

## Research Article

# Generation of Synthetic Density Log Data Using Deep Learning Algorithm at the Golden Field in Alberta, Canada

Sungil Kim <sup>1</sup>, Kwang Hyun Kim <sup>1</sup>, Baehyun Min <sup>2</sup>, Jungtek Lim,<sup>3</sup>  
and Kyungbook Lee <sup>1</sup>

<sup>1</sup>Petroleum and Marine Research Division, Korea Institute of Geoscience and Mineral Resources, 124, Gwahak-ro, Yuseong-gu, Daejeon 34132, Republic of Korea

<sup>2</sup>Department of Climate and Energy Systems Engineering, Ewha Womans University, 52, Ewhayeodae-gil, Seodaemun-gu, Seoul 03760, Republic of Korea

<sup>3</sup>SmartMind, Inc., 1006, 379, Yeouidaebang-ro, Yeongdeungpo-gu, Seoul 03760, Republic of Korea

Correspondence should be addressed to Kyungbook Lee; [kblee@kigam.re.kr](mailto:kblee@kigam.re.kr)

Received 31 May 2019; Revised 11 November 2019; Accepted 13 December 2019; Published 30 January 2020

Academic Editor: Umberta Tinivella

Copyright © 2020 Sungil Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study proposes a deep neural network- (DNN-) based prediction model for creating synthetic log. Unlike previous studies, it focuses on building a reliable prediction model based on two criteria: fit-for-purpose of a target field (the Golden field in Alberta) and compliance with domain knowledge. First, in the target field, the density log has advantages over the sonic log for porosity analysis because of the carbonate depositional environment. Considering the correlation between the density and sonic logs, we determine the sonic log as input and the density log as output for the DNN. Although only five wells have a pair of training data in the field (i.e., sonic and density logs), we obtain, based on geological knowledge, 29 additional wells sharing the same depositional setting in the Slave Point Formation. After securing the data, 5 wells among the 29 wells are excluded from dataset during preprocessing procedures (elimination of abnormal data and min–max normalisation) to improve the prediction model. Two cases are designed according to usage of the well information at the target field. Case 1 uses only 23 of the surrounding wells to train the prediction model, and another surrounding well is used for model testing. In Case 1, the Levenberg–Marquardt algorithm shows a fast and reliable performance and the numbers of neurons in the two hidden layers are of 45 and 14, respectively. In Case 2, the 24 surrounding wells and four wells from the target field are used to train the DNN with the optimised parameters from Case 1. The synthetic density logs from Case 2 mitigate an underestimation problem in Case 1 and follow the overall trend of the true density logs. The developed prediction model utilises the sonic log for generating the synthetic density log, and a reliable porosity model will be created by combining the given and the synthetic density logs.

## 1. Introduction

Reservoir modelling is an essential work to understand and assess a target reservoir, and a reservoir model is used to implement a reservoir simulation for preparing a field development plan. Well logging is the most important data in reservoir modelling. Based on the petrophysical properties determined from well logging, a spatial correlation (e.g., variogram) is estimated and geostatistical algorithms are applied to build a reliable, three-dimensional reservoir model. Even though reservoir modelling is affected by the amount of

available well log data, data shortage problem always exists because the well log data can only be acquired through an expensive drilling process. Sometimes, although well log data are obtained by drilling, a specific log type that predicts the desired reservoir properties may not have been measured. For instance, when a porosity model is needed, sonic or density logs may not be obtained during well logging. Moreover, the log data may be missing for the depth of interest. In these cases, a solution is to acquire additional well log data by new drilling or by rerunning the well logging to obtain the required log type for an already drilled well. However, drilling

a new well or stopping production to rerun logging causes a huge additional cost, and some log types are not measurable owing to casing [1–3].

Lately, there have been researches on how to handle these problems by generating synthetic (or pseudo) well log data using the concept of machine learning. Rolon et al. designed three synthetic well log prediction models, i.e., resistivity, density, and neutron logs, in the northeast of the United States [4]. They mentioned that the geometry of the training wells was not related with the performance of a neural network. Besides, the quality control of training data considerably affected the trained prediction model. Long et al. built synthetic density log with a concept of pairwise well prediction [3]. Even though eight wells were available in the field, they chose only one ideal well to train a neural network. However, the two aforementioned studies used a simple neural network model with a single hidden layer [3, 4].

Salehi et al. trained a deep neural network (DNN) with three hidden layers using two wells from a carbonate oil reservoir on the southwest of Iran [2]. The target field consisted of eight pays, but they chose one pay zone to generate the prediction models because each pay had different lithology. They generated three prediction models (i.e., true resistivity, sonic, and shallow resistivity logs) with seven input logs, including water saturation, density, neutron, and deep resistivity. Therefore, several well logs will be necessarily required to use the trained prediction models.

Korjani et al. built a resistivity prediction model for heavy oil reservoirs in Joaquin Valley, California [5]. They compared three strategies for the input layer: information of surrounding wells (angle and distance), kriging coefficient, and fuzzy kriging range. After more than 1,200 wells were trained for each strategy, the pseudoresistivity log from the fuzzy case showed the best performance. When the synthetic log was used to build a three-dimensional facies model, it displayed the geological trend (e.g., channel connectivity) properly. Previous studies have predicted missing logs where wells already existed, but in the case of [5], they created a neural network model that predicts log data at arbitrary locations without drilling. Instead of the information of a target location, well log data of 10 surrounding wells and their location data were used to train a prediction model. Further, they predicted three logs simultaneously. However, as a considerable amount of well information was needed to train the model, it will be difficult to apply to areas where the surrounding well information is limited.

Akinnikawe et al. compared several machine learning algorithms (e.g., artificial neural networks (ANN), decision trees, gradient boosting, and random forest) for generation of synthetic well logs [6]. In addition, they predicted unusual logs such as the photoelectric (PE) and unconfined compressive strength (UCS) logs because the PE log is often not measured during well logging and the USC log requires an expensive core experiment. Although the neural networks and random forest outperformed the other algorithms, they had a disadvantage of requiring more than 10 input data.

Because previous studies applied simple feed forward neural networks, the trained prediction models may not consider the sequence of log curves in depth. Zhang et al. used a

recurrent neural network to analyse well logs as sequence data [1]. They discussed that the previous ANN-based prediction models found the correlation among well logs at the same depth ignoring the geological trend in reservoirs. A long short-term memory algorithm was successfully introduced to generate missing or whole well logs for both vertical and horizontal wells in the Eagle Ford Shale. They also emphasised the importance of geological criteria for training data because a log curve has a unique hidden pattern for each stratum.

The previous studies have focused on the application of machining learning algorithms for well logging. In this research, we set two criteria for a practical application of machine learning: fit-for-purpose for a target field and domain knowledge. First, the input and output layers of a prediction model should be determined by the status of availability of well log data for a target field. If the goal is to construct a reliable porosity model and a target field lacks sonic log, a prediction model for sonic log should be trained with the available log data. However, previous prediction models tried to make a prediction model without considering which logs are needed for the prediction and why [2, 3, 6]. Therefore, a trained model required numerous well logs for the input layer and hundreds of wells were used for training [5, 7]. If we already have hundreds of well logs of several log types for a target area, a reliable reservoir model could be built without synthetic log.

Second, it is essential to preprocess the log data based on knowledge of petroleum geology and engineering, rather than using all available data. In [8], because of the importance of geological criteria, they generated a prediction model by dividing the vertical log data according to formation. Therefore, the goal in this study is to apply a machine learning algorithm correctly and effectively to well logging prediction based on the given field conditions and domain knowledge.

The target field in this study is the Golden field, which belongs to the Beaverhill Lake Group and was deposited in the Western Canada Sedimentary Basin during the middle to upper Devonian age. The main production zone is the Slave Point (SLVP) Formation, which is a subdivision of the group. The depositional environment of the formation has been interpreted as a shallow marine environment and its sedimentary facies showed complex reef carbonate deposits [9–11]. Dolomitisation affected the carbonate rocks at the Golden field extensively, and dissolution developed secondary pore-like intercrystalline and interconnected pore spaces [10]. Due to the aforementioned diagenesis effects, the porosity calculated by sonic log data could be underestimated when compared to the total porosity obtained by density log data [12]. Because the estimated porosity data from well logging are critical information for geostatistics, underestimated porosity data at a well location affect the overall trend of a three-dimensional porosity model. Consequently, it will cause an inaccurate reservoir simulation result and unreasonable economic evaluation for the target field. In this study, the density log means bulk density log data.

In the case of density log, it can consider the effect of secondary porosity properly and can be utilised for identification of evaporate minerals [4]. Regarding the target field

conditions (e.g., depositional setting and dolomitisation), density log data are key information to build a reliable porosity model instead of sonic log. However, in the target field of this study, only 17 wells have density log data, and thus, additional density log data are obviously required for reasonable porosity modelling. Therefore, the density log is assigned to the output layer in a neural network.

Long et al. [3] created a prediction model of synthetic density log but it required more than 30 data for the input layer. In the target field, this trained model cannot be applied because the log data for the input layer are not available for most of the wells. Although the problem of the target field is that sonic log is not acceptable to estimate porosity, it still has high correlation with density log. Therefore, we determine the sonic log with three-dimensional coordinates (i.e., latitude, longitude, and depth) as the input layer in a neural network.

In the target field, 12 wells have the sonic log without density log. If synthetic density log can be generated from the sonic log, the reliability of a porosity model can be improved by using both the 12 synthetic density logs from the sonic log and the existing 17 density logs. Note that the basic structure of a neural network (i.e., the input and output layers) was determined based on fit-for-purpose for the target field.

A pair of input and output data points is needed to train supervised machine learning algorithms. For the target field, only five wells have both sonic and density logs. Therefore, an additional pair of data points is searched based on domain knowledge. The previous studies selected a set of train wells according to the distance between the prediction location and the available well location. However, geological similarity is more important than the physical distance. If a prediction model is trained by well log data from the same geological formation of the target reservoir, the performance of the trained model will be improved.

In this research, we examine the effect of two criteria (fit-for-purpose for a target field and domain knowledge) for preparing training data on a synthetic well log prediction model. In Section 2, we explain in detail a specific workflow of the proposed method to generate synthetic density log for the target reservoir. It consists of data acquisition, preprocessing of selected data, structuring of a neural network, and determination of hyperparameters. In Section 3, we analyse the synthetic density log from a trained prediction model. Two cases are considered depending on the usage of well information at the target field. One case generates a prediction model using information only from wells around the target field. In another case, not only the surrounding wells but also the wells belonging to the target field are included. Then, the key outcomes are summarised in Conclusions (Section 4).

## 2. Methodology

This study follows the procedure described in Figure 1. First, additional sonic and density log data are collected because only five wells in the target field have both logs. Well log data are obtained in the SLVP Formation near the Golden field from the AccuMap database (Figure 1(a)). Note that instead of simply selecting the nearest wells from the target field, we

selected additional wells that have the same depositional environment. The data consist of the sonic and density logs with the location information of the well (i.e., depth, latitude, and longitude). Preprocess procedures are applied to the obtained data, which include elimination of abnormal values and data normalisation (Figure 1(b)). After a sensitivity analysis for a default neural network structure (Figure 1(c)), a proper training function (Figure 1(d)) and the number of nodes in the hidden layers (Figure 1(e)) are fixed. Finally, the best and worst trained neural networks (Figure 1(f)) are verified by test wells (Figure 1(g)).

*2.1. Data Acquisition and Preprocessing.* An objective of this study is to suggest how to predict synthetic density log from the sonic log. Firstly, well log data, which have sonic and density data at the same time, are necessary. It is well known in deep learning that obtaining qualified training data is the most important factor to build a reliable prediction model because enough data are necessary to train a DNN properly [13, 14]. Intuitively, the closer the wells are located in the target area, the higher the quality. However, we consider not only the spatial relationship but also the geological similarity. In other words, we acquire well data in the same depositional environment rather than just based on a physical distance.

A target of this study is the SLVP Formation at the Golden field and Figure 2 shows the oil and gas wells (the black and red circles) near the target field. The empty circles mean wells with no gas or oil and the black-coloured ones are wells showing oil or gas. The SLVP Formation was deposited during the transgressive sequence, and its sequence is divided into two major sea level rise cycles according to the relative degree of rise. For this reason, the carbonate depositional environment between two cycles is different [9, 15]. In Figure 2(a), the solid blue line indicates the Bank margin, cycle 2, and the solid red line is the depositional limit of the transgressive sequence. Those two lines are supposed to be boundaries defining a depositional setting similar to that of the target field. Therefore, the blue highlighted area in Figure 2(a) indicates a territory between the two geological frontiers and it has a consistent geological environment. Even though there are lots of wells (the black color) nearby the target field (the purple circle) in Figure 2(a), only the wells located in the blue area are interested in terms of depositional setting.

However, most wells in the blue area do not have information in the SLVP Formation. The red wells were drilled into the SLVP Formation and they are located in the blue area at the same time. In the previous researches for synthetic logging, training well data have been selected by physical distance without considering domain knowledge but the data are chosen based on geological meaning in this study. Figure 2(a) is redrawn into Figure 2(b), which clearly shows the red coloured wells in the research area without the boundaries and highlighted area. Some wells have neither sonic nor density logs, although both data types are required to train a DNN model. Among the red wells, only 34 wells have both sonic and density data, and they are carried to the preprocessing stage.

Figure 3(a) presents the location of the 34 wells. First, we check the quality of the wells based on domain knowledge.

