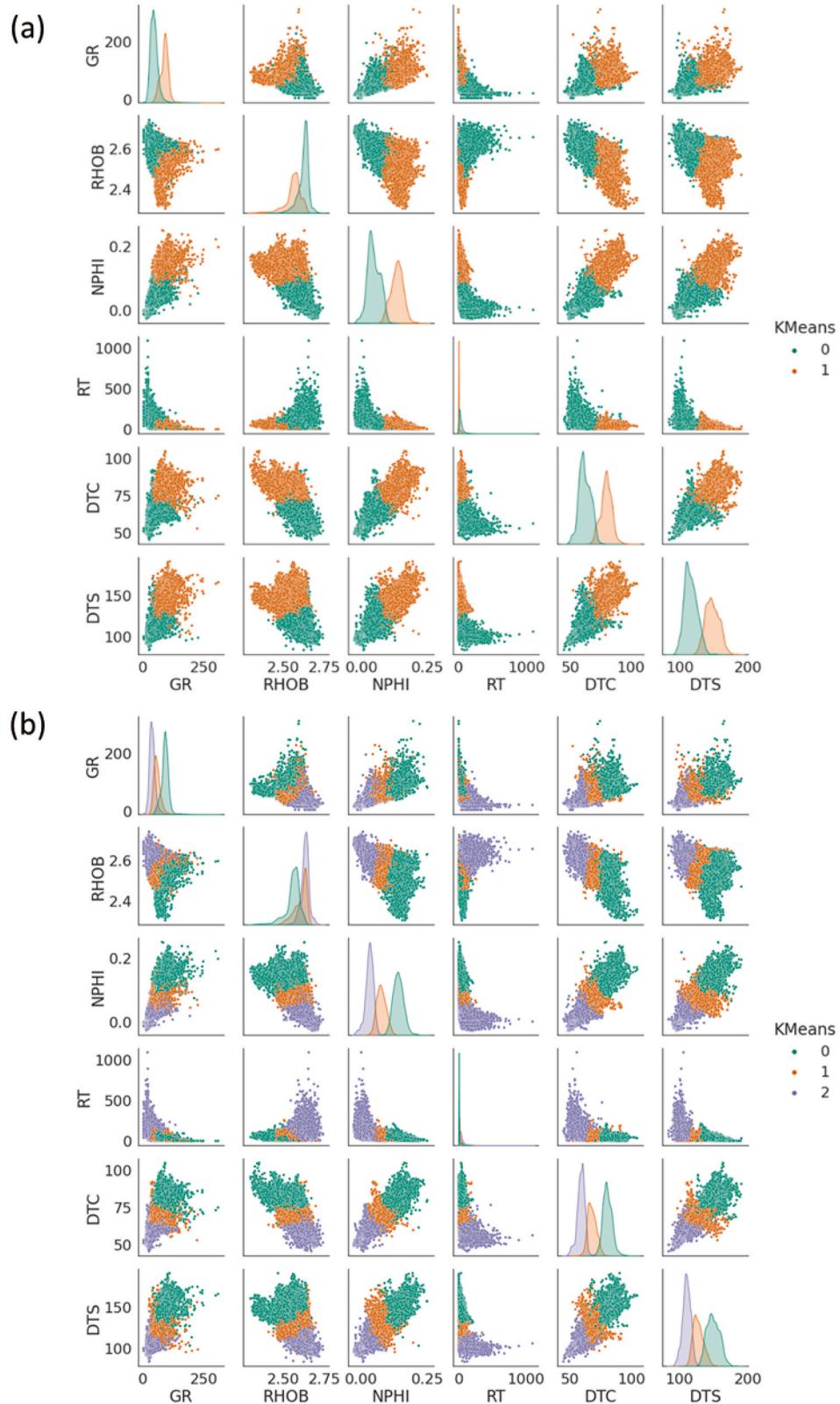


**Fig. 7.** Silhouette coefficients for various cluster-cases: (a) 2 clusters, (b) 3 clusters, (c) 4 clusters, (d) 5 clusters. The dotted line in red shows an average silhouette coefficient for each case.

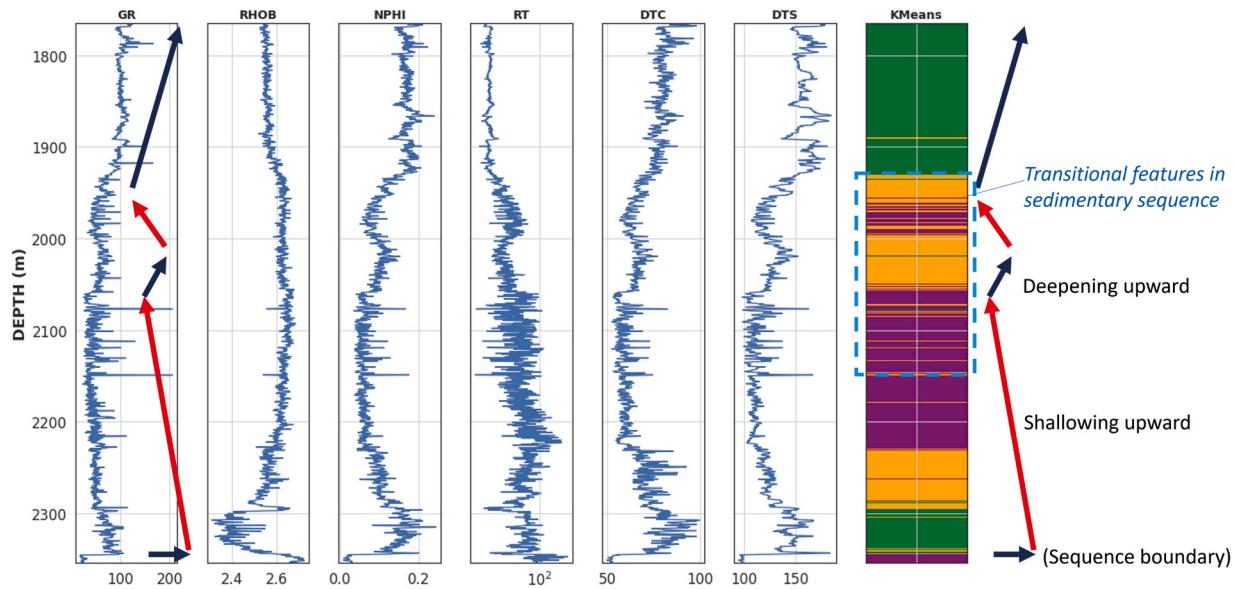
**Table 3**  
Average silhouette coefficients of various cluster cases.

Clusters	2	3	4	5
Silhouette coefficient (Average)	0.503	0.507	0.366	0.374

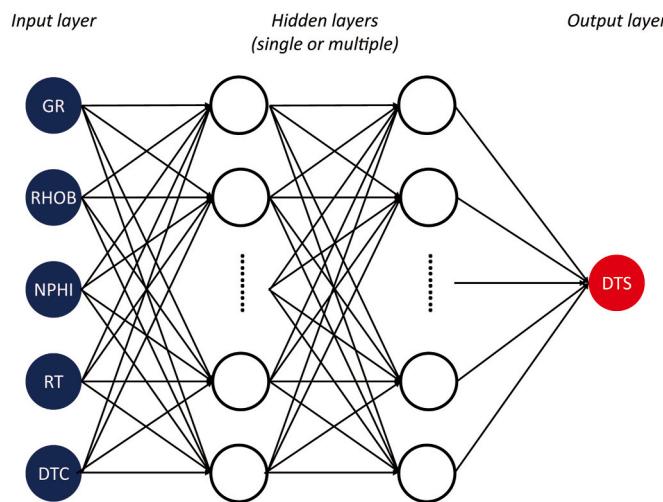
provides an absolute value of how concentrated data samples are to a centroid of each cluster, but the silhouette provides relative values of how well-grouped clusters are from other clusters (Eq. (3)). For this reason, good values in the elbow method do not necessarily ensure the same implication in the silhouette method if clusters are close or



**Fig. 8.** Clustered data distribution: (a) 2-cluster cases, (b) 3-cluster case. The total number of data samples is 37060 from ten wells.



**Fig. 9.** Wireline logs and a cluster log (Well-A, 3-cluster case). The vertical variation of the cluster group is equivalent to the overall sedimentary sequences as well as captures transitional features interpreted using wireline logs. It supports the idea that the clustering method could help identify geological implications despite the unsupervised approach without manual intervention.



**Fig. 10.** Schematic architecture of the ANN or DNN.

overlapped each other.

$$\text{Silhouette coefficient (at } i\text{-th data point)} = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (3)$$

where  $a(i)$  is the mean distance of other data in the same cluster and  $b(i)$  is the minimum of the mean distances from data in other clusters. The silhouette coefficient ranges from  $-1$  (worst) to  $1$  (best), and the value closer to  $1$  indicates that the data better fit the cluster. Fig. 7 and Table 3 outline the result of the silhouette method. 2- and 3-cluster cases showed relatively well-grouped features ( $0.503$  and  $0.507$  in average coefficient, respectively) over four- and five-cluster cases ( $0.366$  and  $0.374$ ). Based on the result, 2- and 3-cluster cases were finally selected for testing the machine learning model. Since the two cluster types (2- and 3-clustering) showed similar result in the silhouette method, both cluster types are respectively tested as an additional input feature. Fig. 8 illustrates actual data distributions with the cluster information. Clusters are grouped well in the crossplots, and each cluster shows distinguishable concentrations in histogram charts. As clustering is a data-driven

approach using petrophysical measurements, it could capture geological features like a proxy of lithofacies even though it is an unsupervised approach without manual geological interpretation. Fig. 9 is an example of wireline log data and the cluster log corresponded. The vertical trend of the cluster log shows characteristics of the overall sedimentary sequence that can be manually interpreted using wireline logs. In particular, the clustering log captures the transitional features in the middle of the well section, which implies gradual changes of the depositional environment.

