

Article

Permeability Prediction Using Machine Learning Methods for the CO₂ Injectivity of the Precipice Sandstone in Surat Basin, Australia

Reza Rezaee * and Jamiu Ekundayo 

Western Australian School of Mines, Minerals, Energy and Chemical Engineering, Curtin University, Perth 6102, Australia; jamiu.ekundayo@curtin.edu.au

* Correspondence: r.rezaee@curtin.edu.au

Abstract: This paper presents the results of a research project which investigated permeability prediction for the Precipice Sandstone of the Surat Basin. Machine learning techniques were used for permeability estimation based on multiple wireline logs. This information improves the prediction of CO₂ injectivity in this formation. Well logs and core data were collected from 5 boreholes in the Surat Basin, where extensive core data and complete sets of conventional well logs exist for the Precipice Sandstone. Four different machine learning (ML) techniques, including Random Forest (RF), Artificial neural network (ANN), Gradient Boosting Regressor (GBR), and Support Vector Regressor (SVR), were independently trained with a wide range of hyper-parameters to ensure that not only is the best model selected, but also the right combination of model parameters is selected. Cross-validation for 20 different combinations of the seven available input logs was used for this study. Based on the performances in the validation and blind testing phases, the ANN with all seven logs used as input was found to give the best performance in predicting permeability for the Precipice Sandstone with the coefficient of determination (R^2) of about 0.93 and 0.87 for the training and blind data sets respectively. Multi-regression analysis also appears to be a successful approach to calculate reservoir permeability for the Precipice Sandstone. Models with a complete set of well logs can generate reservoir permeability with R^2 of more than 90%.



Citation: Rezaee, R.; Ekundayo, J. Permeability Prediction Using Machine Learning Methods for the CO₂ Injectivity of the Precipice Sandstone in Surat Basin, Australia. *Energies* **2022**, *15*, 2053. <https://doi.org/10.3390/en15062053>

Academic Editor: Dameng Liu

Received: 22 February 2022

Accepted: 8 March 2022

Published: 11 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Surat Basin represents a highly prospective area for CO₂ storage in Eastern Australia [1,2], with a thick, relatively undisturbed sedimentary sequence providing large potential storage volume adjacent to major emission sources from coal-fired power stations. The Early Jurassic Precipice Sandstone is the target reservoir for upscaled storage trials in the area. Despite this potential for CO₂ storage, the need for better characterization of the storage site has been recommended [3]. The variation of porosity and permeability values and their ranges of uncertainties need to be realistically quantified for better prediction of CO₂ injectivity by a reservoir model [3–6].

For the CO₂ sequestration purpose, an optimum injection rate is necessary to increase the lifetime of a CO₂ storage operation. The “injectivity” term in the CCS community is defined as the “flow rate” which is controlled by several parameters such as reservoir permeability, thickness, and fluid properties [7,8]. A direct way of getting the well injectivity is to conduct an injectivity test for a borehole. However, apart from the cost and the technical issues associated with the test, the regional reservoir injectivity is more likely different from that of a single well injectivity test due to the reservoir heterogeneity [9,10]. There are some alternative methods to get reservoir injectivity, such as numerical simulations

and analytical models. For these methods, permeability is one of the most important input parameters that need to be provided with the highest possible accuracy.

Unlike other petrophysical properties, permeability is a difficult parameter to acquire from conventional well logs due to its dynamic nature. Several methods have been proposed for the estimation of permeability. Mohaghegh et al. [11] reported three major approaches for permeability estimation, including theoretical, statistical, and soft computing methods. Chehrazi and Rezaee [12] proposed a classification scheme of permeability prediction models, including theoretical models, soft computing models, and porosity-facies models. In theoretical modeling, usually, porosity and irreducible water saturation are inputs to calculate permeability. In models based on pore dimension, the pore dimension is obtained from mercury injection measurements and is interpreted as the pore throat size of some interconnected fraction of the pore system. One of the main drawbacks of theoretical models is the difficulty of obtaining parameters that need the core data.

Nowadays, with the increasing availability of cost-effective and efficient computing power, machine learning (ML) and artificial intelligence (AI) techniques are increasingly used to replace or augment traditional workflows in several industries. Simply put, AI is the branch of computer science focusing on developing machines and programs that can emulate human intelligence when performing assigned tasks or making a decision [13,14]. Machine learning is the subset of AI that deals with algorithms that allow machines to learn useful patterns from data [13,14].

2. Data Acquisition and Preparation

The required data for this study is collected from 5 wells (Woleebee Creek GW4, West Wandoan 1, West Moonie 1, Trelinga 1, Kenya East GW7) where extensive core data (a total of 460 core data) and complete sets of well logs exist for the Precipice Sandstone. Figure 1 shows an example of composite well logs of Woleebee Creek GW4. The Precipice Sandstone in most of the wells can be subdivided into the Lower and Upper members and a transition zone can also be identified between these two zones in some wells. The upper Precipice Sandstone is a shaly formation and shows relatively low porosity and permeability, whereas the Lower Precipice Sandstone is mostly a clean sandstone with relatively higher porosity and permeability.