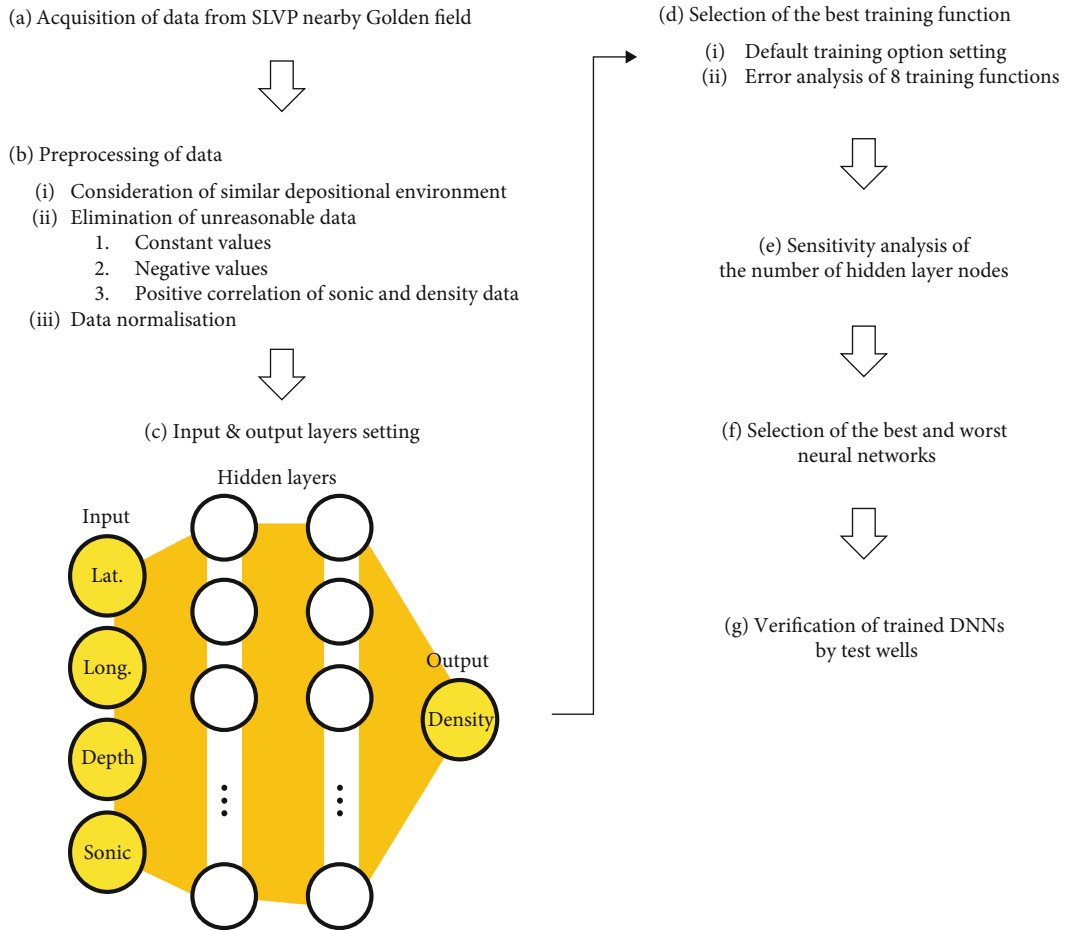


FIGURE 1: Overall procedure for the generation of pseudodensity log data using DNN.

Five wells belong to the black dotted ellipse in Figure 3(a), which indicates the Golden field. We excluded abnormal sonic and density data according to density correction log and caliper log, which provide conditions of borehole such as mud cake or washing-out. Typically, we expect the correlation between the sonic and density data to have a positive value because the higher the speed of the sonic data is, the higher the density of the rock. In this study, the correlation between the sonic and density logs is supposed to have a negative value because the sonic data have a unit of time over distance. Therefore, if a correlation coefficient presents a positive value, the data should be not used for training a prediction model.

In Figure 3(b), the data of the five blue coloured wells, numbers 4, 9, 19, 22, and 25, are removed from the well list because they showed strong positive correlation coefficients or curiously constant values. The wells 4, 9, and 25 among the excluded five wells in Figure 3(b) have more than +0.3 as correlation coefficient between the sonic and density logs. In addition, the wells 19 and 22 are deleted from the training data because of their flat values, without any trend.

In Case 1, from a total of 34 wells in Figure 3(a), five wells (the blue circles) are removed because of its poor

data quality and other five wells (in the black dotted area) in the Golden field are hypothetically considered as not existent (Figure 3(b)). Thus, the remaining 24 wells are used for further processing, and one well among the 24 wells is set as the test well (the red circle).

Case 2 is set for comparison with Case 1 so that the effect of well data at the target field could be examined. Case 2 utilises both the 24 wells of Case 1 and the five wells in the target field (Figure 3(c)). The test well in Case 1 belongs to a training set, and the five abnormal data are still not used. Case 2 assumes one well among the five wells at the target field as the test well. In Case 1, trainings are conducted to select the preferred training conditions (e.g., optimisation algorithm and the number of neurons in the hidden layers) and the hyperparameters from Case 1 are applied to Case 2.

A reliable neural network can be built by properly prepared training data. We already applied the two steps of preprocessing. After the selection of well data from a similar geologic depositional system to that of the Golden field, the five well data were entirely eliminated because their overall trends are not acceptable. In the case of the remaining 24 wells, partly wrong logs are removed and the remaining logs are utilised for training.

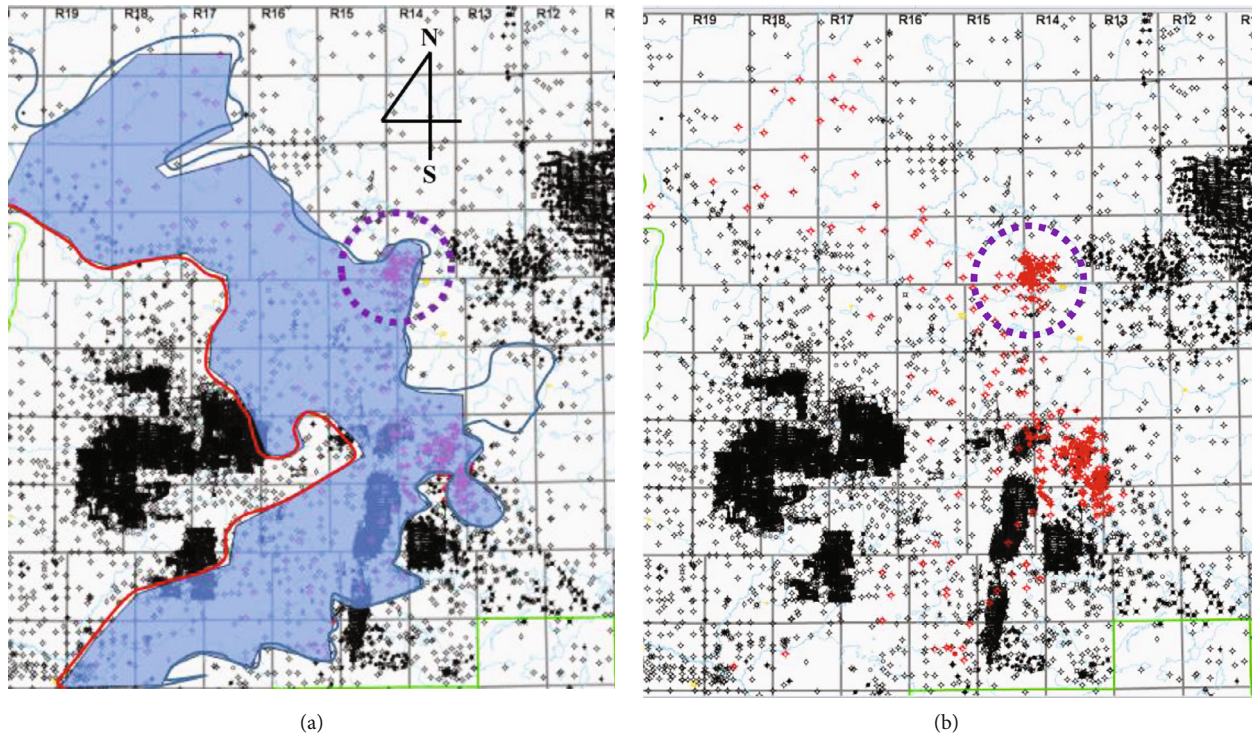


FIGURE 2: Filtered wells, in red colour, located in the SLVP Formation, Alberta, Canada, (a) with the geological frontier (solid red and blue lines) and the same depositional area (blue coloured highlight) and (b) without them. The purple circle indicates the Golden field.

Figure 4 presents examples of proper or improper log data. The  $x$ -axis means data depth and the  $y$ -axes at the left and the right denote the sonic and density data, respectively. Figures 4(a)–4(d) correspond to the wells 1, 6, 2, and 4 of Figure 3(a). Figures 4(a) and 4(b) are examples of the qualified data showing negative correlation between the sonic and density data without abnormal values. In Figure 4(c), the well 2 has partly not proper sonic and density logs. Both sonic and density data are acceptable from 1607 to 1620 m. However, after 1620 m, the sonic data have unreasonable constant values. Moreover, after approximately 1635 m, all the density data indicate  $-999$ , which means malfunction of the logging equipment or some problems. Those useless parts are eliminated for training. The correlation coefficient of the well 2 improves after the elimination ( $-0.0573$  to  $-0.2885$  in Figure 4(c)). In the case of Figure 4(d), because the overall trend of the correlation shows a high positive value,  $0.4861$ , the entire data of the well 4 are removed instead of partial treatment.

Figure 5 shows the effect of elimination of improper data for the 29 wells. Figure 5(a) presents the correlation coefficients of sonic and density data from each of the 29 wells before any data preprocessing. The correlation coefficients are separately calculated for each well. Figure 5(b) shows the changed distribution of correlation coefficients after the entire or partial elimination of abnormal values for each well. The mean of the correlation coefficient is improved in terms of the negative correlation between sonic and density data from  $-0.2647$  to  $-0.3913$ . For example, well 27 has a positive correlation coefficient (Figure 5(a), the red coloured frequencies). However, that

is because of a strange flat constant, not because of a wrong relationship between the sonic and density log data. After the elimination, all the data presenting positive correlation coefficients are removed (Figure 5(b)). The preprocessing for the data is critical for the overall training performance, and the difference in prediction performance depending on the data preprocess is presented in Results and Discussion.

For a single well, the log data are evenly spaced and each well has nearly two hundred data points. However, the number of data points is different depending on the wells because the interval of log data may differ for each well (e.g., 2.5, 10, 12.5, 15.2 cm). For Case 1, the training and validation data points are 8,684 from the 23 wells and the test data points are 219 for the test well (Figure 3(b)). Note that each data point consists of latitude, longitude, depth, sonic, and density data. Thus, the preprocessed training and validation dataset is a 5 by 8,684 matrix and the test dataset is a 5 by 219 matrix. For Case 2, the training and validation data points are more than 9,000 because Case 2 has more data from the additional wells. Compared to Case 1, Case 2 includes the test well of Case 1 and the four wells at the Golden field. In Case 2, the number of data points depends on which well, among the five wells at the Golden field, is classified into the test data.

One of the factors affecting the training performance of a neural network is normalisation of the given data. All the data, such as latitude, longitude, depth, sonic, and density, have different units and scales. The depth unit is m and the sonic and density log units are  $\mu\text{s}/\text{m}$  and  $\text{kg}/\text{m}^3$ , respectively. Table 1 presents statistical information of the training and

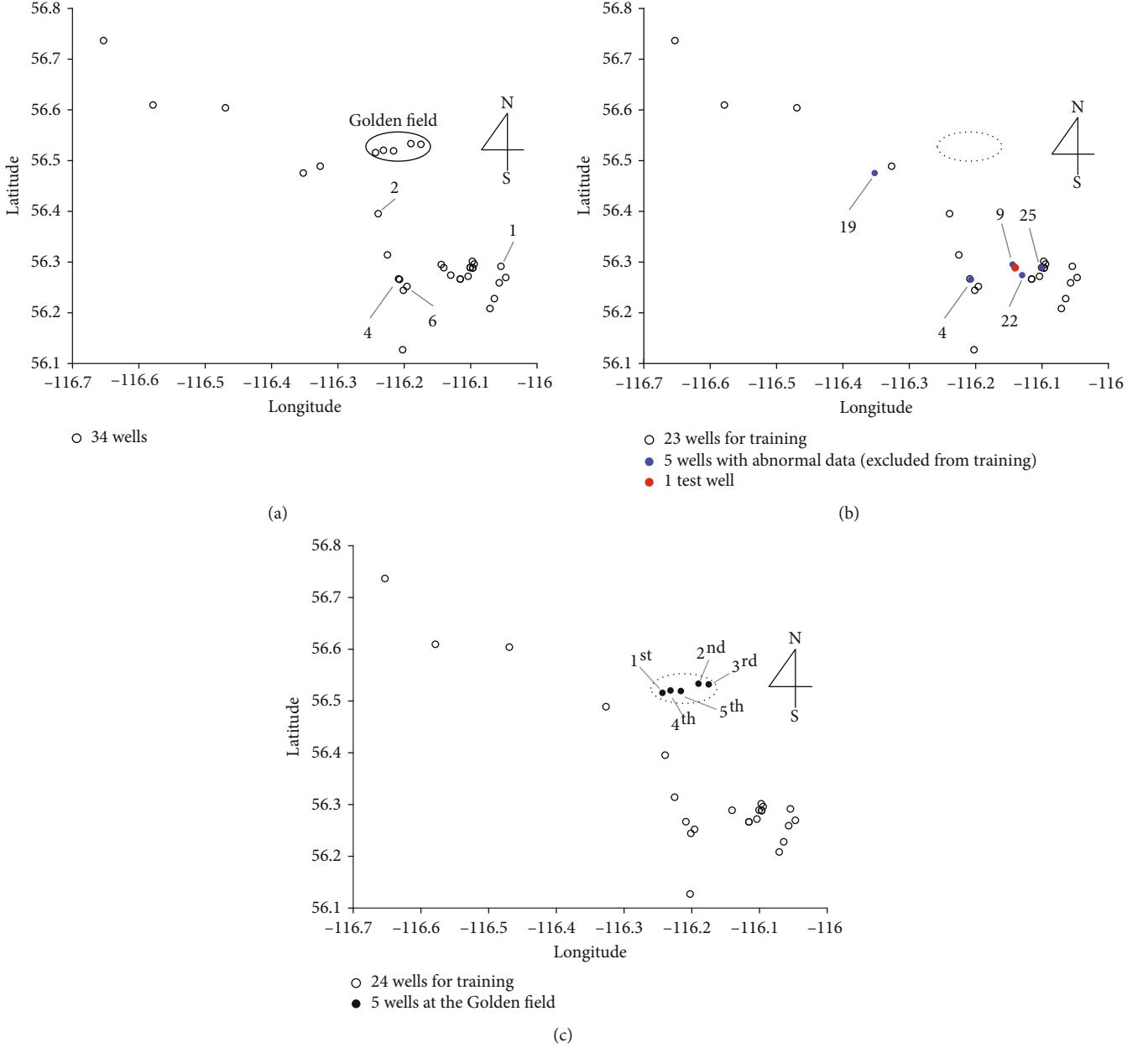


FIGURE 3: Location of wells in the same depositional environment. (a) Total wells, (b) 29 wells without 5 wells at the Golden field for Case 1, and (c) 29 wells including 5 wells at the Golden field for Case 2.

validation data. Those values in different units need to be normalised in a consistent way for proper training of a DNN model [16]. Min-max normalisation is usually applied for deep learning data because it can transform the given data into a range between 0 and 1 without exception [17, 18]. Each category of data is normalised by the following equation:

$$\text{Data}_{\text{norm}} = \frac{\text{Data} - \text{Data}_{\text{min}}}{\text{Data}_{\text{max}} - \text{Data}_{\text{min}}}, \quad (1)$$

where  $\text{Data}_{\text{norm}}$  is the normalised data between 0 and 1,  $\text{Data}$  are the raw data of each parameter (i.e., latitude, longitude, depth, sonic, and density data), and  $\text{Data}_{\text{max}}$  and  $\text{Data}_{\text{min}}$  are

the maximum and minimum values of the raw data, respectively. These preprocessed data are utilised for training of a DNN-based prediction model.