In the original cluster log, the 2-cluster case has  $0$  or  $1$  as a cluster code for each single data point based on the clustering result, and the 3-cluster case has one among  $0$ ,  $1$ , and  $2$ . However, the cluster code doesn't have numerical or ordinary implications because it is used for categorical (label) features as a proxy of lithofacies that the same cluster code implies similar petrophysical characteristics (Fig. 8). Based on this reason, the original cluster log was one-hot encoded to avoid unexpected damage on the learning process. As the result of one-hot encoding, code ' $0$ ' becomes ' $1-0$ ' while code ' $1$ ' becomes ' $0-1$ ' in the 2-cluster case. In the same manner, the original code  $0$ ,  $1$ , and  $2$  become ' $1-0-0$ ', ' $0-1-0$ ', and ' $0-0-1$ ', respectively. This means two input features (one-hot encoded cluster logs) are added to default input features (i.e., wireline logs such as GR, RHOB, etc.) when testing the 2-cluster case whereas three input features are added for testing the 3-cluster case.

## 2.5. Machine learning algorithms

### 2.5.1. Support vector machine

Support Vector Machine (SVM) is one of the supervised learning algorithms developed to find a solution for nonlinear regressions or classification problems. Support Vector Machine models established for regression are called Support Vector Regression (SVR). The main idea of the SVR is that complicated relationships between input and target values can be interpreted as a hyperplane in higher dimensional spaces. The SVR model maps a set of data into the higher dimensional space so that enables to find an optimal function between input and target values in a way of minimizing deviation (Lu et al., 2009; Tan et al., 2015). The mathematical equation of the SVR model can be described as a simple term in Eq. (4).

$$y = f(x) = (w \bullet \varphi(x)) + b \quad (4)$$

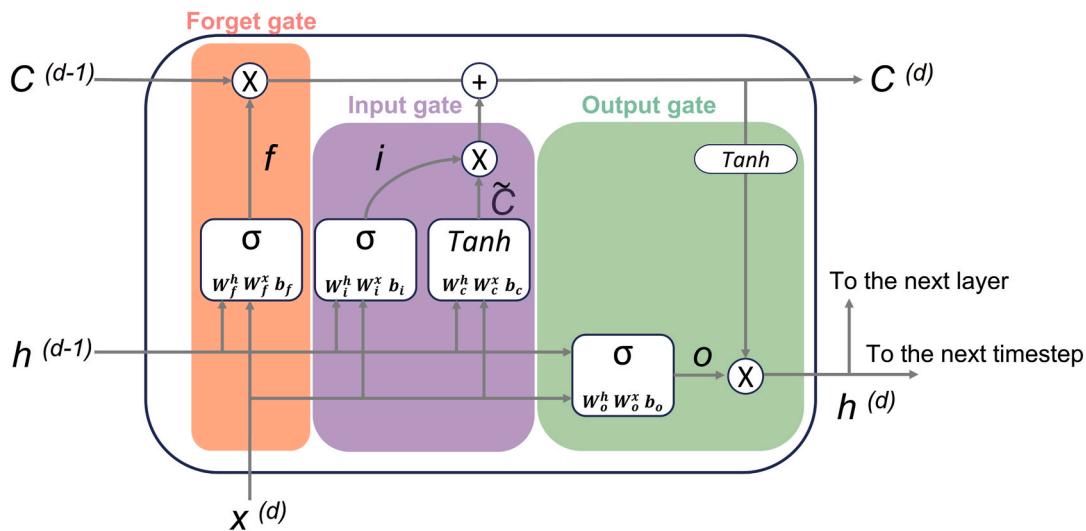


Fig. 11. Illustration of the LSTM cell structure (Modified from Raschka and Mirjalili, 2019).

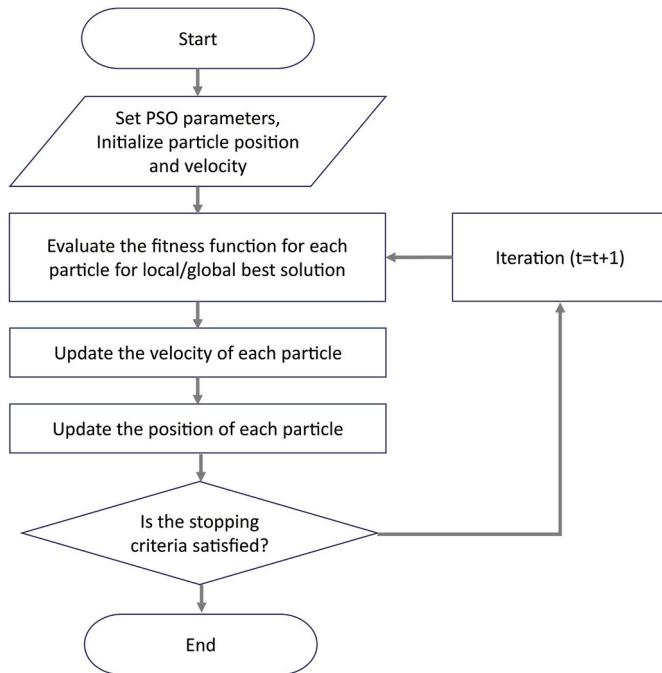


Fig. 12. Flowchart of the PSO algorithm.

where  $w$  is a weight vector,  $\varphi(x)$  is a function defining how to map in the feature space, and  $b$  is a bias term.

The SVR algorithm is designed to minimize the differences between input and target values through the optimization described as Eq. (5).

$$\text{minimize } \frac{1}{2}w^2 + \frac{1}{2}\gamma \sum_{i=1}^N d_i^2 \quad (5)$$

$$\text{subject to } y_i = w^T \varphi(x_i) + b + d_i$$

where  $\gamma$ ,  $d_i$ ,  $N$  represent regularization parameter, error at  $i$ -th data point, and the number of input data points.

Regarding the mapping approach, kernel trick is one of the mapping functions that helps set a hyperplane in the higher dimensional space. The kernel function is commonly used since it doesn't require a large computational burden even if the dimension of data increases. In

Table 4  
Hyperparameters tested by PSO optimization.

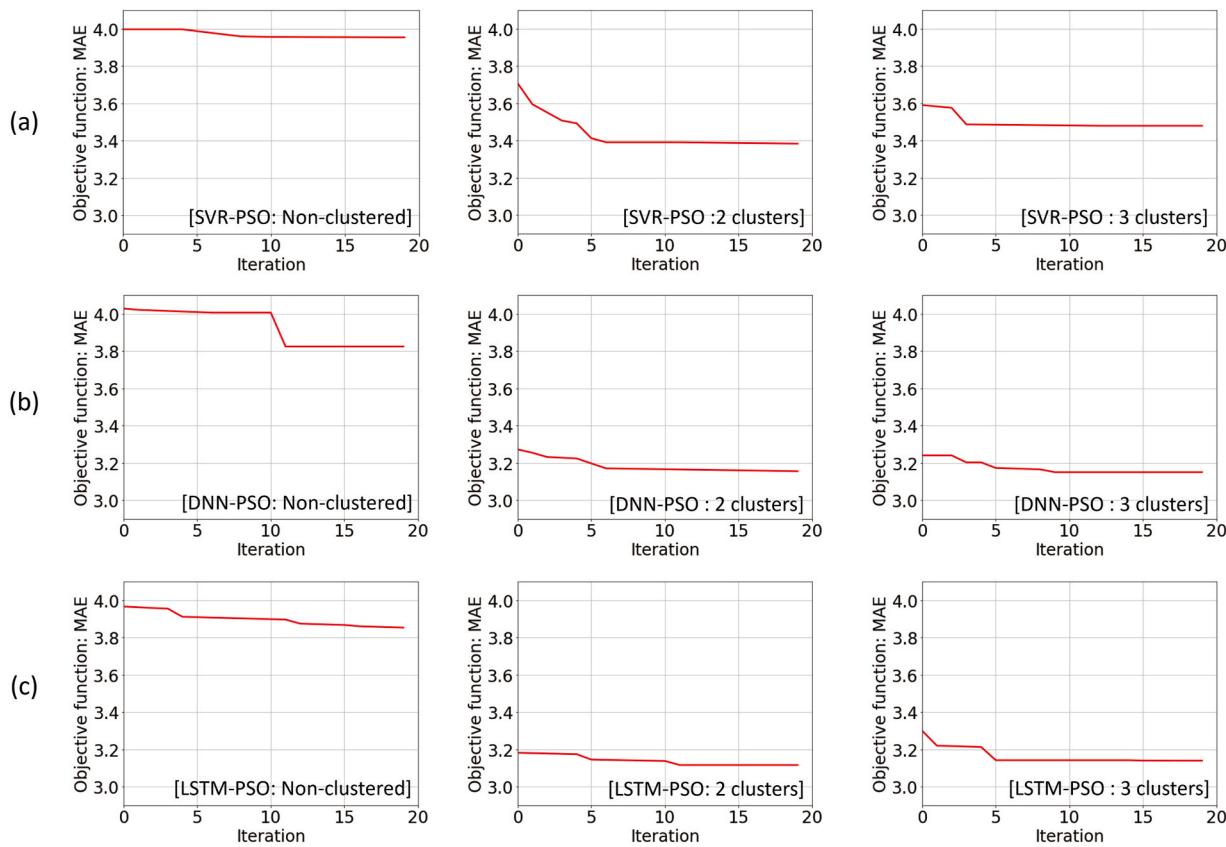
Model	Hyperparameter
SVR	<ul style="list-style-type: none"> <li>• C: 1-50</li> <li>• Gamma: 0.01–10</li> <li>• Epsilon: 0.1–10</li> <li>• Kernel: Linear, Sigmoid, Polynomial, RBF</li> <li>• Degree: 1–5 (for polynomial kernel)</li> <li>• Number of hidden layers: 1-3</li> </ul>
DNN	<ul style="list-style-type: none"> <li>• Number of output layer: 1</li> <li>• Number of neurons at 1st hidden layer: 3-50</li> <li>• Number of neurons at 2nd hidden layer (if needed): 3-50</li> <li>• Number of neurons at 3rd hidden layer (if needed): 3-10</li> <li>• Dropout: 0–0.3</li> <li>• Learning rate: <math>10^{-n}</math> (<math>n = 2-4</math>)</li> <li>• Batch size: <math>2^n</math> (<math>n = 4-9</math>)</li> <li>• Optimizer: ADAM</li> <li>• Activation function: ReLU</li> </ul>
LSTM	<ul style="list-style-type: none"> <li>• Number of LSTM layers: 1-2</li> <li>• Number of FCNN layers: 0-1</li> <li>• Number of output layer: 1</li> <li>• Number of neurons at 1st LSTM layer: 3-50</li> <li>• Number of neurons at 2nd LSTM layer (if needed): 3-50</li> <li>• Number of neurons at FCNN layer (if needed): 3-50</li> <li>• Dropout: 0–0.3</li> <li>• Timesteps: 3-20</li> <li>• Learning rate: <math>10^{-n}</math> (<math>n = 2-4</math>)</li> <li>• Batch size: <math>2^n</math> (<math>n = 4-9</math>)</li> <li>• Optimizer: ADAM</li> <li>• Activation function: Tanh (LSTM), ReLU (FCNN)</li> </ul>