**2.2. Structure of Neural Network.** The structure of a neural network is definitely important for generation of pseudodensity data based on sonic data. Because there might be nonlinear relationship between the sonic and density log data, it is difficult to generate pseudodensity data from a single input feature (sonic data). In log data, there is information of 3D location: latitude, longitude, and depth, and this spatial information would help find the intrinsic relationship between the sonic and density data. Thus, the latitude, longitude, depth, and sonic data are applied to the input layer of a neural

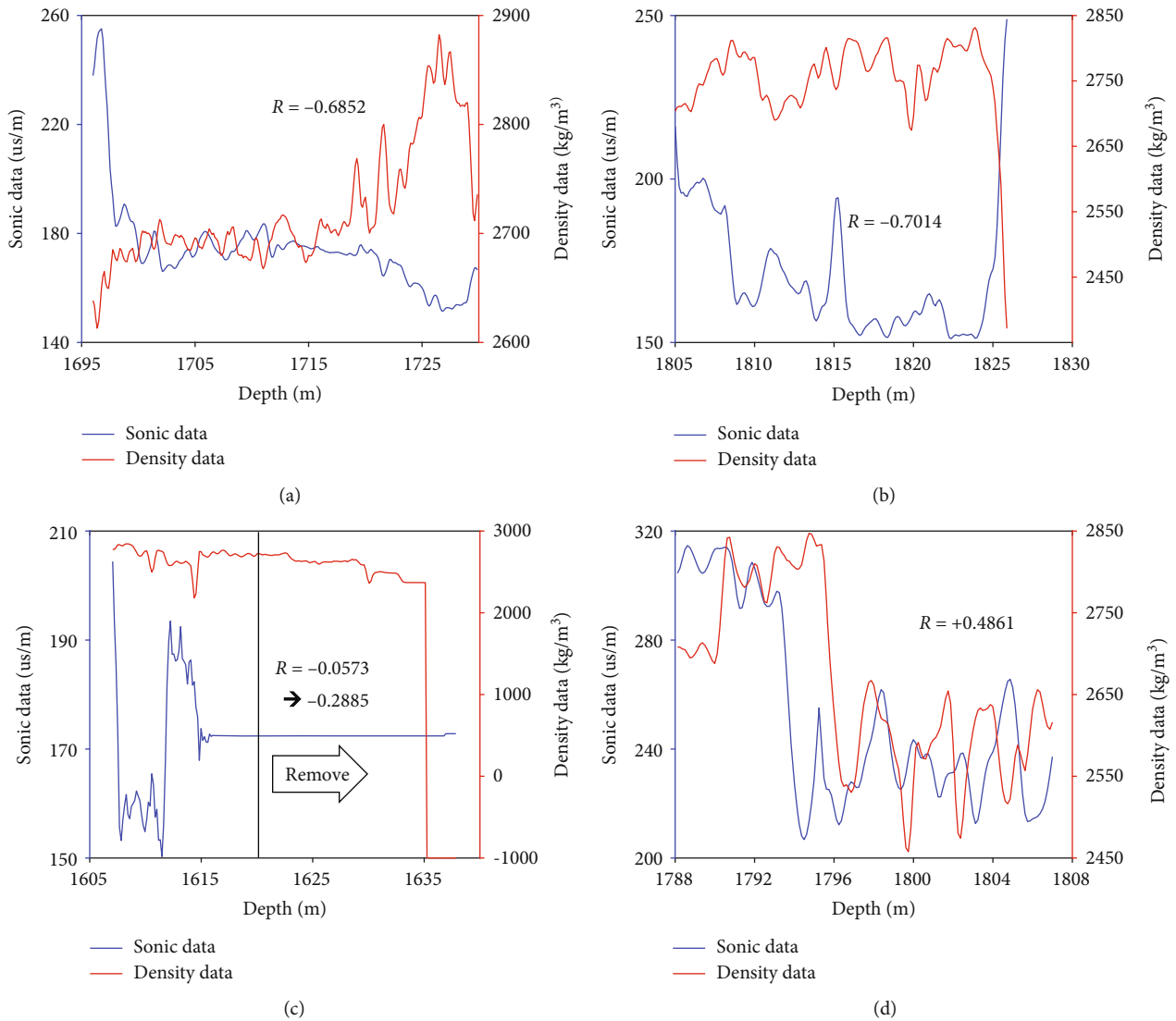


FIGURE 4: Examples of well- or poor-qualified sonic and density data. (a), (b), (c), and (d) correspond to wells 1, 6, 2, and 4 of Figure 3(a), respectively.

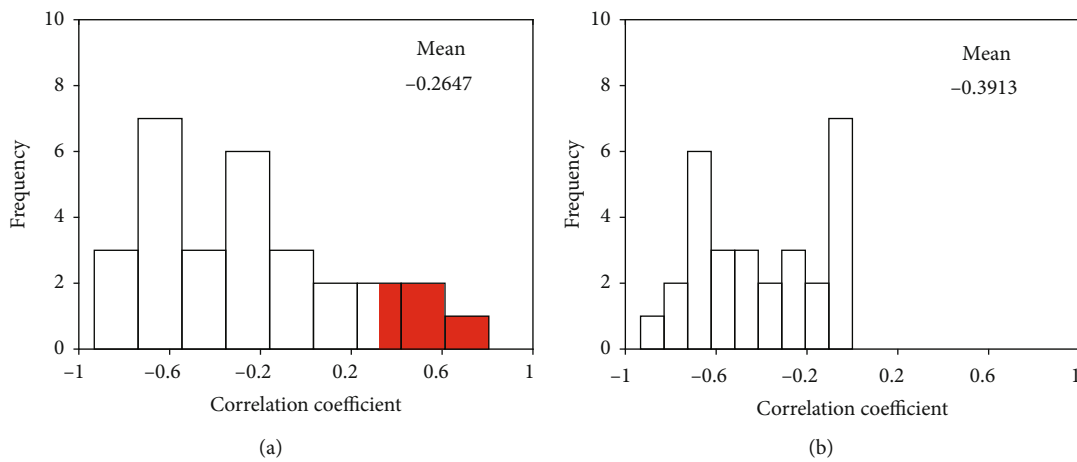


FIGURE 5: Correlation coefficients between the sonic and density data of (a) the 29 wells without the preprocessing and (b) the 24 wells with the preprocessing.

TABLE 1: Statistics of training and validation data.

	Latitude (degree)	Longitude (degree)	Depth (m)	Sonic ( $\mu\text{s/m}$ )	Density ( $\text{kg/m}^3$ )
Min	56.127	-116.65	1,595	134	2,079
Max	56.737	-116.05	1,893	298	2,885
Average	56.302	-116.17	1,720	186	2,695
Standard deviation	0.1397	0.1493	75	29	88

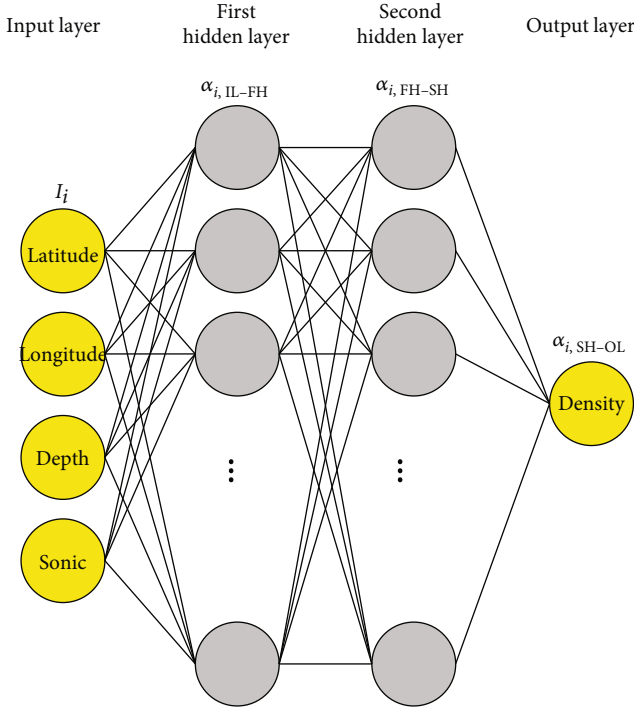


FIGURE 6: Structure of the DNN for generation of pseudodensity data from sonic data.

network and the density data are placed on the output layer. Figure 6 shows the neural network used in Case 1. The subscript  $i$ , IL, FH, SH, and OL refer to the  $i$ th training data, input layer, first hidden layer, second hidden layer, and output layer, respectively.

The number of hidden layers is set as two because two hidden layers are likely to be advantageous than one hidden layer for solving the complicated and nonlinear relation between the sonic and density data. However, three hidden layers are excessive as they increase the computational cost, and thus, result in inefficient performance. Each hidden layer has 10 nodes for the basic case.

**2.3. Selection of the Training Algorithm and Hidden Layer Nodes.** The process of training a neural network is, in practice, the updating of weights and biases between layers to obtain an optimised network. Usually, the default object function is the mean square error (MSE) between the outputs by the DNN and the target outputs (Equations (1) and (3)). In this study, the target outputs are the original density data

and they are compared with the synthetic density data from the DNN model.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (D_i - a_{i,\text{SH-OL}})^2, \quad (2)$$

$$a_{i,\text{SH-OL}} = f \left( \sum_{j=1}^{N_S} a_{i,\text{FH-SH}} \times W_{j,\text{SH-OL}} + \text{bias}_{\text{SH-OL}} \right), \quad (3)$$

where  $f$  is an activation function, MSE is the error between the target density data  $D_i$  and the predicted density data  $a_{i,\text{SH-OL}}$ , and  $n$  and  $N_S$  are the number of training data and the number of neurons in the second hidden layer.

To minimise the objective function MSE, weights and biases are updated by the training algorithm. This study is implemented with the deep learning package in MATLAB and it provides some training functions. We tested eight training functions, listed in Table 2, to find the appropriate one. The algorithms are divided into three categories: gradient descent, conjugate gradient, and quasi Newton. Gradient descent is a method to minimise an objective function by taking steps proportional to the negative of the gradient of the objective function [19]. The conjugate gradient method is the numerical solution algorithm of linear equation systems whose matrix is symmetric and positive-definite [20, 21]. Quasi Newton is an alternative method to the full Newton's method when its application is too cost-expensive and complicated [22].

In terms of the hidden layer's node, there could be infinite combinations theoretically. As mentioned in Section 2.2, the number of hidden layers is set as two with 10 nodes for each layer during determination of the best training algorithm. Then, the number of nodes in the hidden layers is analysed to find a proper combination of nodes.

**2.4. Cases 1 and 2.** Table 3 compares Cases 1 and 2 with regard to training and test data. In Case 1, the DNN is optimised using the 23 wells for training and it is verified by one well data for the test (Figure 3(b)). The test well (the well 29) is selected from the central part because it is spatially not biased. Compared to Case 1, Case 2 has data from the five additional wells, which belong to the Golden field (black dotted ellipse in Figure 3(c)). In Case 2, one of the five wells in the Golden field is set as the test well, and the remaining four are used as the training and validation data. According to the combinations of training and test data, five subordinate cases exist: Case 2-1, 2-2, 2-3, 2-4, and 2-5, listed in



TABLE 2: Eight training algorithm functions for optimizing a neural network.

Category	Code symbol in MATLAB	Training algorithm
Gradient descent	<i>traingd</i>	Gradient descent
	<i>traingdm</i>	Gradient descent with momentum backpropagation
	<i>trainrp</i>	Resilient backpropagation
Conjugate gradient	<i>trainscg</i>	Scaled conjugate gradient
	<i>traincgp</i>	Polak-Ribière conjugate gradient
	<i>traincgf</i>	Fletcher-Powell conjugate gradient
Quasi Newton	<i>trainbfg</i>	Broyden-Fletcher-Goldfarb-Shanno
	<i>trainlm</i>	Levenberg-Marquardt

TABLE 3: Comparison of Cases 1 and 2 in terms of training, validation, and test data.

Category	Case 1	Case 2
Training and validation data	23 wells	28 wells: 24 wells (the 23 training wells and 1 test well in Case 1 nearby the field + 4 wells among the 5 additional wells at the field (Case 2-1: 2 <sup>nd</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> , and 5 <sup>th</sup> ) (Case 2-2: 1 <sup>st</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> , and 5 <sup>th</sup> ) (Case 2-3: 1 <sup>st</sup> , 2 <sup>nd</sup> , 4 <sup>th</sup> , and 5 <sup>th</sup> ) (Case 2-4: 1 <sup>st</sup> , 2 <sup>nd</sup> , 3 <sup>rd</sup> , and 5 <sup>th</sup> ) (Case 2-5: 1 <sup>st</sup> , 2 <sup>nd</sup> , 3 <sup>rd</sup> , and 4 <sup>th</sup> )
Test data	1 well (no. 29)	The remaining 1 well among the 5 additional wells at the field (Case 2-1: 1 <sup>st</sup> ) (Case 2-2: 2 <sup>nd</sup> ) (Case 2-3: 3 <sup>rd</sup> ) (Case 2-4: 4 <sup>th</sup> ) (Case 2-5: 5 <sup>th</sup> )

Table 3. Note that the five additional wells (1<sup>st</sup> to 5<sup>th</sup>) at the Golden field are presented in Figure 3(c).

### 3. Results and Discussion

**3.1. Case 1.** As mentioned in Section 2.1, the preprocessed data are applied to the default neural network: two hidden layers with the basic 10-10 node combination. The structure of the default neural network is schematically presented in Figure 7. More detailed training options are summarised in Table 4. In Section 3.1.1., the default training condition of the neural network is decided first. Then, a sensitivity analysis of the training algorithms is conducted using the default settings because it is important to efficiently find a training condition to show a trustworthy training performance considering a training cost. Moreover, based on the default condition of the neural network, the performances of the trained neural networks before and after the preprocess of data are analysed to verify the effectiveness and necessity of the preprocess. In Section 3.1.2., another sensitivity analysis is performed to determine the number of nodes in the hidden layers. A hierarchical analysis is used to find the best combination of the number of neurons in the hidden layers.

**3.1.1. Sensitivity Analysis for Training Algorithm.** There are two aspects that we considered to decide whether a training is properly designed and well performed: errors with the validation data and test data. In Case 1, the training and validation data are randomly selected among 8,684 data points with the assigned ratios of 0.85 and 0.15 (Table 4), although the well 29 is fixed as the test well data (219 data points). The validation and test data ratio over the training data is about 18%. Figure 8 presents two examples of the dataset representing randomness about the selection of training and validation data. The  $x$ - and  $y$ -axes are the normalised scale of the sonic and density logs, respectively. Figures 8(a) and 8(b) show two randomly selected training and validation data, and Figures 8(c) and 8(d) depict the zoomed-in area of the black dotted rectangle in Figures 8(a) and 8(b), respectively. The red circles mean the validation data that are differently chosen in each case.