Table 5  
Cluster cases and input wireline logs.

Case	Case 1 (Non-clustered)	Case 2 (2 clusters)	Case 3 (3 clusters)
Input logs	5 logs (GR, RHOB, NPHI, RT, DTC)	7 logs (GR, RHOB, NPHI, RT, Cluster 1 <sup>a</sup> , Cluster 2 <sup>a</sup> )	8 logs (GR, RHOB, NPHI, RT, Cluster 1 <sup>a</sup> , Cluster 2 <sup>a</sup> , Cluster 3 <sup>a</sup> )

<sup>a</sup> Cluster logs were generated as many as the number of clusters since it was one-hot encoded.

general, choosing optimal kernel function is regarded as having a significant impact on the performance of the SVR model (Anemangely et al., 2019). In this study, four kernel types (Linear, Sigmoid, Polynomial, and Radial basis function) were employed, and an optimal



**Fig. 13.** MAE reduction with increasing iteration of the PSO algorithm coupled to (a) SVR, (b) DNN, and (d) LSTM model. Each model was tested using three input cases such as non-clustered, 2 cluster-, and 3 cluster-case.

kernel option was selected through the optimization process.

#### 2.5.2. Artificial Neural Network/deep neural network

The Artificial Neural Network (ANN) model is one of the widely used machine learning algorithms for both linear and nonlinear regression. The basis of the ANN model is known to be inspired by the neuron system of the human brain (Russell and Norvig, 2002). The ANN model has three groups of layers, such as an input layer, single or multiple hidden layers, and an output layer (Fig. 10). Each layer has a one-dimensional data array that consists of multiple nodes which are also called neurons or perceptrons. In the early time, the ANN was designed as a feed-forward neural network that only allows the information to flow one way from the input layer to the output layer. Through the process, the ANN algorithm finds an optimum set of weights for neurons between each two layers interfacing. The mathematical equation can be defined as follows:

$$y_i = f \left( \sum_{i=1}^N w_i x_i + b \right) \quad (6)$$

where  $f$  is the activation function.  $x$  and  $y$  are values of neurons at two layers interfacing in the ANN structure.  $w$  is a set of weights, and  $b$  is a bias term. The ANN model iterates calculation until it identifies an optimum set of weights and biases that minimizes residuals between the model predictions and the true values. In particular, the development of the backpropagation (BP) algorithm has dramatically improved the performance of the ANN algorithm (Wang et al., 2019b). It returns errors backward from the latter layer to the former layer, which ultimately enables the model can adjust the initial set of weights to minimize the prediction error defined as loss function through iterated calculation. The BP algorithm is available in the open-source library Keras with TensorFlow.

#### 2.5.3. Long short-term memory

Long Short-Term Memory (LSTM) is a special kind of Recurrent Neural Network (RNN) that is specialized in sequential data. Unlike DNN which is a fully connected network system, the RNN has a loop in its structure. The loop transfers the information of the previous step to the next step so that the previous information can be utilized for the next data. This nature of RNN takes advantage of analyzing sequential data compared to the ANN model. However, the RNN is known to have a significant decline in learning ability in case the distance between the relevant information in the previous step and the present step increases (Zhang et al., 2018). LSTM was introduced to deal with this vanishing gradient problem (Hochreiter and Schmidhuber, 1997; Chaikine and Ian, 2020). A schematic of the LSTM network is illustrated in Fig. 11 the LSTM is composed of cells and three gate units inside (i.e., input, output, and forget gate unit). The model can efficiently preserve the long-range sequential information of input data using the internal network of the cell states (Hochreiter and Schmidhuber, 1997). Mathematical equations of the LSTM elements are described in Eqs. (7)–(12). Give the input data that comprises  $L$  types of  $D$ -sequence well-log values ( $[X^{(d-1)}, \dots, X^{(d-1)}, X^{(d)}]^T$ ,  $D \times R$ ), the estimated value at  $d$ -th depth ( $\widehat{y}_d$ ) is derived using Eq. (12).

$$\vec{i}_d = \varphi_i \left( \sum_{j=0}^D W_i^{x(d-j)} x^{(d-j)^T} + W_i^h h_{d-1}^- + b_i \right) \quad (7)$$

$$\vec{f}_d = \varphi_f \left( \sum_{j=0}^D W_f^{x(d-j)} x^{(d-j)^T} + W_f^h h_{d-1}^- + b_f \right) \quad (8)$$

$$\vec{o}_d = \varphi_o \left( \sum_{j=0}^D W_o^{x(d-j)} x^{(d-j)^T} + W_o^h h_{d-1}^- + b_o \right) \quad (9)$$

**Table 6**  
Hyperparameters used for final models.

Data Clustering	SVR	DNN	LSTM
No cluster	C: 28.1 Gamma: 0.86 Epsilon: 2.9 Kernel: RBF	Number of hidden layers: 3 Number of output layer: 1 Number of neurons: 41/15/6/1 Dropout: 0.06/0/0/0 Learning rate: 0.01 Batch size: 64 Optimizer: ADAM Activation function: ReLU	Number of LSTM layers: 2 Number of dense layers: 1 Number of neurons: 26/3/1 Dropout: 0/0/0 Timesteps: 8 Learning rate: 0.001 Batch size: 128 Optimizer: ADAM Activation function: Tanh, ReLU
2 clusters	C: 36.4 Gamma: 1.02 Epsilon: 3.7 Kernel: RBF	Number of hidden layers: 2 Number of output layer: 1 Number of neurons: 35/31/1 Dropout: 0.2/0.15/0 Learning rate: 0.001 Batch size: 64 Optimizer: ADAM Activation function: ReLU	Number of LSTM layers: 1 Number of dense layers: 2 Number of neurons: 33/10/1 Dropout: 0/0/0 Timesteps: 9 Learning rate: 0.001 Batch size: 128 Optimizer: ADAM Activation function: Tanh, ReLU
3 clusters	C: 39.3 Gamma: 0.96 Epsilon: 3.9 Kernel: RBF	Number of hidden layers: 3 Number of output layers: 1 Number of neurons: 50/33/4/1 Dropout: 0/0/0/0 Learning rate: 0.001 Batch size: 128 Optimizer: ADAM Activation function: ReLU	Number of LSTM layers: 1 Number of dense layers: 2 Number of neurons: 48/47/1 Dropout: 0/0.2/1 Timesteps: 4 Learning rate: 0.001 Batch size: 128 Optimizer: ADAM Activation function: Tanh, ReLU

$$\vec{c}_d = \vec{f}_d * \vec{c}_{d-1} + \vec{i}_d * \varphi_c \left( \sum_{j=0}^D W_c^{x(d-j)} x^{(d-j)^T} + W_c^h \vec{h}_{d-1} + b_c \right) \quad (10)$$

$$\vec{h}_d = O^{(d)} * \varphi_h (\vec{c}_d) \quad (11)$$

$$\widehat{y}_d = \varphi_x (w \vec{h}_d + b) \quad (12)$$

where  $\varphi$  denotes activation functions such as sigmoid, hyperbolic tangent and  $b$  is bias terms.  $\vec{i}_d$ ,  $\vec{f}_d$ ,  $\vec{o}_d$  are the input, forget, and output gates, respectively.  $\vec{c}_d$  is the cell state, and  $\vec{h}_d$  is the hidden state.  $W^x$  denotes the weight matrix between the input and LSTM units, and  $W^h$  denotes the weight matrix between the hidden states. The symbol  $*$  denotes elementwise multiplication.  $w$  is the weight between the hidden state and output.  $\widehat{y}_d$  is the DTS (i.e., predicted target value) at d-th depth.

## 2.6. Model optimization

### 2.6.1. Hyperparameter tuning: particle swarm optimization (PSO)

The PSO is one of the optimization algorithms employed to find an optimal solution through iterative computation using a variety of variables (Bonyadi and Michalewicz, 2017). In the PSO algorithm, a population of particles (i.e., a swarm) of candidate solutions (i.e., swarms) is designed to seek the best solution in the search space while moving positions. The initial position and velocity of the swarms are randomly selected within the upper and lower bound. Each position of the particles is set as their best-known position, and the whole swarm's best-known position (the best global position) is determined by comparing the cost functions of the particles. Then, the new velocity for the particle is calculated based on the previous velocity of the particle and the distance to the best global position (Eq. (13)). Based on the new velocity and previous position of the particle, the new location of the particle is determined (Eq. (14)). The PSO algorithm iterates this procedure as many as the pre-set number and updates the best global position in case the cost function of the new location is better than the global best

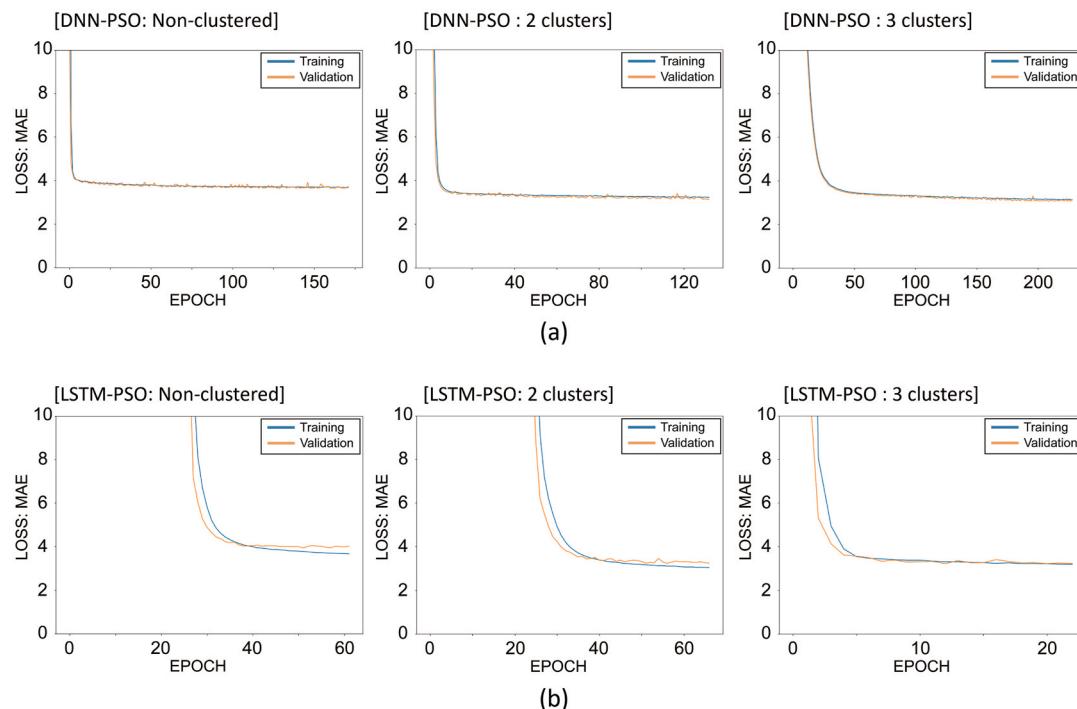


Fig. 14. Loss and epoch: (a) DNN-PSO model and (b) LSTM-PSO model.

**Table 7**  
Model accuracy (SVR-PSO).

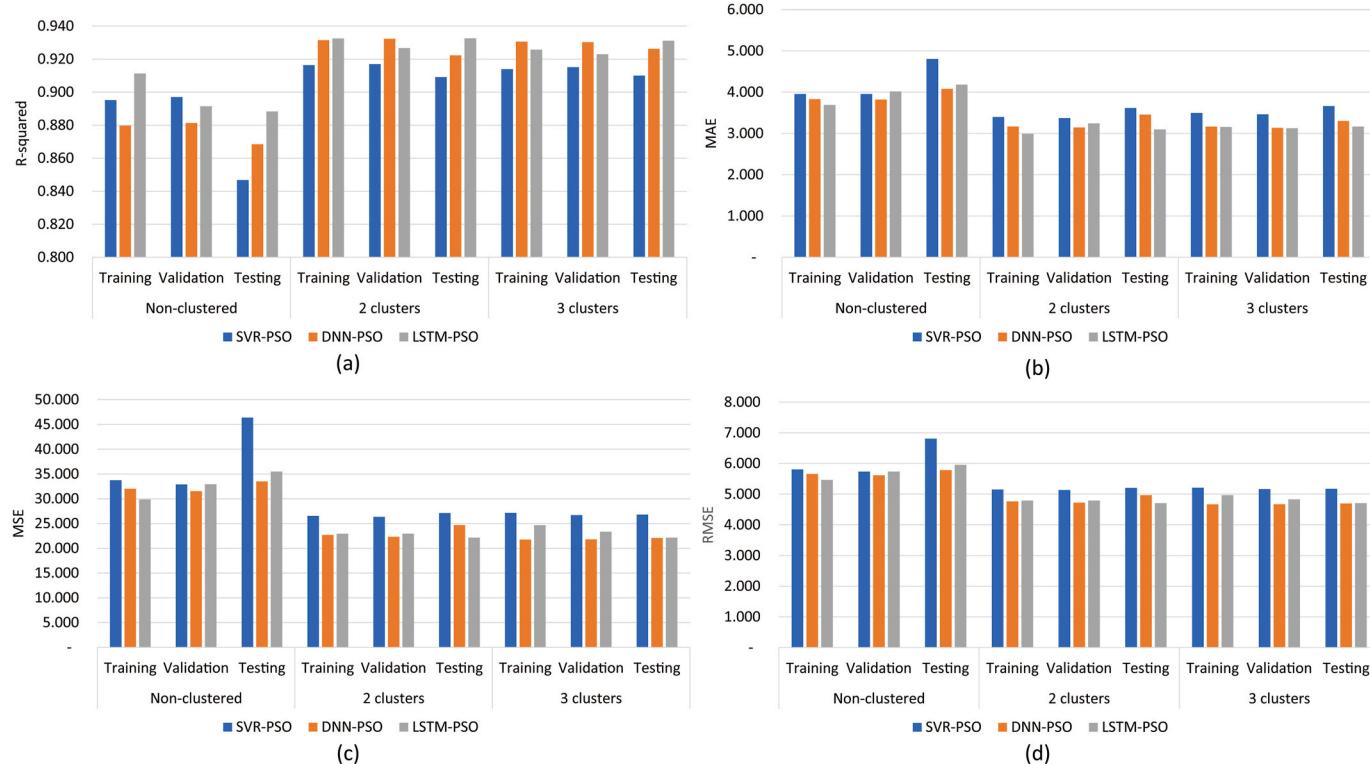
	Non-clustered			2 clusters			3 clusters		
	Training	Validation	Test	Training	Validation	Test	Training	Validation	Test
R <sup>2</sup>	0.895	0.897	0.847	0.916	0.917	0.909	0.914	0.915	0.910
MAE	3.956	3.954	4.804	3.397	3.370	3.612	3.498	3.461	3.664
MSE	33.746	32.893	46.400	26.550	26.346	27.112	27.145	26.703	26.796
RMSE	5.809	5.735	6.812	5.153	5.133	5.207	5.210	5.168	5.177

**Table 8**  
Model accuracy (DNN-PSO).

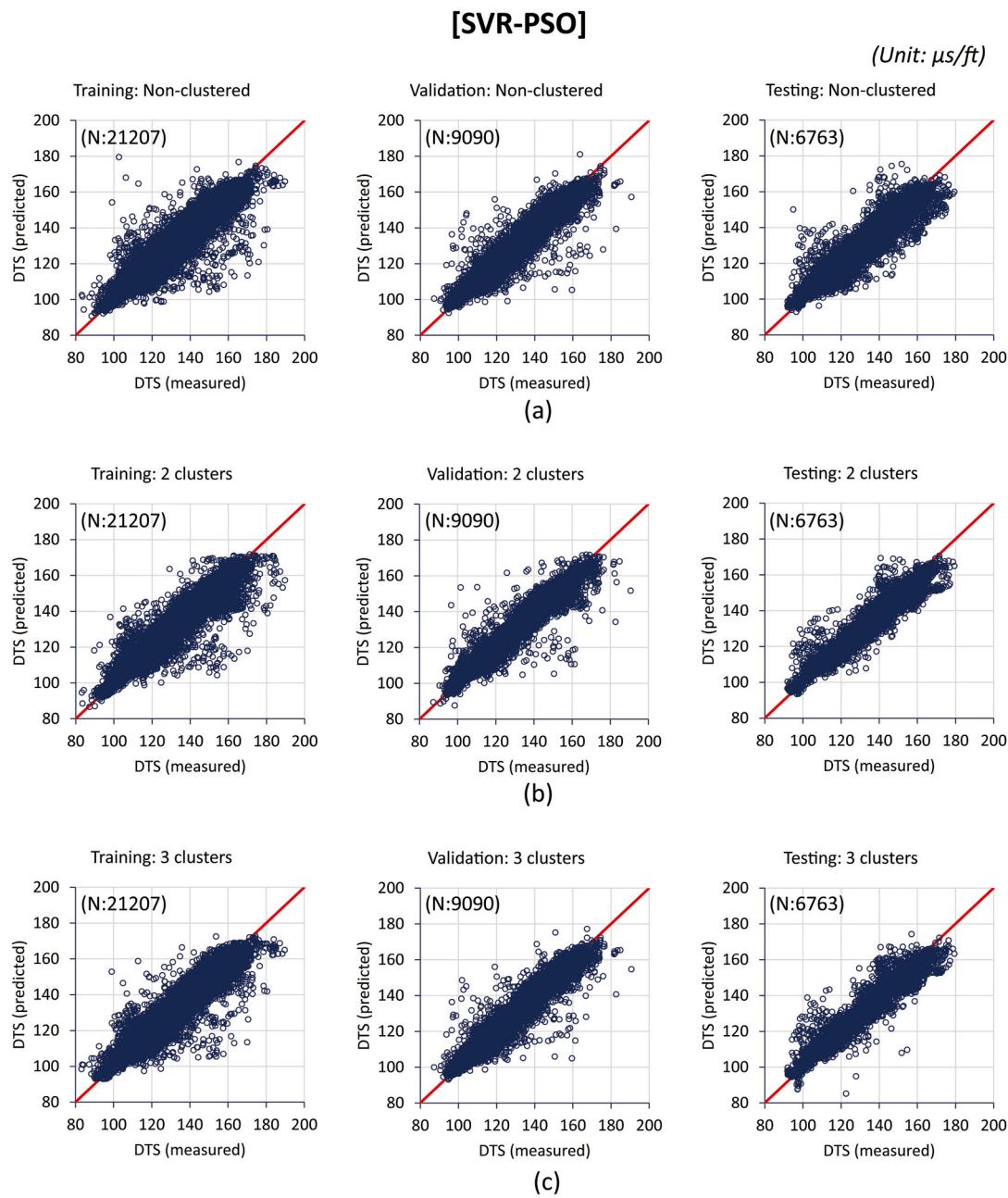
	Non-clustered			2 clusters			3 clusters		
	Training	Validation	Test	Training	Validation	Test	Training	Validation	Test
R <sup>2</sup>	0.880	0.881	0.868	0.931	0.932	0.922	0.931	0.930	0.926
MAE	3.831	3.818	4.081	3.169	3.142	3.457	3.164	3.137	3.301
MSE	32.029	31.531	33.487	22.705	22.338	24.700	21.773	21.808	22.078
RMSE	5.659	5.615	5.787	4.765	4.726	4.970	4.666	4.670	4.699