Even if the structure and hyperparameter of a specific neural network are the same, it might give somewhat different trained results owing to the random selection of the training and validation datasets. We build several prediction models to mitigate the effect of the random selection. Based on the central limit theorem, a total of 30 prediction models are created and their average is regarded as the performance

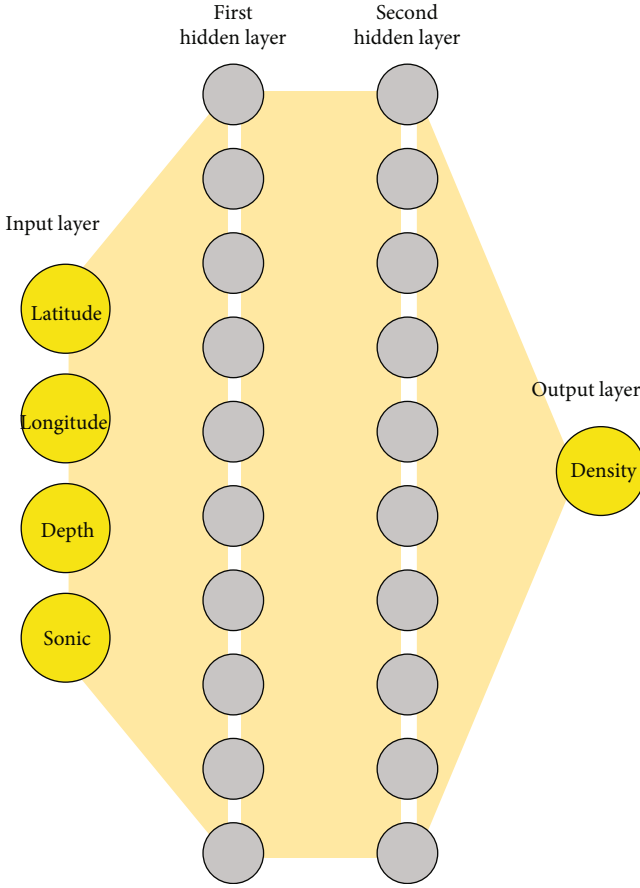


FIGURE 7: Default condition of the DNN for the generation of pseudodensity data using spatial information and sonic data. Network lines are presented alternatively by the highlighted area.

TABLE 4: Training options for the DNN.

Parameter	Condition
Training algorithm	One of the algorithms from Table 1
Activation function	Log-sigmoid (for both the first and second hidden layers)
Maximum number of epochs	1,000
Maximum number of validation failures	10
Ratio of training and validation data	85%, 15% (random selection)

of the trained DNN in this study. In other words, we predict the 30 synthetic density logs for the same input sonic log according to each prediction model.

In terms of the validation and test data, two errors are compared in the eight training algorithm functions. Figure 9 shows the MSEs of the validation data in each training algorithm in a bar chart. The bar of each algorithm means the average error of 30 pseudodensity logs from randomly selected training and validation datasets under the same training conditions. In Figures 9(a) and 9(b), both validation

and test errors are consistently calculated with the preprocessed values by Equation (1). These results are normalised in the range between 0 and 1 to fairly reflect both the validation and test errors (Figures 9(c) and 9(d)).

Some training algorithms such as *trainbfg*, *trainrp*, and *trainscg* (Table 2) show decent performances in the validation error compared to *trainlm* (Table 2). However, there is a difficulty in having a consistent result for *trainlm* for both the validation and test errors. Figure 9(e) is the sum of Figures 9(c) and 9(d) to provide a general comparison of the errors of the eight algorithms. The best training algorithm is *trainrp*, because it showed not only a fast but also a stable performance in both cases of errors. Therefore, *trainrp* is selected as the default algorithm and it is used for a sensitivity analysis of the number of neurons in the hidden layers in Section 3.1.2.

Among the eight algorithms, the reconstruction performances for the test data of four representative algorithms are compared in Figure 10. They are chosen from each category of training algorithms: *traingdm* and *trainrp* of gradient descent, *trainscg* of conjugate gradient, and *trainlm* of quasi Newton (Table 2). In Figure 10, the blue lines indicate the mean of 30 synthetic density logs from 30 trained neural networks, and the red lines are the true density log of the test well data. Thus, the closer the match between the blue and red lines is, the better the prediction performance of the training algorithm. The two pictures in the first row are good matching examples (Figures 10(a) and 10(b)). In contrast, the two results in the second row are poor matching examples (Figures 10(c) and 10(d)). In Figure 10(a), *trainscg* gives an average line following the test well trend. In spite of some discrepancy between the test and the reconstruction, the overall trends matched with each other. In Figure 10(b), *trainrp* gives a good matching performance with the test line, as good as that in Figure 10(a), although the middle part between 50 to 150 has an almost flattened trend. In Figure 10(c), compared to the remaining three training algorithms, the average by *trainlm* does not follow the trend of the reference well data. It results in the high error in Figure 9(d). In Figure 10(d), *traingdm* has a flattened prediction and it does not represent the pattern of the real density data.

In Section 2.1, the importance of a proper preprocess for the density and sonic log data is highlighted. The results of Figure 10 are from preprocessed training data and Figure 11 shows the prediction results without a proper preprocess. In Figures 11(a) and 11(b), the trend of the blue lines is similar to those in Figures 10(a) and 10(b). However, the gap between the blue and red lines becomes larger because the deviations of the blue lines decreased in both *trainscg* and *trainrp* (Figures 11(a) and 11(b)). *trainlm* still presents a large discrepancy between the test data and the reconstructed density log (Figure 11(c)). Figure 11(d) highlights how improper training data affect the training performance. It seems that poor training data cause flattening in the estimations of the test data because faulty data make it difficult to find the essential intrinsic relationship between the input and output data.

Although the results of the four algorithms seem to have a similar pattern in Figures 10 and 11, Table 5 quantitatively

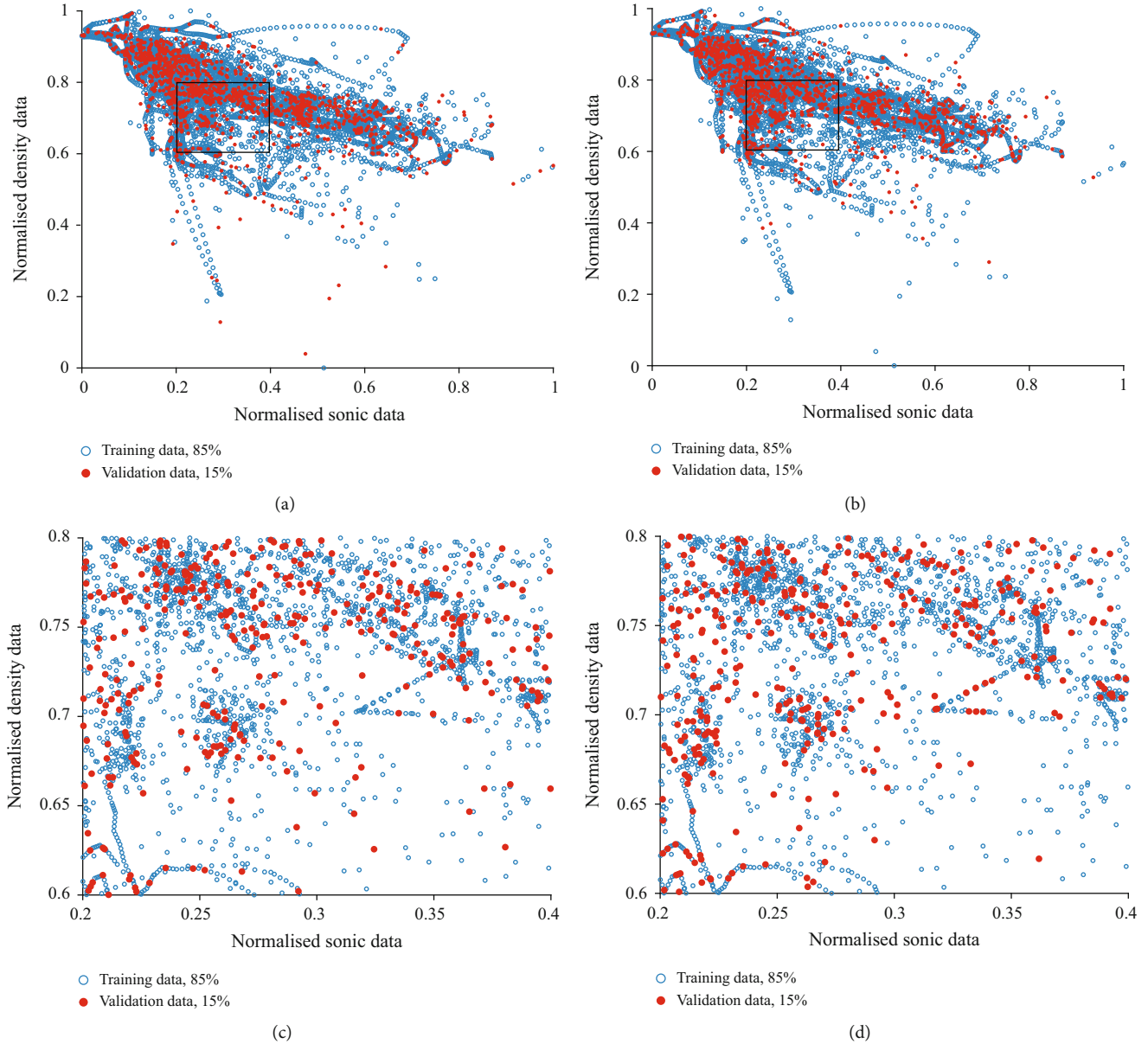


FIGURE 8: Randomness of data selection for training and validation in (a, b) two examples and (c, d) their zoomed-in pictures.

presents the difference in performance according to data preprocess. The discrepancy between the true and predicted density data is calculated with the mean of the RMSE in the following equation:

$$\text{Mean RMSE} = \frac{1}{n} \sum_{j=1}^n \sqrt{\frac{1}{m} \sum_{i=1}^m (\text{Test}_i - \text{Recon}_{i,j})^2}, \quad (4)$$

where Test means the density log of the test well and Recon denotes the reconstruction values corresponding to Test. The subscripts  $i$  and  $j$  indicate the  $i$ th data point and  $j$ th trained model, respectively.  $m$  is the number of data point of the test well, and  $n$  is the number of the trained neural network. In this study,  $m$  and  $n$  are 219 and 30, respectively.

The data preprocess results in decreased errors for the all four training algorithms. *trainlm* shows an unreliable training performance regardless of the data preprocess and the results of *traingdm* are sensitive to the preprocess of data. The overall reduced errors in Figure 10 verify the necessity of proper preprocessing for qualified training data. These results are in agreement with the results of previous studies, which mentioned the importance of data processing [4, 6].

**3.1.2. Sensitivity Analysis for the Number of Neurons in Hidden Layers.** Another sensitivity analysis is implemented for the combination of number of neurons in the first and second hidden layers. Because there are infinite combinations, we use a hierarchical approach to determine the number of neurons efficiently. First, we assume that the two hidden layers have the same number of neurons. After the

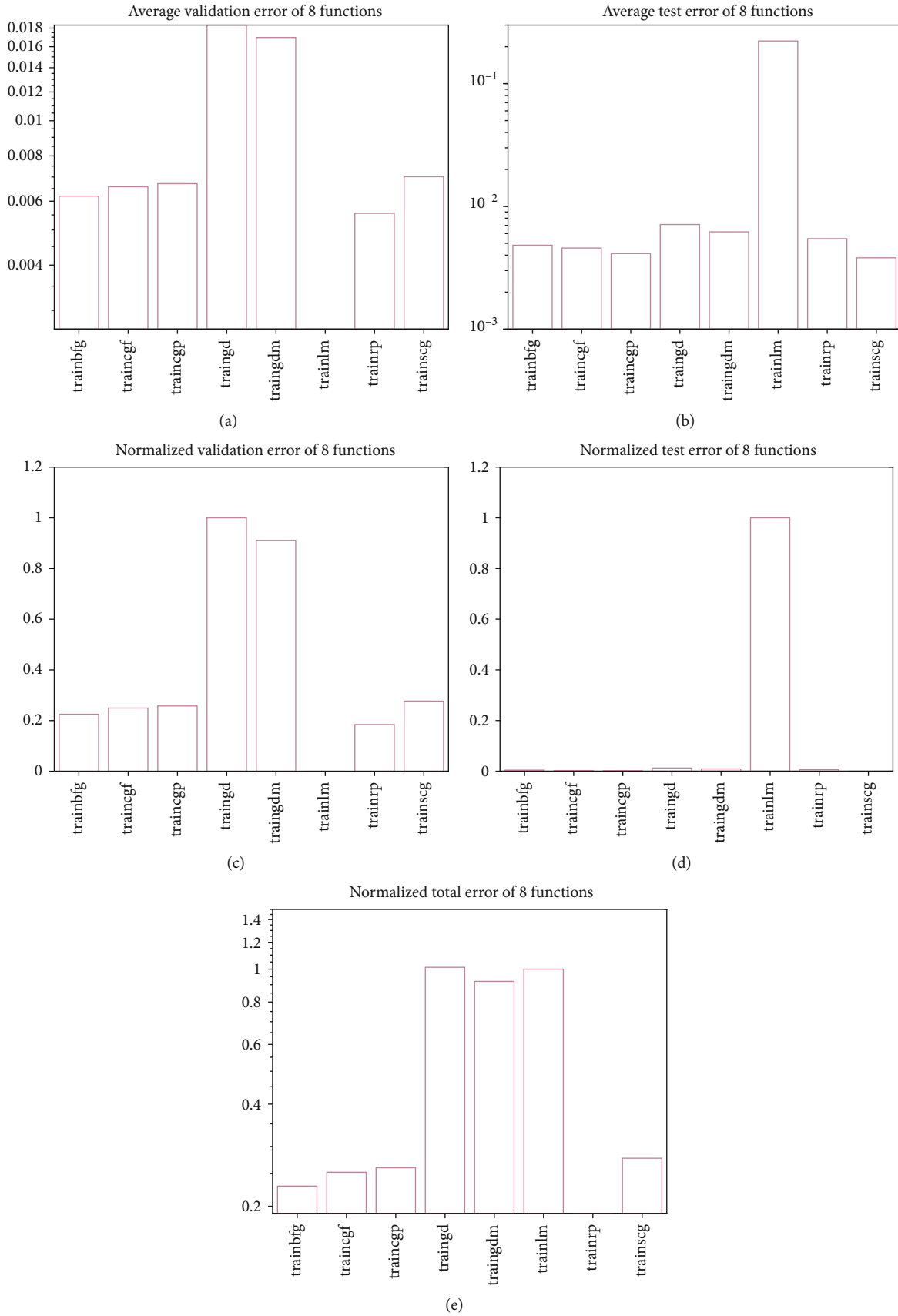


FIGURE 9: Comparison of errors with the validation and test well data. (a) Errors between validation data and reconstructed data, (b) errors between test well data and its reconstruction, (c) normalised errors of (a), (d) normalised errors of (b), and (e) sum of the two normalised errors.