**Table 9**  
Model accuracy (LSTM-PSO).

	Non-clustered			2 clusters			3 clusters		
	Training	Validation	Test	Training	Validation	Test	Training	Validation	Test
R <sup>2</sup>	0.911	0.891	0.888	0.932	0.927	0.933	0.926	0.923	0.931
MAE	3.691	4.016	4.180	2.990	3.242	3.097	3.154	3.125	3.166
MSE	29.864	32.927	35.484	22.946	22.948	22.143	24.669	23.333	22.133
RMSE	5.465	5.738	5.957	4.790	4.790	4.706	4.967	4.830	4.705



**Fig. 15.** Comparison of the machine learning results. (a) Coefficient of determination ( $R^2$ ), (b) Mean Absolute Error, (c) Mean Square Error, and (d) Root Mean Square Error. Note that the sample configuration of training and validation datasets is different between LSTM and SVR/DNN. While the LSTM split the training (70%) and validation data (30%) as depth order (i.e., not mixing samples) from input wells (eight wells), the SVR and DNN algorithms split the two datasets after randomly mixing samples from the same input wells. However, for testing, these three algorithms employed the same dataset (two blind wells) in common. For this reason, testing results are appropriate to compare the performances of the three machine learning algorithms in this study.



**Fig. 16.** Crossplots between measured and predicted DTS by SVR-PSO models using (a) non-clustered data, (b) 2-cluster data, and (c) 3-cluster data.

position in history. As schematic flowchart is illustrated in Fig. 12.

$$V_i(t+1) = wV_i(t) + c_1r_1(Pb_i(t) - x_i(t)) + c_2r_2(G_b(t) - x_i(t)) \quad (13)$$

$$x_i(t+1) = x_i(t) + V_i(t+1) \quad (14)$$

where  $V_i$  is the velocity of  $i$ -th particle and  $w$  is inertia weight that affects the recurrence of particle velocity,  $r_1$  and  $r_2$  are random numbers between 0 and 1,  $C_1$  and  $C_2$  are personal and social factor respectively.

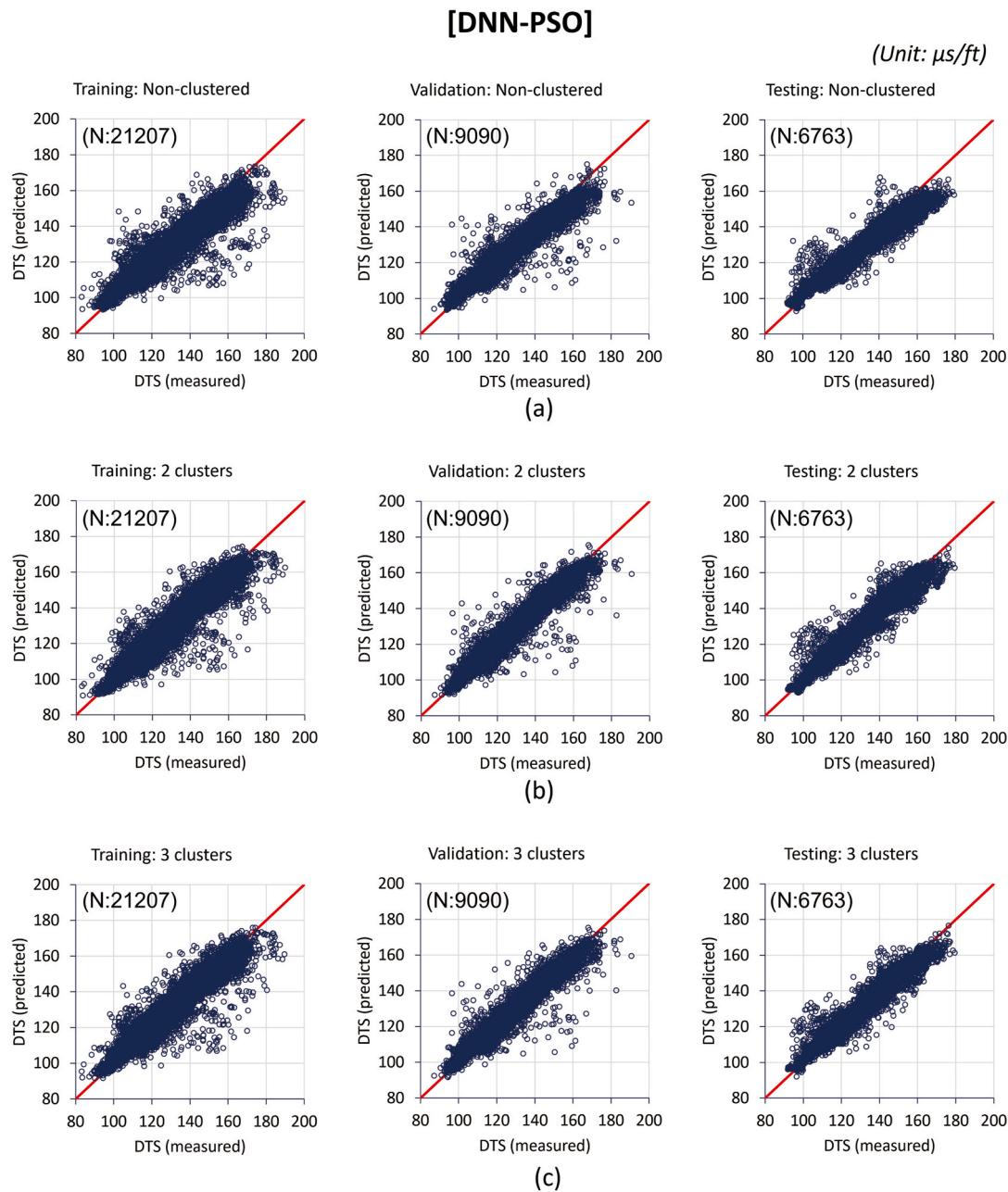
#### 2.6.2. Model and hyperparameter test

Using the PSO algorithm, a wide range of hyperparameters were tested to find optimal solutions for nine cases (i.e., three models (SVR, DNN, and LSTM) X three clustering cases (no cluster, two-cluster, and three-cluster case)). The candidates of hyperparameters and input logs are listed in Tables 4 and 5. For SVR models,  $C$  (regularization term), gamma (defining how far the influence of a single training point),

epsilon (generalization margin), kernel type, and degree (for polynomial kernel type) were determined through the optimization. For DNN and LSTM, the number of layers and neurons, dropout rate, learning rate, timesteps (LSTM), and batch size were determined. For each machine learning model, 20 times of iteration were carried out using 10 particles, which means that each machine learning model was run as much as 200 times with different hyperparameter settings. Inertia weight ( $w = 0.7$ ), personal factor ( $C_1 = 2$ ), and social factor ( $C_2 = 2$ ) were determined from a range of values used for previous studies (Bansal et al., 2011; Anemangely et al., 2019; Zhu and Wang, 2018). Callback and early stopping functions (Maximum epoch: 500, Patience: 10) were applied to reduce unnecessary execution time during each run of DNN and LSTM.

#### 2.7. Evaluation metrics

The Mean Absolute Error (MAE) was used as an objective function



**Fig. 17.** Crossplots between measured and predicted DTS by DNN-PSO models using (a) non-clustered data, (b) 2-cluster data, and (c) 3-cluster data.

for the DNN and LSTM models. And four statistical indicators, such as coefficient of determination ( $R^2$ ), mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE), were all compared in order to evaluate the model performances. Each term is calculated as follows (Eq. (15)–(18)):

$$R^2 = 1 - \frac{\sum_{i=1}^n (x_i - y_i)^2}{\sum_{i=1}^n (x_i - X)^2} \quad (15)$$

$$MAE = \frac{1}{N} \sum_{i=1}^n |x_i - y_i| \quad (16)$$

$$MSE = \frac{1}{N} \sum_{i=1}^n (x_i - y_i)^2 \quad (17)$$

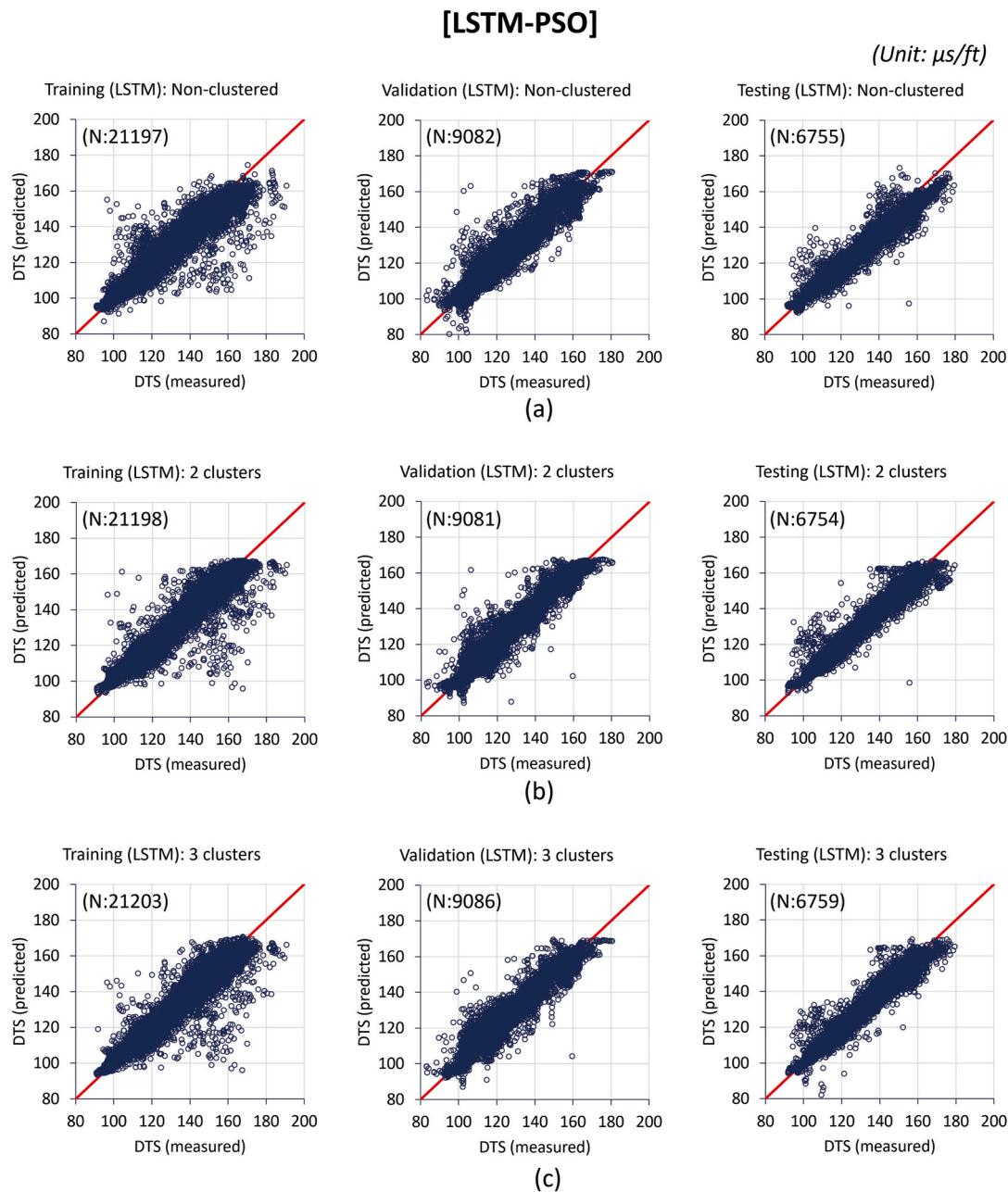
$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^n (x_i - y_i)^2} \quad (18)$$

where  $x_i$  and  $y_i$  are the measured value of DTS and the predicted value of DTS from machine learning models, respectively.  $X$  is the average value of the  $x_i$ .  $N$  is the total number of samples.

### 3. Result and discussion

#### 3.1. Hyperparameter optimization by PSO algorithm

**Fig. 13** shows the reduction trend of the objective functions (MAE) with an increasing iteration in the optimization. Each iteration means 10 runs using a different set of hyperparameters. In most cases, the objective function decreased by the 10–12th iteration (100–120th runs since the initiation). Afterward, there were no significant decreases, so each

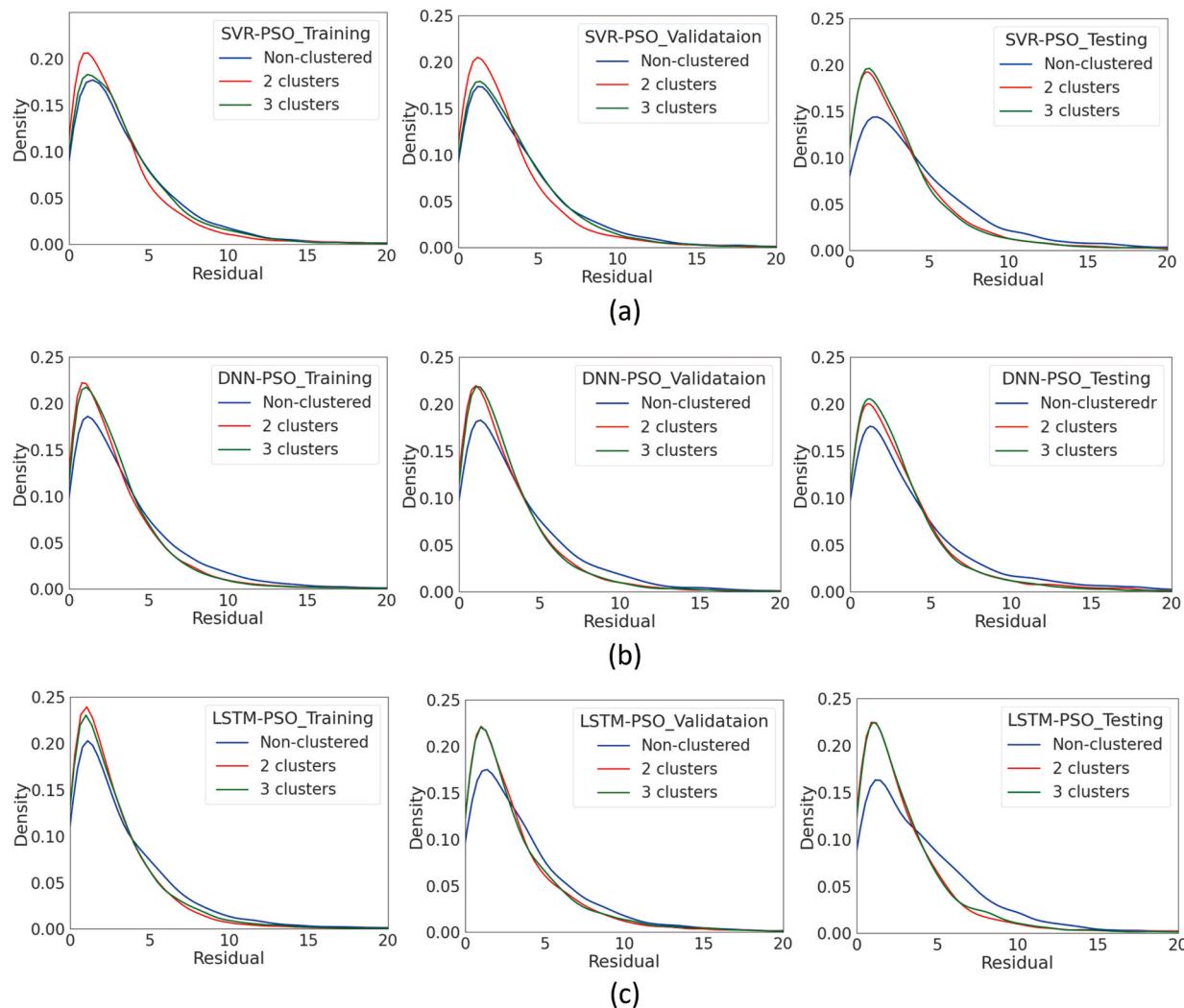


**Fig. 18.** Crossplots between measured and predicted DTS by LSTM-PSO models using (a) non-clustered data, (b) 2-cluster data, and (c) 3-cluster data.

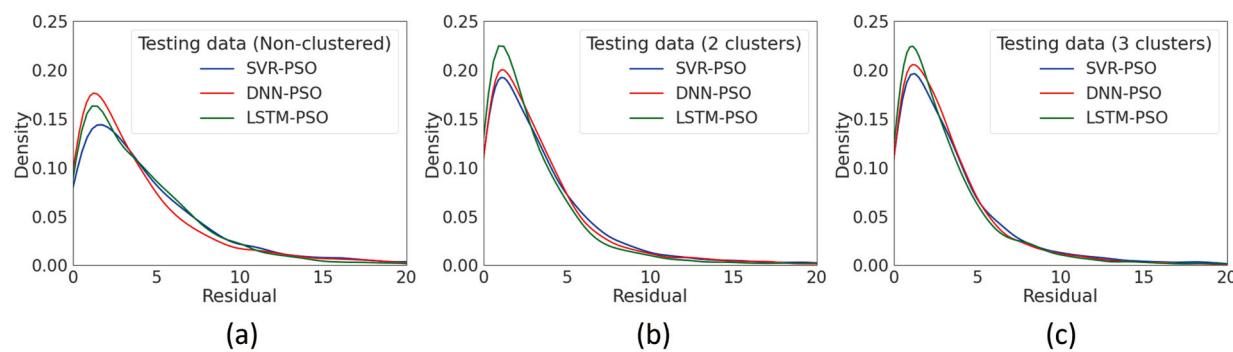
PSO process is regarded close to the optimal result. The main difference between the random search method and the PSO is that the random search method selects hyperparameters randomly whereas the PSO is guided to the optimal condition by previous hyperparameter selections and results. For this reason, the PSO can efficiently narrow down optimal hyperparameters. Considering the possible combinations of hyperparameters for an LSTM model is over 200 million cases in this study, the PSO algorithm could be a more efficient approach to reduce uncertainty with respect to searching optimal hyperparameters compared to the random search or manual adjustments. The optimal hyperparameters determined by the PSO algorithm are listed in Table 6. Based on the optimal hyperparameters, final models were established. Fig. 14 shows the loss function trend over increasing epoch for final DNN and LSTM models using the optimal hyperparameters.