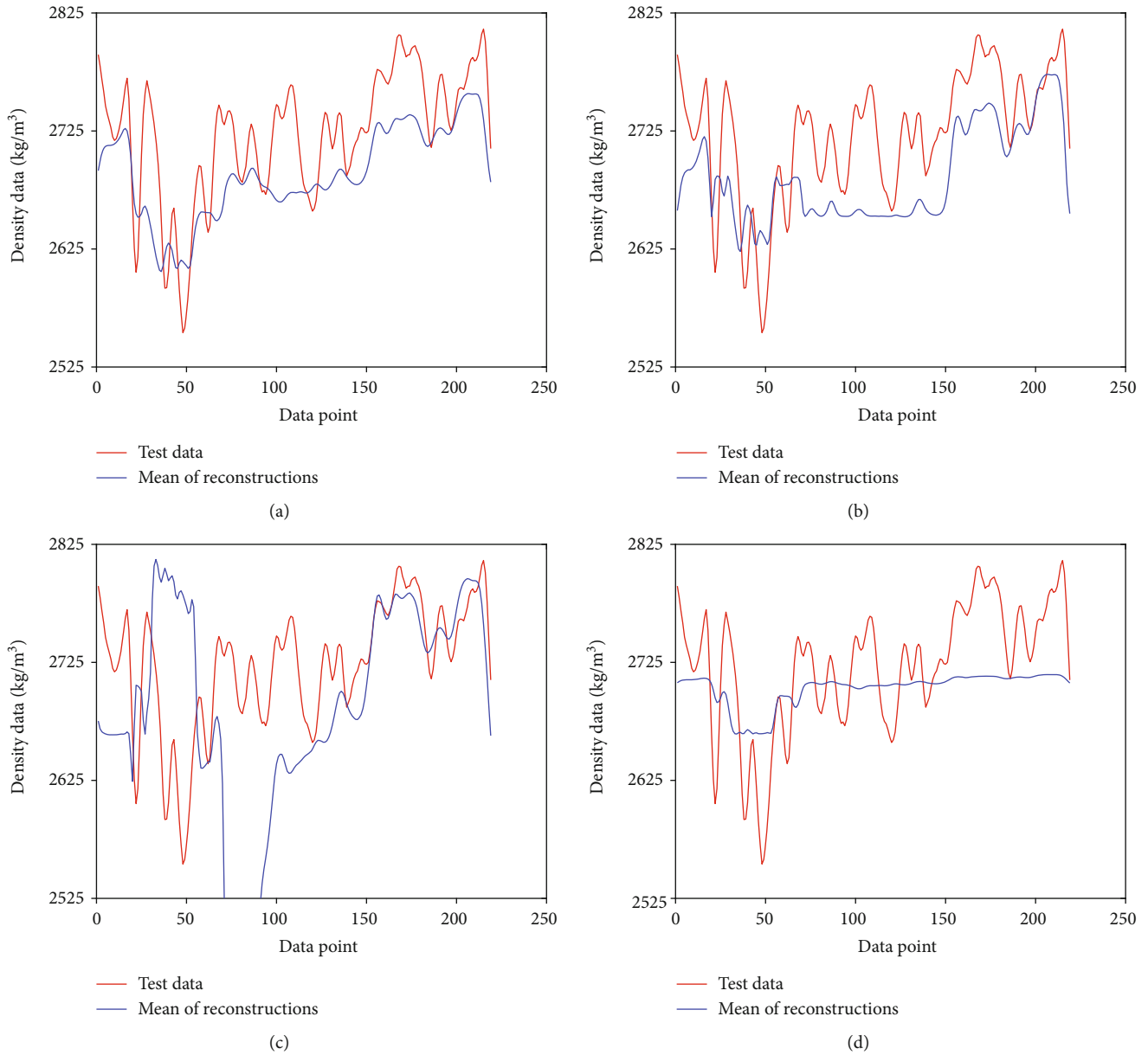


FIGURE 10: Comparison of the test well data and reconstructed data using four training algorithm functions. (a) *trainscg*, (b) *trainrp*, (c) *trainlm*, and (d) *traingdm*.

eight combination cases, 10-10, 20-20, 30-30, 40-40, 50-50, 60-60, 70-70, and 80-80, are compared by both validation and test errors, a preferred range of the number of neurons is determined approximately. Second, to find the best case, we set 200 combinations by varying the two parameters within the preferred range independently. Note that the train algorithm is fixed as *trainrp*.

Figure 12 shows the error results of the eight combinations. Here, one error value for each combination means an average error of 30 synthetic density data. Figures 12(a) and 12(b) are the validation and test errors, respectively. In Figure 12(a), the validation error consistently decreases as the number of nodes of the hidden layers increases. A large number of nodes have advantage in the validation data error. However, the test error

moves up and down as the number of nodes in the hidden layers changes (Figure 12(b)). The lowest test error is shown in the combination of 30-30 hidden layer nodes. It seems that the behaviours of the validation and test errors are different because the test dataset does not have exactly the same distribution of the training or validation data. This problem can be solved if sufficient data are available.

In spite of that discrepancy of trend in the validation and test errors, we should make a compromise between the two errors to decide an appropriate combination of hidden layer nodes. Thus, we calculate the total error of the two normalised errors (Figure 12(e)) after the validation and test errors are normalised, as shown in Figures 12(c) and 12(d). Consequently, there would be a

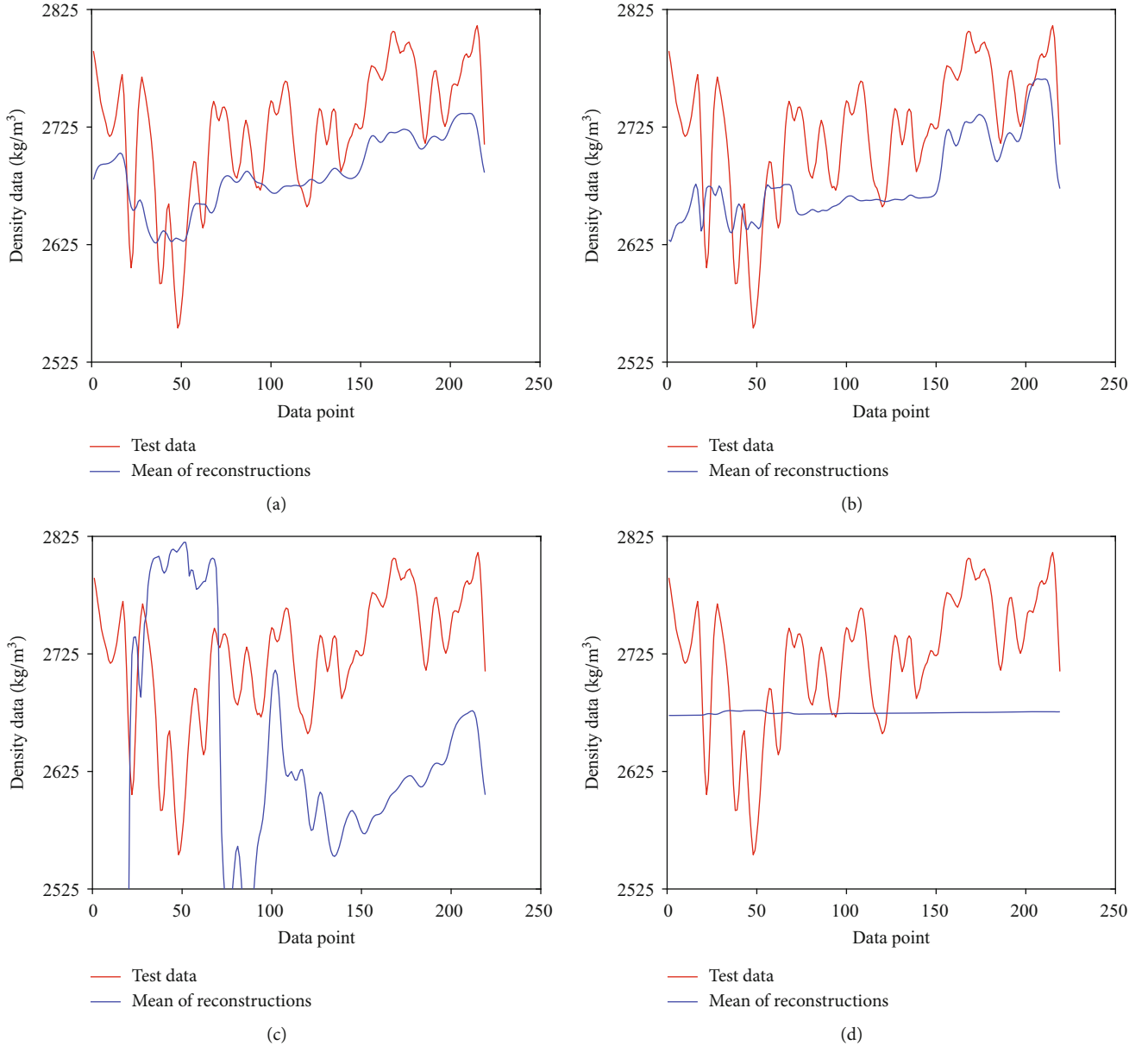


FIGURE 11: Comparison of the test well data and reconstructed data by four training algorithm functions without the proper preprocess of the training data. (a) *trainscg*, (b) *trainrp*, (c) *trainlm*, and (d) *traingdm*.

TABLE 5: Comparison of test well data error (RMSE) by data preprocess.

Training algorithm	Preprocess (Figure 10)	No preprocess (Figure 11)	Increased error (%)
<i>trainscg</i>	49	54	10
<i>trainrp</i>	58	62	7
<i>trainlm</i>	288	316	10
<i>traingdm</i>	60	76	27

proper combination of hidden layer nodes around the 30-30 case. From the eight combinations at the first level of the hierarchical approach, we can make a reasonable combination of neurons for further analysis.

We randomly generate 200 combinations by changing the number of neurons for the two hidden layers from 5 to 45 because the 30-30 combination is preferred in Figure 12. Figure 13 shows the 200 combinations of the first and second hidden layer nodes. The  $x$  and  $y$ -axes mean the number of nodes in the first and second hidden layers, respectively. Each circle represents one combination. The red and blue circles are the best and worst 20 combinations, respectively. The performance of the 200 combinations is estimated in the same way as that in Figures 9(e) and 12(e), which is the sum of the normalised validation and test errors. The blue and red circles are obviously separated into the left and right sides. It is revealed that, first, a large number of nodes are needed in the first hidden layer to achieve a high performance. Second, compared to the first hidden layer, the

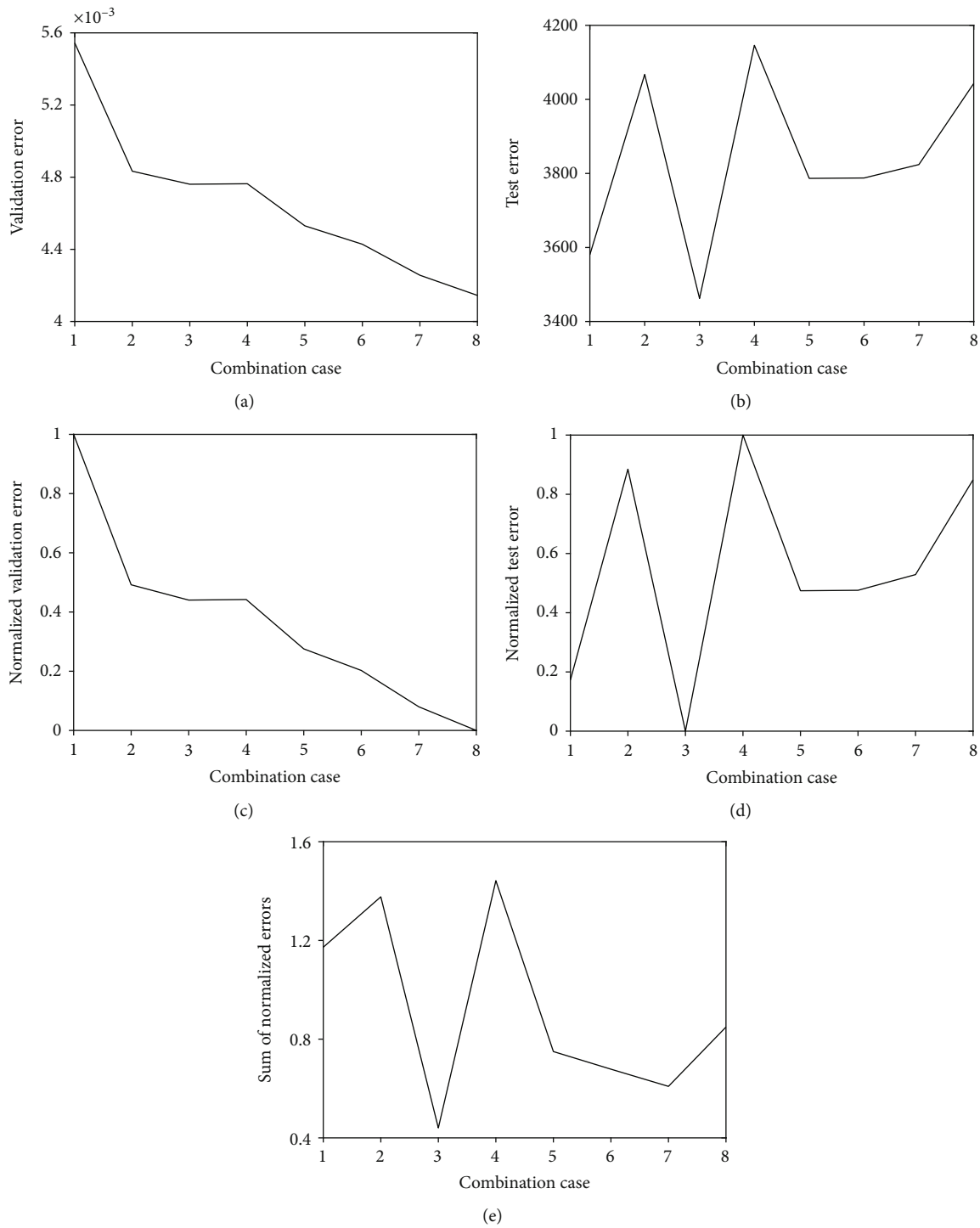


FIGURE 12: Error estimation of validation and test data depending on combination of hidden layer nodes (first-second hidden layer: 10-10, 20-20, 30-30, 40-40, 50-50, 60-60, 70-70, and 80-80). (a) Validation error, (b) test error, (c) normalised validation error, (d) normalised test error, and (e) sum of two normalised errors.

overall performance by the second hidden layer is not sensitive to the number of nodes.