### 3.2. Model performance comparison

Tables 7–9 and Fig. 15 summarize the performances of SVR-PSO, DNN-PSO, and LSTM-PSO models for three data clustering cases. In all the cases, machine learning models using clustered datasets showed better performances compared to models using the non-clustered datasets. In SVR-PSO models, the non-clustered model showed accuracies of 0.847–0.897 in  $R^2$ , 3.956–4.804 in MAE, 32.893–46.400 in MSE, and 5.735–6.812 in RMSE while clustered models (i.e., 2, and 3 clusters) showed 0.909–0.971 in  $R^2$ , 3.370–3.664 in MAE, 26.346–26.796 in MSE, and 5.133–5.207 in RMSE. DNN-PSO and LSTM-PSO models also showed better performances when using clustered data. In DNN-PSO and LSTM-PSO models using non-clustered data, the accuracy is 0.868–0.911 in  $R^2$ , 3.691–4.180 in MAE, 29.864–35.484 in MSE, 5.465–5.957 in RMSE while models using clustered data is in 0.922–0.933 in  $R^2$ , 2.990–3.301 in MAE, 22.143–24.669 in MSE, and 4.765–4.970 in RMSE. This supports that unsupervised classification can



**Fig. 19.** Residual (MAE) distribution (Cluster case comparison): (a) SVR-PSO, (b) DNN-PSO, (c) LSTM-PSO models.



**Fig. 20.** Residual (MAE) distribution (Model comparison using testing data): (a) non-clustered case, (b) 2-cluster case, (c) 3-cluster case.

**Table 10**  
Execution time (Unit: Second).

Case	SVR-PSO	DNN-PSO	LSTM-PSO
Non-cluster	8030	12199	17130
2 clusters	8237	13841	17828
3 clusters	8438	13212	18237

be efficient to enhance the model performance in regression problems. Back in Figs. 8 and 9., each cluster shows mostly isolated, distinguishable features in data distributions. This classification makes an effect setting limited upper and lower bounds for each group, which ultimately narrows down the possible target range for data samples in a specific cluster compared to when using non-clustered data that has a relatively wider target range. In the crossplot between predicted DTS and measured DTS (Figs. 16–18), distributions of non-clustered data cases (Fig. 16-(a), 17-(a), 18-(a)) are more scattered than clustered data cases

**Table 11**

Sensitivity test using different combinations of testing wells.

Training/Validation (8 wells)	Testing (2 wells)	Training				Validation				Testing			
		R2	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2	MAE	MSE	RMSE
Other 8 wells	Well A and B	0.922	3.159	25.043	5.004	0.930	3.120	20.781	4.559	0.933	3.120	21.248	4.610
	Well C and D	0.921	3.297	26.944	5.191	0.932	3.123	21.045	4.587	0.930	3.381	22.886	4.784
	Well E and F	0.917	3.307	26.700	5.167	0.930	3.122	21.082	4.592	0.926	3.292	22.853	4.781
	Well G and H	0.928	3.136	24.730	4.973	0.932	3.120	21.256	4.610	0.936	3.104	21.217	4.606
	Well I and J	0.926	3.154	24.669	4.967	0.923	3.125	23.333	4.830	0.931	3.166	22.133	4.705

(Fig. 16-(b), (c), 17-(b), (c), and 18-(b), (c)), which supports performance enhancements by the data-driven classification. Especially models using non-clustered datasets showed relatively bigger residuals at specific intervals that have high DTS (See the attachment, 1340–1350m and 2050–2070m intervals in Fig. A-1 to 3, 2255–2275m interval in Fig. A-4 to 6). Residual distributions (Fig. 19) also support the enhanced model performance by the data-clustering. These distribution curves were made by kernel density estimation (KDE) in order to see a continuous data distribution without discrete bin boundaries. Both 2- and 3-clustered datasets show relatively higher concentrations at low residual values compared to the non-clustered dataset. In machine learning model comparison, the LSTM-PSO and DNN-PSO models showed better accuracies than the SVR-PSO model. The comparison was carried out using test datasets (i.e., two blind wells). This is because the input data (i.e., training and validation data) configuration is not the same among these three models. LSTM-PSO model uses the depth-ordered sequential data without data mixing whereas SVR-PSO and DNN-PSO models use randomly mixed data. However, these three models use two blind wells as testing data in common. For this reason, the test dataset is appropriate to compare the model performances. SVR-PSO models showed accuracies of 0.847–0.910 in  $R^2$ , 3.612–4.804 in MAE, 26.796–46.400 in MSE, and 5.177–6.812 in RMSE. DNN-PSO models' accuracies are 0.868–0.926 in  $R^2$ , 3.301–4.081 in MAE, 22.078–33.487 in MSE, and 4.699–5.787 in RMSE. LSTM-PSO models' accuracies are in 0.888–0.931 in  $R^2$ , 3.079–4.180 in MAE, 22.133–35.484 in MSE, 4.705–5.957 in RMSE. LSTM-PSO models were better at  $R^2$  and MAE compared to DNN-PSO models but in terms of MSE and RMSE, DNN-PSO models were better than LSTM-PSO models. This indicates that LSTM-PSO models are more impacted by outliers despite overall high accuracies compared to DNN-PSO models. In Fig. 20, LSTM-PSO models show relatively better concentrations at low residuals (<5 in MAE), but it doesn't maintain the superiority at high residuals (>10 in MAE). In addition, LSTM-PSO models have more large outliers compared to DNN-PSO models (Figs. 17 and 18), which is supposed to give an additional impact on MSE and RMSE. The execution time of each model is listed in Table 10. Each execution time is the result of 200 model runs in the PSO process. SVR-PSO models took the smallest execution time while LSTM-PSO models took the largest.

### 3.3. Sensitivity analysis using different combinations of testing wells

Additional analysis was conducted using different combinations of testing wells in order to see if the study result could be impacted by specific data configuration. Five cases were designed to have different two wells as testing data (Table 11). Each case with different training, validation, and testing datasets shows overall consistent results.

## 4. Conclusion

This study proposes a data-driven approach to generating the synthetic shear sonic log (DTS) using five types of wireline logs (GR, RHOB, NPHI, RT, and DTC) that are commonly accessible in hydrocarbon production fields or exploration sites. Through the feature selection process using the Pearson correlation and multi-linear regression model, the five types of wireline logs were decided to be used as input data. A

hybrid machine learning model was introduced for the purpose of enhancing the accuracy in predicting DTS log values. An unsupervised classification algorithm utilizing K-means clustering was combined with three types of machine learning models such as SVR, DNN, and LSTM. Optimal hyperparameters were determined by the PSO algorithm. The main findings are as follows:

- 1) In the comparison of prediction accuracies between data-clustered models and non-clustered models, data clustered models consistently showed better performances compared to non-clustered models.
- 2) The PSO algorithm was employed for automated hyperparameter tuning. Model performances improved by 10 to 12th iteration which is equivalent to 100 to 120 model runs with different hyperparameter combinations. Considering a significant number of possible hyperparameter combinations (e.g., over 200 million cases for the LSTM model in this study), the PSO is recommendable for searching optimal hyperparameters. The PSO is a time-effective, automated approach backed by the advantage of getting feedback from previous trials and being able to gradually narrow down candidates for optimal hyperparameters.
- 3) Three machine learning models showed fair to good performances in DTS log predictions. LSTM-PSO and DNN-PSO models outperformed over SVR-PSO models. LSTM-PSO models were better than DNN models in  $R^2$ , MAE, and residual distributions, but it contains more outliers that ultimately raise MSE and RMSE higher than DNN-PSO models.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.aiig.2022.09.001>.

## References

- Akhundi, H., Ghafoori, M., Lashkaripour, G., 2014. Prediction of shear wave velocity using artificial neural network technique, multiple regression and petrophysical data: a case study in Asmari reservoir (SW Iran), 2014 Open J. Geol. 303–313. <https://doi.org/10.4236/ojg.2014.47023>.
- Alexeyev, A., Ostadhassan, M., Mohammed, R.A., Bubach, B., Khatibi, S., 2017. Well Log Based Geomechanical and Petrophysical Analysis of the Bakken Formation, 51st US Rock Mechanics/Geomechanics Symposium Held in San Francisco, California, USA, pp. 25–28, June 2017.
- Ali, M., Jiang, R., Ma, H., Pan, H., Abbas, K., Ashraf, U., 2021. Machine learning - a novel approach of well logs similarity based on synchronization measures to predict shear sonic logs. J. Petrol. Sci. Eng. 203 <https://doi.org/10.1016/j.petrol.2021.108602>.
- Al-Mudhafar, W.J.M., Bondarenko, M.A., 2015. Integrating K-means clustering analysis and generalized additive model for efficient reservoir characterization. In: 77th EAGE Conference and Exhibition 2015, vol. 1, pp. 1–6 (European Association of Geoscientists & Engineers).
- Ameen, M.S., Smart, B.G.D., Somerville, J. Mc, Hamilton, S., Naji, N.A., 2009. Predicting rock mechanical properties of carbonates from wireline logs (A case study: arab-D reservoir, Ghawar field, Saudi Arabia). Mar. Petrol. Geol. 26 (4), 430–444. <https://doi.org/10.1016/j.marpetgeo.2009.01.017>.

- Anemangely, M., Ramezanzadeh, A., Tokhmechi, B., 2017. Shear wave travel time estimation from petrophysical logs using ANFIS-PSO algorithm: a case study from Ab-Teymour Oilfield. *J. Nat. Gas Sci. Eng.* 38, 373–387. <https://doi.org/10.1016/j.jngse.2017.01.003>.
- Anemangely, M., Ramezanzadeh, A., Amiri, H., Hoseinpour, S.A., 2019. Machine learning technique for the prediction of shear wave velocity using petrophysical logs. *J. Petrol. Sci. Eng.* 174, 306–327. <https://doi.org/10.1016/j.petrol.2018.11.032>.
- Bagheripour, P., Gholami, A., Asoodeh, M., Vaezzadeh-Asadi, M., 2015. Support vector regression based determination of shear wave velocity. *J. Petrol. Sci. Eng.* 125, 95–99. <https://doi.org/10.1016/j.petrol.2014.11.025>.
- Bansal, J.C., Singh, P.K., Saraswat, M., Verma, A., Jadon, S.S., Abraham, A., 2011. Inertia weight strategies in particle swarm optimization. In: Third World Congress on Nature and Biologically Inspired Computing, pp. 633–640. <https://doi.org/10.1109/NaBIC.2011.6089659>.
- Bonyadi, M.R., Michalewicz, Z., 2017. Particle swarm optimization for single objective continuous space problems: a review. *Evol. Comput.* 25 (1), 1–54. [https://doi.org/10.1162/EVCO\\_00180.PMID26953883.S2CID8783143](https://doi.org/10.1162/EVCO_00180.PMID26953883.S2CID8783143).
- Brantut, N., David, E.C., 2019. In: *Geophysical Journal International, Influence of Fluids on VP/VS Ratio: Increase or Decrease?*, vol. 216, pp. 2037–2043. <https://doi.org/10.1093/gji/ggy518>.
- Brocher, T.M., 2005. Empirical relations between elastic wave speeds and density in the earth's crust. *Bull. Seismol. Soc. Am.* 95 (6), 2081. <https://doi.org/10.1785/0120050077>.
- Bukar, I., Adamu, M.B., Hassan, U., 2019. A machine learning approach to shear sonic log prediction. *Nigeria. Annu. Conf. Int. Conf. Exhibit.* <https://doi.org/10.2118/198764-MS>. August 5–7, 2019.
- Castagna, J.P., Batzle, M.L., Eastwood, R.L., 1985. Relationships between compressional-wave and shear-wave velocities in clastic silicate rocks. *Geophysics* 50 (4). <https://doi.org/10.1190/1.1441933>.
- Chaikine, I.A., Ian, D.G., 2020. A new machine learning procedure to generate highly accurate synthetic shear sonic logs in unconventional reservoirs. Virtual, October 2020. In: *SPE Annual Technical Conference and Exhibition*. <https://doi.org/10.2118/201453-MS>.
- Close, D., Cho, D., Horn, F., Edmundson, H., 2009. The sound of sonic: a historical perspective and introduction to acoustic logging. *CSEG Recorder* 34 (5), 34–43.
- Greenberg, M.L., Castagna, J.P., 1992. Shear-wave velocity estimation in porous rocks: theoretical formulation, preliminary verification and applications. *Geophys. Prospect.* 40 (2), 195–209. <https://doi.org/10.1111/j.1365-2478.1992.tb00371.x>.
- He, J., Li, H., Misra, S., 2019. Data-driven in-situ sonic-log synthesis in shale reservoirs for geomechanical characterization. *SPE Reservoir Eval. Eng.* 22 (4), 1225–1239. <https://doi.org/10.2118/191400-PA>.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Iverson, W.P., Walker, J.N., 1988. Shear and compressional log derived from nuclear logs. In: *SEG Technical Program Expanded Abstracts 1988*, pp. 111–113. <https://doi.org/10.1190/1.1892237>. Tulsa: Society of Exploration Geophysicists.
- Jeong, J., Park, E., Emelyanova, I., Pervukhina, M., Esteban, L., Yun, S., 2021. Application of conditional generative model for sonic log estimation considering measurement uncertainty. *J. Petrol. Sci. Eng.* 196 <https://doi.org/10.1016/j.petrol.2020.108028>.
- Kang, B., Choe, J., 2020. Uncertainty quantification of channel reservoirs assisted by cluster analysis and deep convolutional generative adversarial networks. *J. Petrol. Sci. Eng.* 187 <https://doi.org/10.1016/j.petrol.2019.106742>.
- Ketchen, D.J., Shook, C.L., 1996. The application of cluster analysis in strategic management research: an analysis and critique. *Strat. Manag. J.* 17, 441–458. [https://doi.org/10.1002/\(SICI\)1097-0266\(199606\)17:6<441::AID-SMJ819>3.0.CO;2-G](https://doi.org/10.1002/(SICI)1097-0266(199606)17:6<441::AID-SMJ819>3.0.CO;2-G).
- Lee, M.W., 2006. A simple method of predicting S-wave velocity. *Geophysics* 71 (6), F161. <https://doi.org/10.1190/1.2357833>.
- Lu, C., Lee, T.S., Chiu, C.C., 2009. Financial time series forecasting using independent component analysis and support vector regression. *Decis. Support Syst.* 47 (2), 115–125. <https://doi.org/10.1016/j.dss.2009.02.001>.
- Maleki, S., Moradzadeh, A., Riabi, R.G., Gholami, R., Sadeghzadeh, F., 2014. Prediction of shear wave velocity using empirical correlations and artificial intelligence methods. *NRIAG J. Astron. Geophys.* 3 (1), 70–81. <https://doi.org/10.1016/j.nrjag.2014.05.001>.
- Mavko, G., Mukerji, T., Dvorkin, J., 2009. *The Rock Physics Handbook*, second ed. Cambridge University Press.
- Olayiwola, T., Sanuade, O.A., 2021. A data-driven approach to predict compressional and shear wave velocities in reservoir rocks. *Petroleum* 7 (2), 199–208. <https://doi.org/10.1016/j.petlm.2020.07.008>.
- Oloruntobi, O., Butt, S., 2020. The shear-wave velocity prediction for sedimentary rocks. *J. Nat. Gas Sci. Eng.* 76 <https://doi.org/10.1016/j.jngse.2019.103084>.
- Pham, N., Naeini, E.Z., 2019. Missing Well Log Prediction Using Deep Recurrent Neural Networks. In: *81<sup>st</sup> EAGE Conference & Exhibition*, 2019.
- Rajabi, M., Hazebeh, O., Davoodi, S., Wood, D.A., Tehrani, P.S., Ghorbani, H., Mehrad, M., Mohamadian, N., Rukavishnikov, V.S., Radwan, A.E., 2022. Predicting shear wave velocity from conventional well logs with deep and hybrid machine learning algorithms. *J. Pet. Explor. Prod. Technol.* <https://doi.org/10.1007/s13202-022-01531-z>.
- Raschka, S., Mirjalili, V., 2019. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow* 2, third ed. Packt Publishing, ISBN 9781789955750.
- Rasouli, V., Pallikathethathil, Z.J., Mawuli, E., 2011. The influence of perturbed stresses near faults on drilling strategy: a case study in Blacktip field, North Australia. *J. Petrol. Sci. Eng.* 76 (1–2) <https://doi.org/10.1016/j.petrol.2010.12.003>.
- Rezaee, M.R., Ilkhchi, A.K., Barabadi, A., 2007. Prediction of shear wave velocity from petrophysical data utilizing intelligent systems: an example from a sandstone reservoir of Carnarvon Basin, Australia, 3–4. *J. Petrol. Sci. Eng.* 55, 201–212. <https://doi.org/10.1016/j.petrol.2006.08.008>.
- Saumya, S., Narasimhan, B., Singh, J., Yamamoto, H., Vij, J., Sakiyama, N., Kumar, P., 2019. Acquisition of logging-while-drilling (LWD) multipole acoustic log data during the India national gas hydrate program (NGHP) expedition 02, 8172 Mar. *Petrol. Geol.* 108, 562–569. <https://doi.org/10.1016/j.marpetgeo.2018.10.011>.
- Tan, M.J., Song, X.D., Yang, X., Wu, Q.Z., 2015. Support-vector-regression machine technology for total organic carbon content prediction from wireline logs in organic shale: a comparative study. *J. Nat. Gas Sci. Eng.* 26, 792–802. <https://doi.org/10.1016/j.jngse.2015.07.008>.
- Wang, B., Zhou, F., Zou, Y., Liang, T., Wang, D., Xue, Y., Gao, L., 2019. Quantitative investigation of fracture interaction by evaluating fracture curvature during temporarily plugging staged fracturing. *J. Petrol. Sci. Eng.* 172, 559–571. <https://doi.org/10.1016/j.petrol.2018.08.038>.
- Wang, H., Wu, W., Chen, T., Dong, X., Wang, G., 2019. An improved neural network for TOC, S1 and S2 estimation based on conventional well logs. *J. Petrol. Sci. Eng.* 176, 664–678. <https://doi.org/10.1016/j.petrol.2019.01.096>.
- You, J., Cao, J., Wang, X., Liu, W., 2021. Shear wave velocity prediction based on LSTM and its application for morphology identification and saturation inversion of gas hydrate. *J. Petrol. Sci. Eng.* 205 <https://doi.org/10.1016/j.petrol.2021.109027>.
- Zhang, D., Chen, Y., Meng, J., 2018. Synthetic well logs generation via recurrent neural networks. *Petrol. Explor. Dev.* 45 (4), 629–639. [https://doi.org/10.1016/S1876-3804\(18\)30068-5](https://doi.org/10.1016/S1876-3804(18)30068-5).
- Zhu, X., Wang, H., 2018. A new inertia weight control strategy for particle swarm optimization. *AIP Conf. Proc.* 1955 (1) <https://doi.org/10.1063/1.5033759>.
- Zimmer, M., Prasad, M., Mavko, G., 2002. Pressure and porosity influences on VP-VS ratio in unconsolidated sands. *Lead. Edge* 21 (2), 178–183. <https://doi.org/10.1190/1.1452609>.