The best combination among the 200 combinations is the 45-14 case, which is marked with a red dotted circle in Figure 13 and the worst combination, the 5-33 case, is marked with a blue dotted circle. After the best and worst combina-

tions are trained with the 23 wells in Case 1, the two prediction models are applied to the test well in Case 1 (Figure 14) and the additional five test wells in the Golden field (Figure 15). As aforementioned, although the five additional wells are supposed to be used in Case 2 (Figure 3(c)), they are used for a comparison of the best and worst combina-

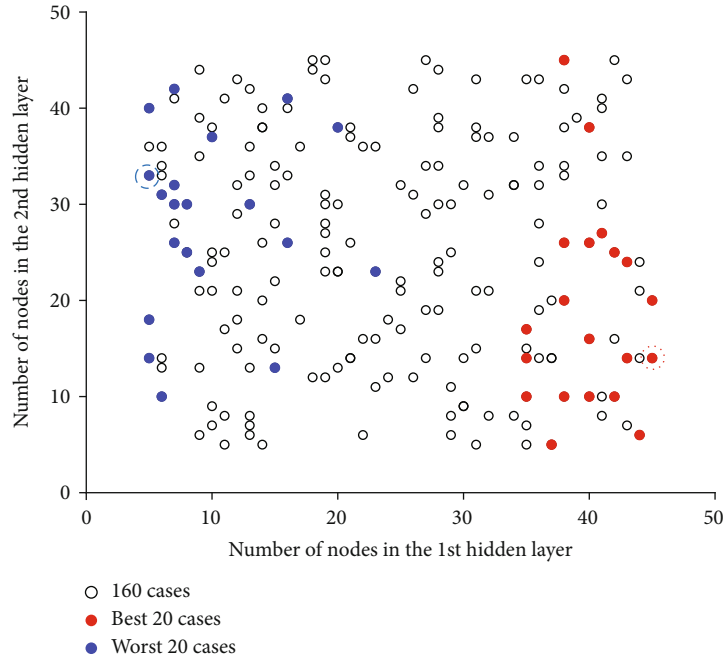


FIGURE 13: Distribution of the 200 random combinations for the number of hidden layer nodes.

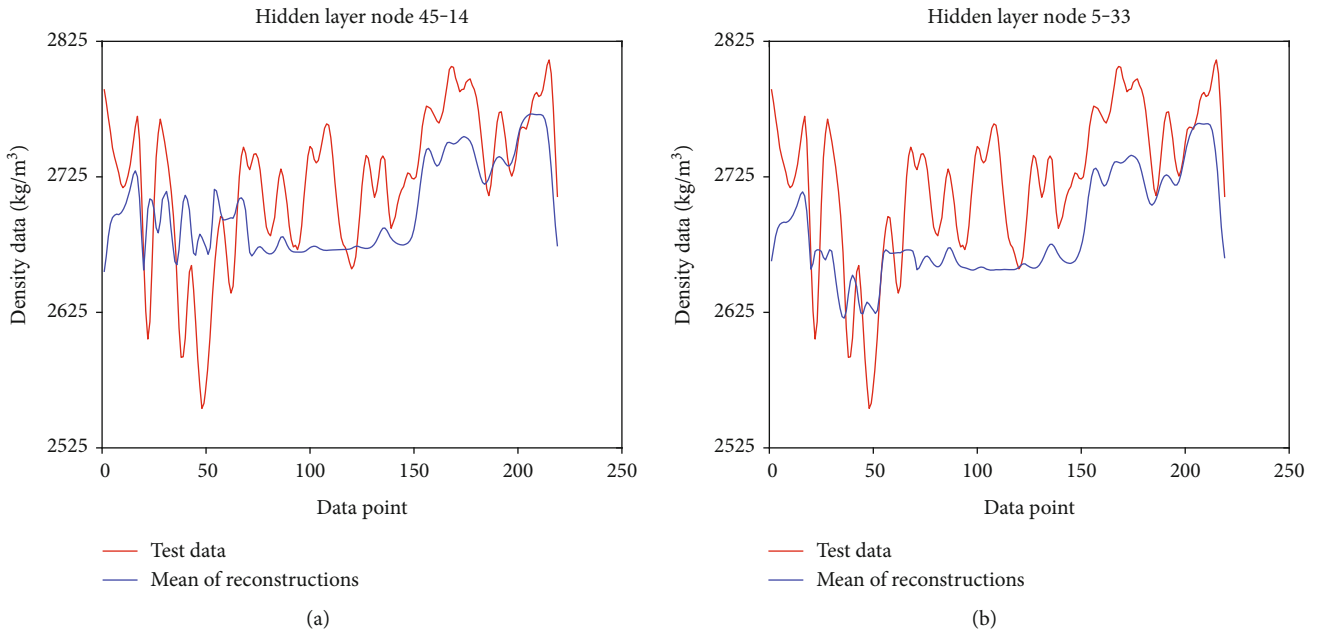


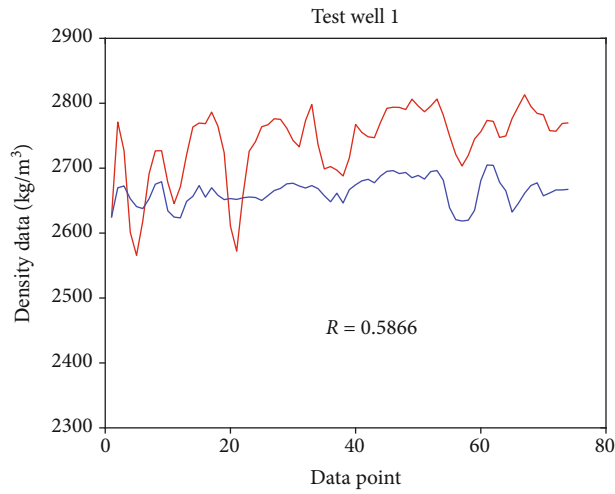
FIGURE 14: Comparison of reconstructions for the test well data in Case 1 by (a) the best and (b) the worst combination of nodes in the hidden layers.

Figures 14(a) and 14(b), pseudodensity data are generated by the DNNs, which have the best and worst combinations of hidden layer nodes. Compared to the 5-33 combination, the average of the reconstructions by the 45-14 combination gives a better matching with the trend of the test data.

The difference in performance of the best and worst combinations appears more clearly in the additional five test well data (Figure 15). In both the best and worst combinations,

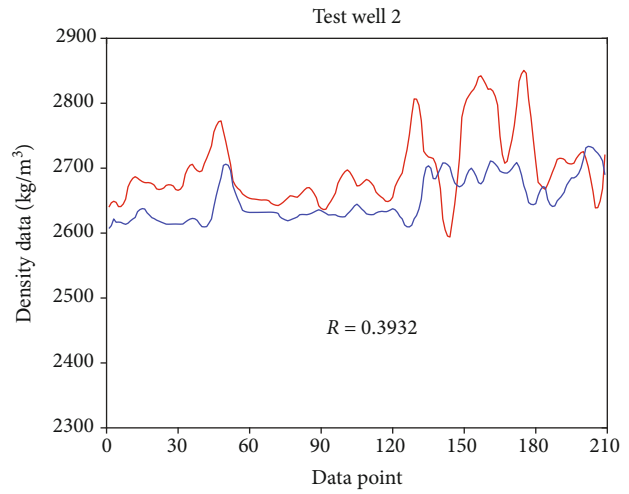
there must be a large uncertainty due to the limited well log data for the additional five test wells (Figure 3(c)). Even though the best combination is trained with limited information, its averages (the blue lines in Figures 15(a) to 15(e)) tend to follow the overall trend of the test wells. In contrast, the worst one presents large discrepancies between the blue and red lines (Figures 15(f) to 15(j)). Although the blue lines seem to mimic the pattern of the test data, they underestimate the





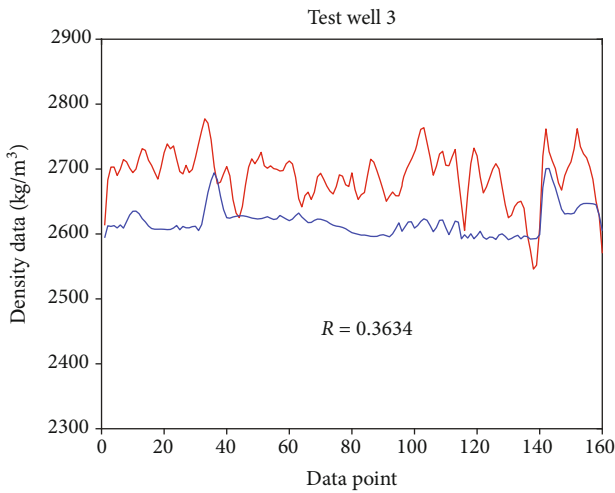
— Test data  
— Mean of reconstructions

(a)



— Test data  
— Mean of reconstructions

(b)



— Test data  
— Mean of reconstructions

(c)



— Test data  
— Mean of reconstructions

(d)

FIGURE 15: Continued.

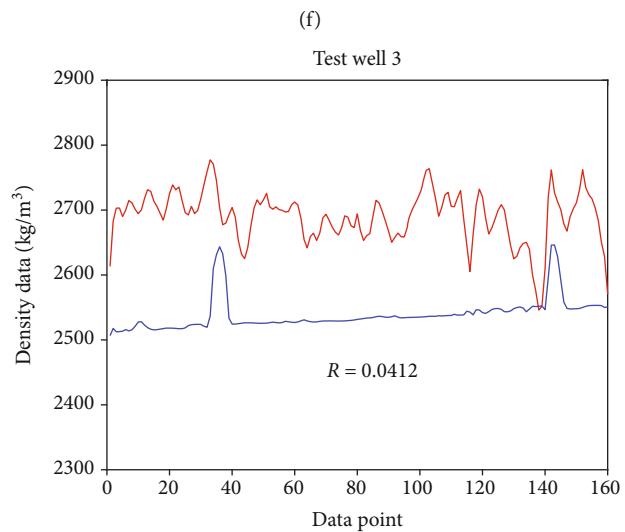
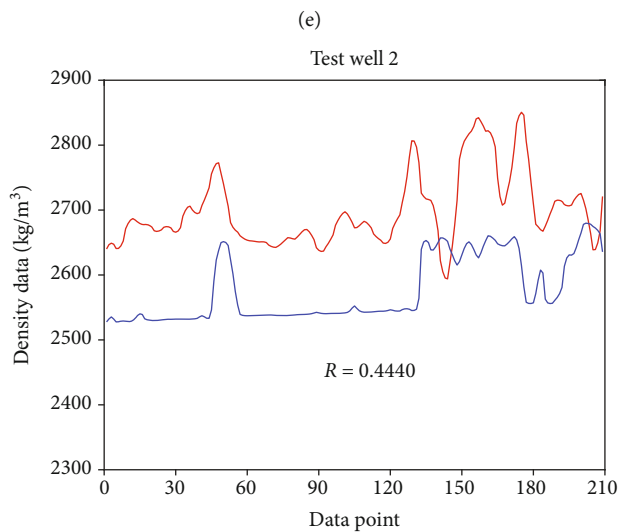
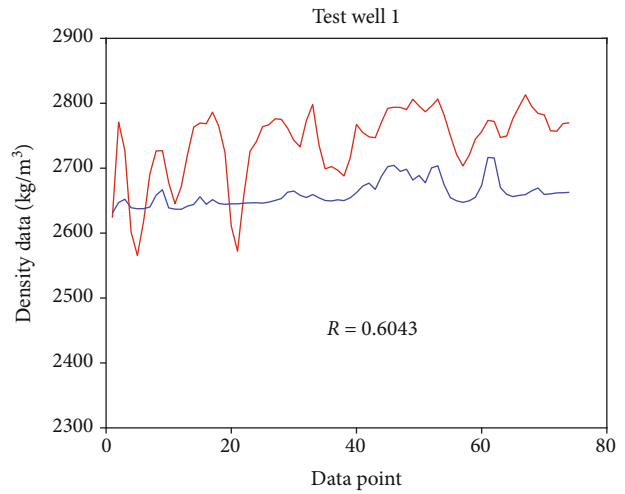
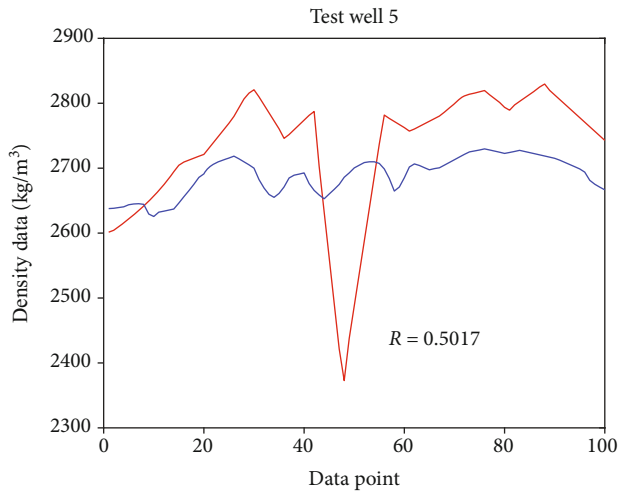


FIGURE 15: Continued.

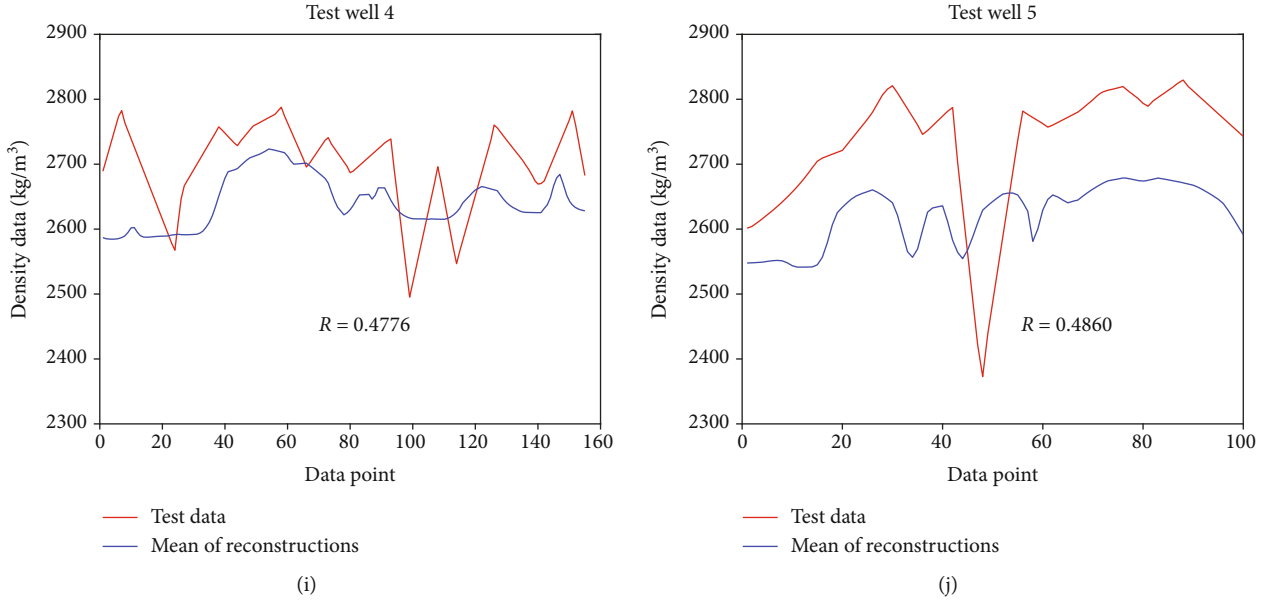


FIGURE 15: Comparison of reconstructions for a test well among the five additional wells at the Golden field by (a)–(e) the best and (f)–(j) the worst combination of nodes in the the hidden layers.

TABLE 6: Comparison of test data error (RMSE) in Cases 1 and 2.

Test well	Case 1 (the best)	Case 1 (the worst)	Case 2
The 1 <sup>st</sup> well	119	129	68
The 2 <sup>nd</sup> well	121	260	57
The 3 <sup>rd</sup> well	129	237	48
The 4 <sup>th</sup> well	107	172	61
The 5 <sup>th</sup> well	139	292	88
Average	123	218	64

density data of the test wells and the degree of underestimation is worse than that of the best combination. Table 6 compares the performance of the best and worst cases with the RMSE. Each RMSE indicates the mean of the reconstruction results from 30 trained networks (Equation (4)). The worst case has generally two times larger RMSE compared to that of the best case. The larger discrepancy with the test data mostly results from the underestimation. Except for the test well 1, the worst case for the rest of the four wells shows worse underestimation than the best (Figure 15).

3.2. Case 2. In both the best and worst combinations of neurons in the hidden layers, the underestimation problem of the pseudodensity data compared to the actual test data occurs (Figure 15). In [4], the authors mentioned that the prediction of performance of the wells located in the middle is better because a neural network may interpolate the information of adjacent wells. Also, Zhang et al. stressed that the prediction performance can be significantly improved by the information in the target field [1]. Therefore, to solve the problem, Case 2 uses the 5 additional

wells in the Golden field to train and test the best combination case, as presented in Table 3 and Figure 3(c). Figure 16 reveals that Case 2 has five subordinate cases: 2-1, 2-2, 2-3, 2-4, and 2-5. For example, Case 2-2 has one test well (number 2 among the five wells) and the remaining four well data (numbers 1, 3, 4, 5) as training data (the upper graph in Figure 16(b)). It is expected that Case 2 would bring better test performance compared to Case 1 because of the additional amount of training data from the target field. Note that the test well in Case 1 also belongs to training data in Case 2.

In Figures 16(a)–16(e), the graphs in the first row are the synthetic density logs for each test well. They should be compared with Figures 15(a)–15(e) to analyse the effect of additional training data. The pictures in the second row are the location of the wells. In case of Figure 15, only the 23 training wells are used, and they are separated to some extent from the interested target field. The 23 wells are usually located in the longitude  $-116.1^\circ$  and the latitude  $56.3^\circ$ , although the Golden field including the five wells is positioned in the longitude  $-116.2^\circ$  and the latitude  $56.5^\circ$ . Therefore, it is a difficult to properly predict the density logs for the five test wells in Figure 15 without the geologically and spatially related data.

The prediction of the test data in Figure 16 shows an improved performance compared to the prediction in Figure 15. No matter which well is set as the test data among the additional five wells, they show better results compared to those in Figures 15(a)–15(e). Especially, in Figures 16(b) and 16(c), the underestimation problem in Case 1 is mitigated compared to that in Figures 15(b) and 15(c). Figure 16(d) shows better matching with the fluctuating trend of the true log. Especially, it properly follows most peaks of the test data. For a quantitative comparison of Cases 1 and 2, the RMSEs of the test data for each well are calculated (the last column in

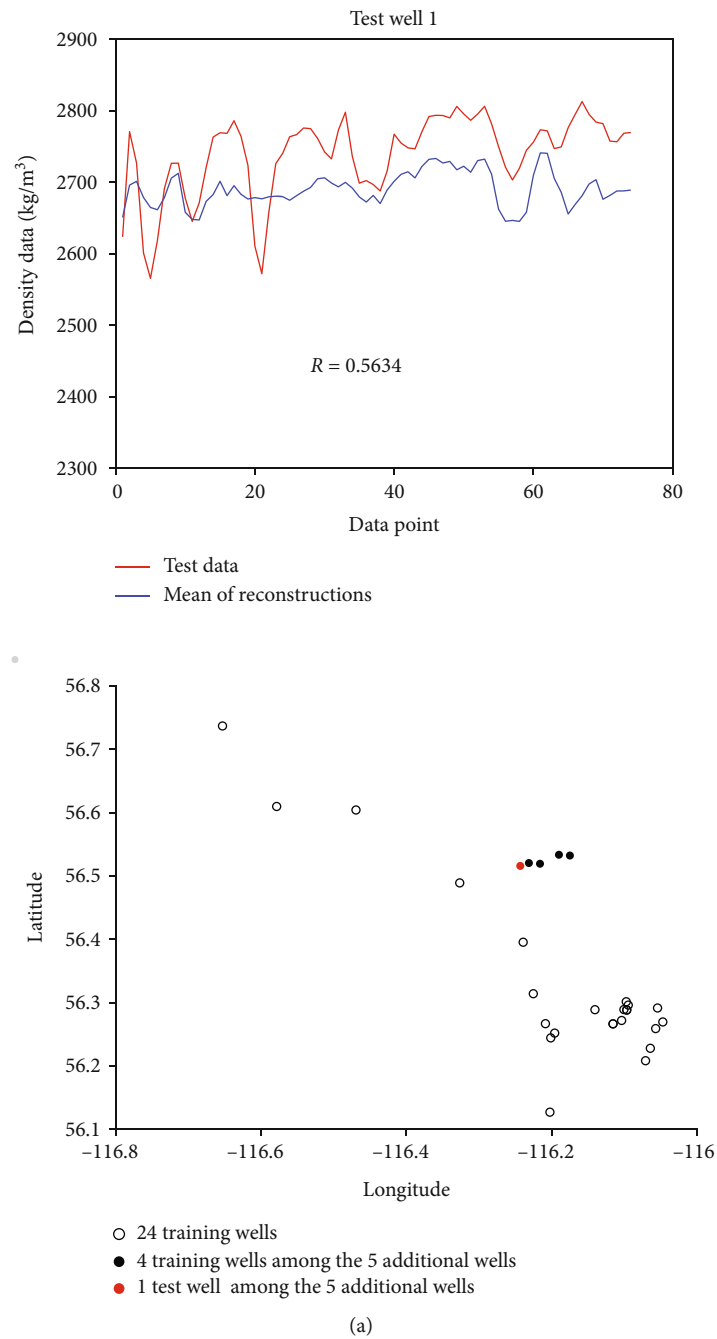
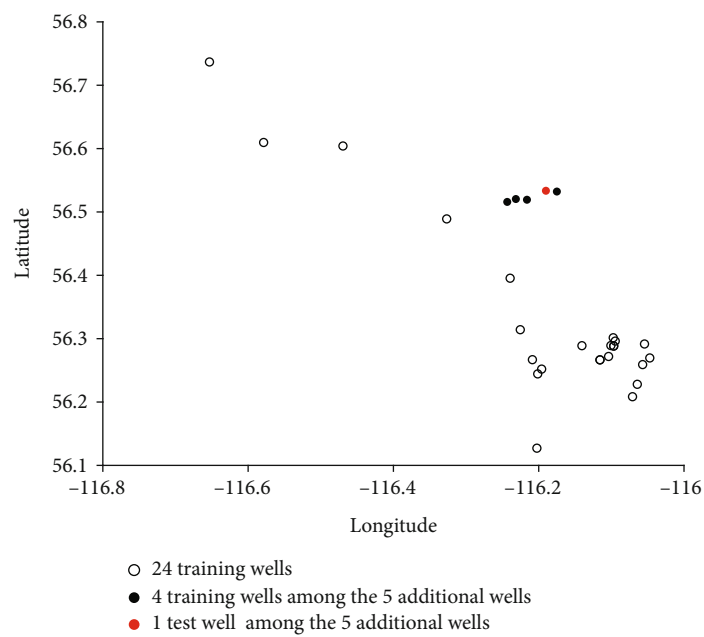
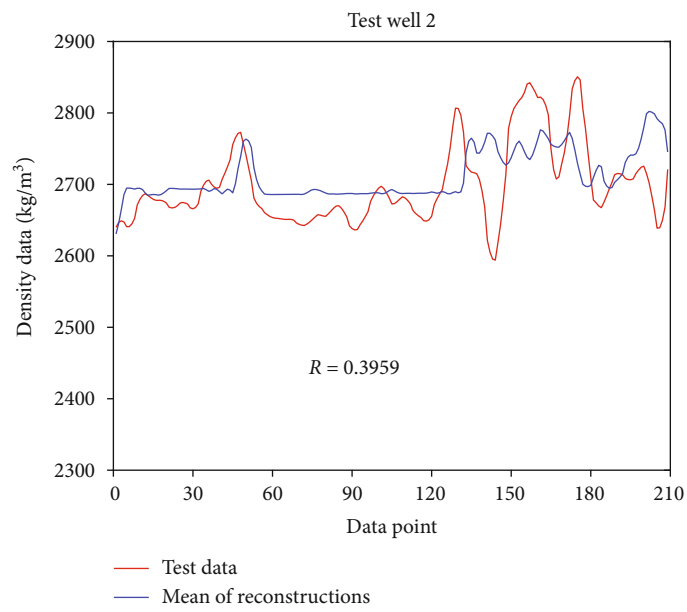
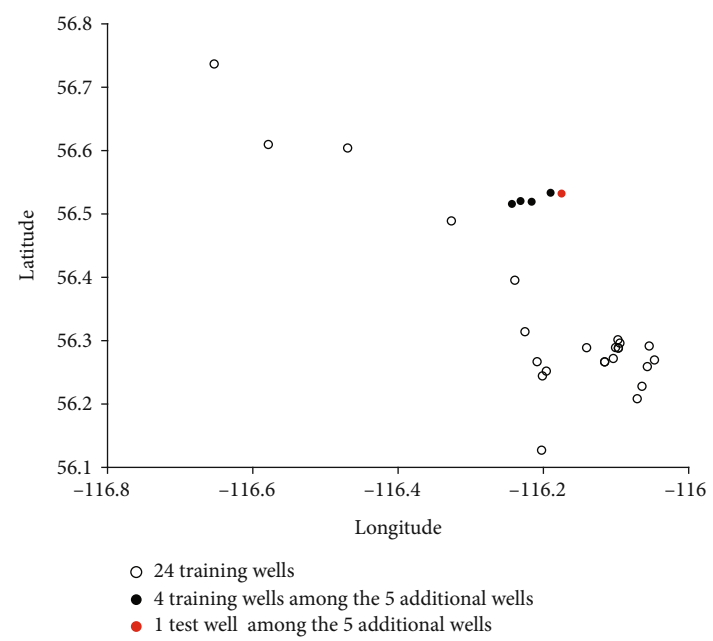
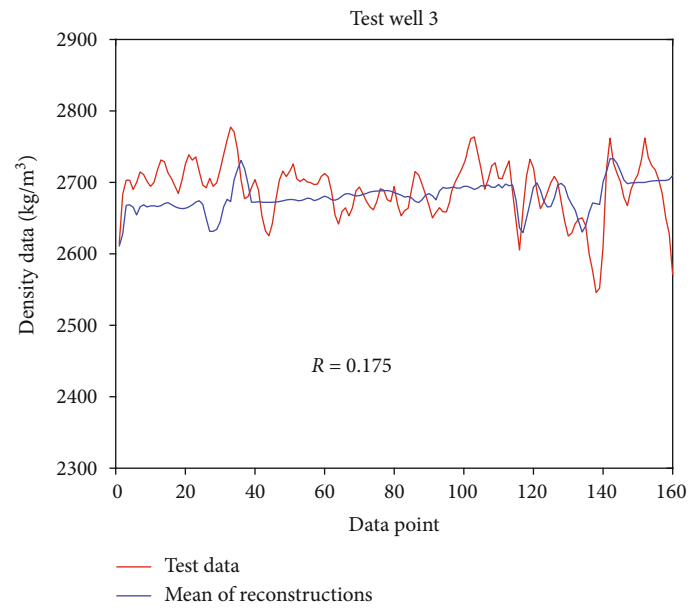


FIGURE 16: Continued.



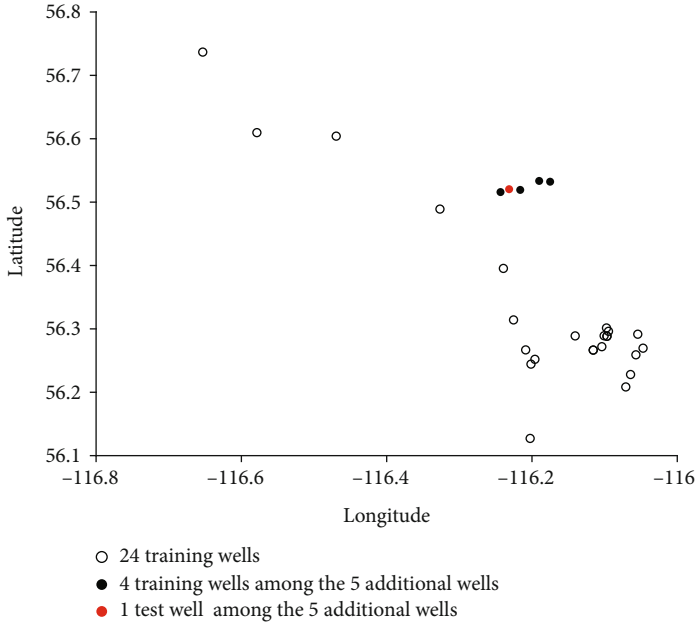
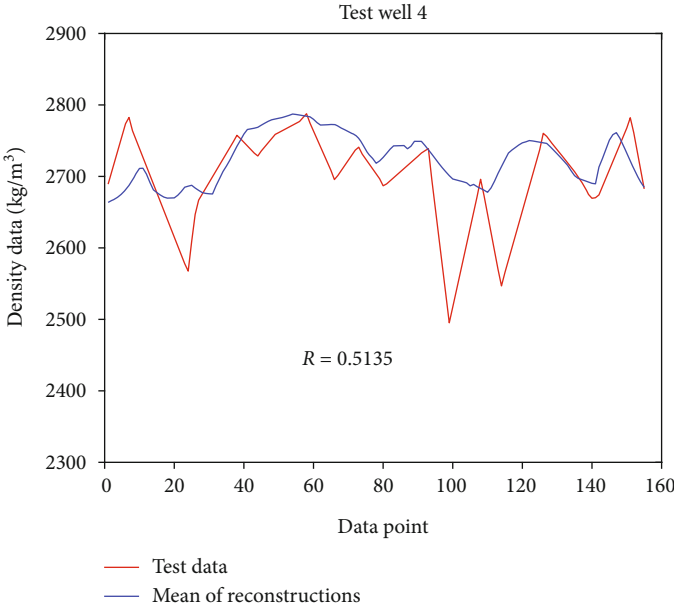
(b)

FIGURE 16: Continued.



(c)

FIGURE 16: Continued.



(d)

FIGURE 16: Continued.

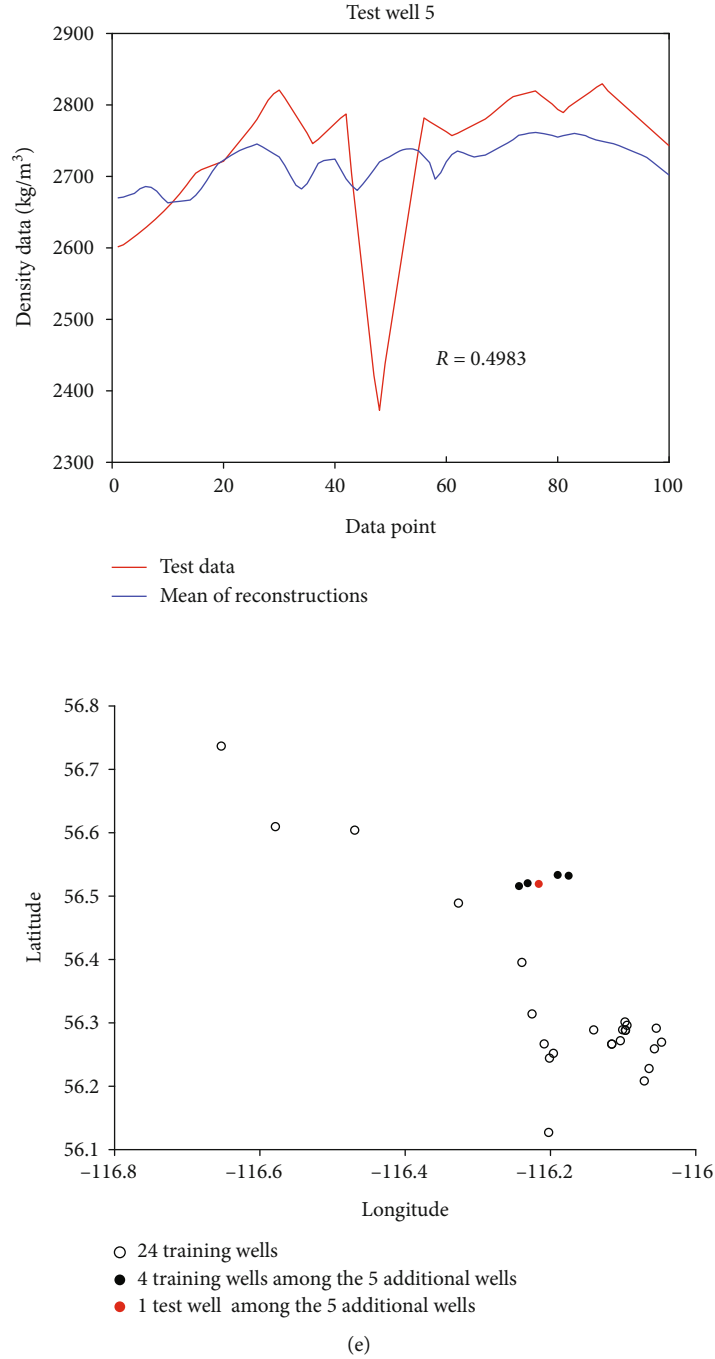


FIGURE 16: The five subordinate cases 2-1, 2-2, 2-3, 2-4, and 2-5 with different test wells in Case 2.

Table 6). Generally, the test error of Case 2 decreases by about a half of the error from the best of Case 1. These results can be derived owing to two aspects. First, the number of training data increases in Case 2 over Case 1. Second, geologically and spatially suitable wells are helpful to figure out the nonlinear relationship between the sonic and density logs for the target field.

However, a limitation still exists. Although the synthetic log in Figure 16(e) mimics the overall trend of the true density curve, it fails to predict the abnormal value around 2,400 kg/m<sup>3</sup> near the 50th data point. Despite this problem,

the results in Figure 16 can be seen as a reasonable prediction because it is more difficult to generate density log than other logs, such as acoustic and resistivity logs [1, 4].

Table 7 summarises the results of error calculation according to the following equation.

$$\text{Error}_k = \frac{1}{m_k} \sum_{l=1}^{m_k} \frac{|\text{Recon}_{k,l} - \text{Test}_{k,l}|}{\text{Test}_{k,l}} \times 100 (\%), \quad \text{for } k = 1, \dots, 5 \tag{5}$$



TABLE 7: Comparison of the relative test data error in Cases 1 and 2.

Test well	Case 1 (the best)	Case 1 (the worst)	Case 2
The 1 <sup>st</sup> well	3.03	3.11	2.23
The 2 <sup>nd</sup> well	2.06	4.61	1.50
The 3 <sup>rd</sup> well	2.71	5.56	1.26
The 4 <sup>th</sup> well	1.93	2.49	1.60
The 5 <sup>th</sup> well	2.98	4.75	2.34
Average	2.54	4.10	1.79

where  $\text{Error}_k$  is the error of the  $k$ th test well and  $m_k$  is the number of data points of the  $k$ th test well.  $\text{Recon}_{k,l}$  is the reconstruction for the  $l$ th data point of the  $k$ th test well, and  $\text{Test}_{k,l}$  is the true density log corresponding to  $\text{Recon}_{k,l}$ . The difference between the error calculations in Equations (4) and (5) is the use of normalisation with the true value in Equation (5). In general, if an error is 10%, it means that the reconstruction is deviated from the test as much as 10%. In the best and worst of Case 1, the errors are approximately 2% to 5% (Table 7). Especially, the test wells 3 and 5 show large errors. On the contrary, Case 2 gives acceptable errors for both wells. Especially, the error of the test well 3 in Case 2 is outstanding compared to the best one of Case 1. Generally, the overall relative errors by Equation (5) are within 2.5% for Case 2 and they can be taken as reasonable predictions.

#### 4. Conclusions

This study proposed how to determine the structure of a neural network (input and output layers) in terms of fit-for-purpose of a target field and how to prepare the training data based on geological knowledge. After an intensive review of the target field, we decided the sonic and density logs as input and output parameters in the neural network, respectively. Because only the 5 wells were available in the target field for training, the additional 29 pairs of the sonic and density logs were obtained by the criterion of depositional environment of the 5 wells.

Three conclusions were obtained from this study. First, the proper procedure of data acquisition was necessary for the successful generation of pseudodensity data. Although the trained network of Case 1 showed the underestimation problem for the five wells in the Golden field, its hyperparameters were suitable to train the data in Case 2. That meant the data of Case 1 work for determining proper training conditions for the geological environment in the target field. Second, all the obtained data should be carefully preprocessed based on proper domain knowledge considering the circumstances in the target field. We filtered unqualified data based on the correlation between the sonic and density log data. Further, unrealistic log data (e.g., constant or positive values) were eliminated. Because the logs and location data had different scales and units, min–max normalisation was applied for each parameter. Third, additional data that were geologically related to the target field were critical for the performance of a trained network. In Case 2, the four well data in

the Golden field were helpful to solve the underestimation problem of the synthetic density log from Case 1. The average of relative errors between the synthetic and true density logs were 2.54% and 1.79% in Cases 1 and 2, respectively.

In a future study, we will prove the benefit of the proposed approach for building a porosity model and predictions of oil production rates. Using the trained DNN, the 12 sonic log data can be transformed to synthetic density logs and both the given 17 density logs and the synthetic 12 density logs can be used in a combined way for building a porosity model. Then, it can be used to implement reservoir simulation for predicting well performances. These results will be compared with a porosity model from only the 12 density logs and its well predictions.

#### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request and approval of the operator company.

#### Conflicts of Interest

The authors declare no conflict of interest.

#### Acknowledgments

The authors are thankful to the Harvest Operations Corporation (HOC) and Korea National Oil Corporation (KNOC) for providing valuable field data and the approval for publishing this research paper. Especially, the authors are grateful to Mr. Mark Baker and Mr. Ray Pollock in HOC and Mr. Jinwook Chung and Mr. Yanggook Yi in KNOC for their helpful comments on this study. This research was supported by the project of Korea Institute of Geoscience and Mineral Resources (No. GP2017-024) and the project of Korea Institute of Energy Technology Evaluation and Planning granted financial resources from the Ministry of Trade, Industry, and Energy, Republic of Korea (No. 20172510102090). Dr. Baehyun Min has been partially supported by the National Research Foundation of Korea (NRF) grants (no. 2018R1A6A1A08025520 and no. 2019R1C1C1002574).

#### References

- [1] D. Zhang, Y. Chen, and J. Meng, "Synthetic well logs generation via recurrent neural networks," *Petroleum Exploration and Development*, vol. 45, no. 4, pp. 629–639, 2018.
- [2] M. M. Salehi, M. Rahmati, M. Karimnezhad, and P. Omidvar, "Estimation of the non records logs from existing logs using artificial neural networks," *Egyptian Journal of Petroleum*, vol. 26, no. 4, pp. 957–968, 2017.
- [3] W. Long, D. Chai, and R. Aminzadeh, "Pseudo density log generation using artificial neural network," in *SPE Western Regional Meeting*, Anchorage, AK, USA, May 2016.
- [4] L. Rolon, S. D. Mohaghegh, S. Ameri, R. Gaskari, and B. McDaniel, "Using artificial neural networks to generate synthetic well logs," *Journal of Natural Gas Science and Engineering*, vol. 1, no. 4-5, pp. 118–133, 2009.

- [5] M. Korjani, A. S. Popa, E. Grijalva, S. Cassidy, and I. Ershaghi, "A new approach to reservoir characterization using deep learning neural networks," in *SPE Western Regional Meeting*, Anchorage, AK, USA, May 2016.
- [6] O. Akinnikawe, S. Lyne, and J. Roberts, "Synthetic well log generation using machine learning techniques," in *Proceedings of the 6th Unconventional Resources Technology Conference*, Houston, TX, USA, July 2018.
- [7] M. Korjani, A. S. Popa, E. Grijalva, S. Cassidy, and I. Ershaghi, "Reservoir characterization using fuzzy kriging and deep learning neural networks," in *SPE Annual Technical Conference and Exhibition*, Dubai, UAE, September 2016.
- [8] G. A. Alzate, A. Arbelaez-Londono, A. N. Agudelo et al., "Generating synthetic well logs by artificial neural networks (ANN) using MISO-ARMAX model in Cupiagua field," in *SPE Latin America and Caribbean Petroleum Engineering Conference*, Maracaibo, Venezuela, May 2014.
- [9] J. H. Craig, "Depositional environments of the Slave Point Formation, Beaverhill Lake Group, Peace River Arch," in *Devonian Lithofacies and Reservoir Styles in Alberta: 13th CSPG core Conference and Display*, pp. 181–199, Calgary, Alberta, Canada, August 1987.
- [10] E. G. Gosselin, L. Smith, and D. J. Mundy, "The Golden and Evi reef complexes, Middle Devonian, Slave Point Formation, northwestern Alberta," *Canadian Society of Petroleum Geologists Memoir*, vol. 13, pp. 440–447, 1988.
- [11] S. Machielse, "Devonian algae and their contribution to the western Canadian sedimentary basin," *Bulletin of Canadian Petroleum Geology*, vol. 20, no. 2, pp. 187–237, 1972.
- [12] M. H. Rider and M. Kennedy, *The geological interpretation of well logs*, Rider-French Consulting Limited, 3rd revised edition, 2011.
- [13] K. Lee, J. Lim, D. Yoon, and H. Jung, "Prediction of shale-gas production at Duvernay Formation using deep-learning algorithm," *SPE Journal*, vol. 24, no. 6, pp. 2423–2437, 2019.
- [14] S. D. Mohaghegh, R. Gaskari, and M. Maysami, "Shale analytics: making production and operational decisions based on facts: a case study in Marcellus shale," in *SPE Hydraulic Fracturing Technology Conference and Exhibition*, The Woodlands, TX, USA, January 2017.
- [15] J. B. Dunham, G. A. Crwaford, and W. Panasiuk, "Sedimentology of the Slave Point Formation (Devonian) at Slave Field Lubicon Lake, Alberta," in *The Society of Economic Paleontologists and Mineralogists (SEPM) Carbonate Buildups (CW4)*, pp. 73–111, Special Publications of Society for Sedimentary Geology, 1983.
- [16] J. Jin, M. Li, and L. Jin, "Data normalization to accelerate training for linear neural net to predict tropical cyclone tracks," *Mathematical Problems in Engineering*, vol. 2015, Article ID 931629, 8 pages, 2015.
- [17] T. Jayalakshmi and A. SanthaKumaran, "Statistical normalization and back propagation for classification," *International Journal of Computer Theory and Engineering*, vol. 3, no. 1, pp. 89–93, 2011.
- [18] P. Panja, R. Velasco, M. Pathak, and M. Deo, "Application of artificial intelligence to forecast hydrocarbon production from shales," *Petroleum*, vol. 4, no. 1, pp. 75–89, 2018.
- [19] S. Ruder, "An overview of gradient descent optimization algorithms," <https://arxiv.org/abs/1609.04747>.
- [20] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, no. 2, pp. 149–154, 1964.
- [21] X. B. Jin, X. Y. Zhang, K. Huang, and G. G. Geng, "Stochastic conjugate gradient algorithm with variance reduction," <https://arxiv.org/abs/1710.09979>.
- [22] J. Yu, S. V. N. Vishwanathan, S. Gunter, and N. N. Schraudolph, "A quasi-Newton approach to nonsmooth convex optimization problems in machine learning," *Journal of Machine Learning Research*, vol. 11, pp. 1145–1200, 2010.