Data Preparation

Log data requires proper quality control and editing to make them reliable as input parameters. All well logs were quality controlled, and overburden core porosity and Klinkenberg corrected permeability were calculated for all cored sections. A small number of data points were identified as outliers and were removed from the data set. No depth mismatching was observed and borehole quality over the Precipice Sandstone interval for all the wells was acceptable. No cycle skipping was observed for the sonic log. SP log data in Woleebee Creek GW4 do not correctly respond to the Precipice Sandstone; this could be either due to tool failure or the tool's inability to provide reliable values where the flushed zone thickness is very large. Surface core GR scan and core mini-perm data were used for depth matching between core and well logs.

The well logs used in this study as ML inputs are density (RHOB , g/cc), Neutron (NPHI , v/v), Photoelectric (PEF , b/e), resistivity (deep, shallow, and very shallow, ohm-m), and sonic (DT , us/ft). Instead of gamma-ray (GR), the volume of shale (Vsh , v/v) was used, since GR may vary from well to well for the same formation. Effective porosity calculated from the density tool (PHIDeffe , v/v) was also included to increase ML performance. Based on the heatmap of Spearman's rank correlation coefficients, Figure 2 shows the importance of all log inputs in predicting permeability.

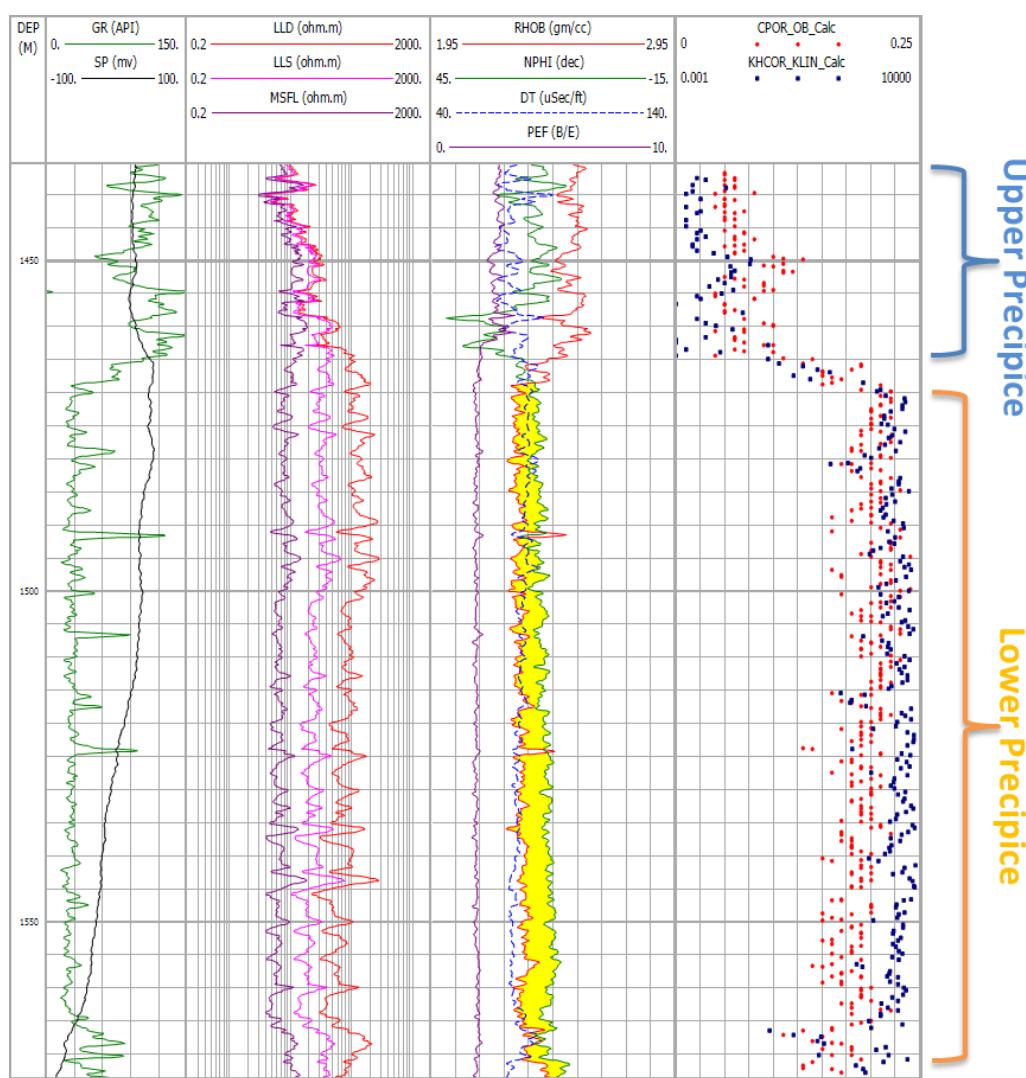


Figure 1. An example of typical well logs for the Precipice Sandstone in Woleebee Creek GW4. As can be seen in Track 1, the SP log is not responding correctly for the Lower Precipice Sandstone. Track 4 shows core porosity and permeability (CPOR_OB = core porosity at overburden condition; KHCOR_KLIN = Klinkenberg corrected core horizontal permeability at reservoir condition).

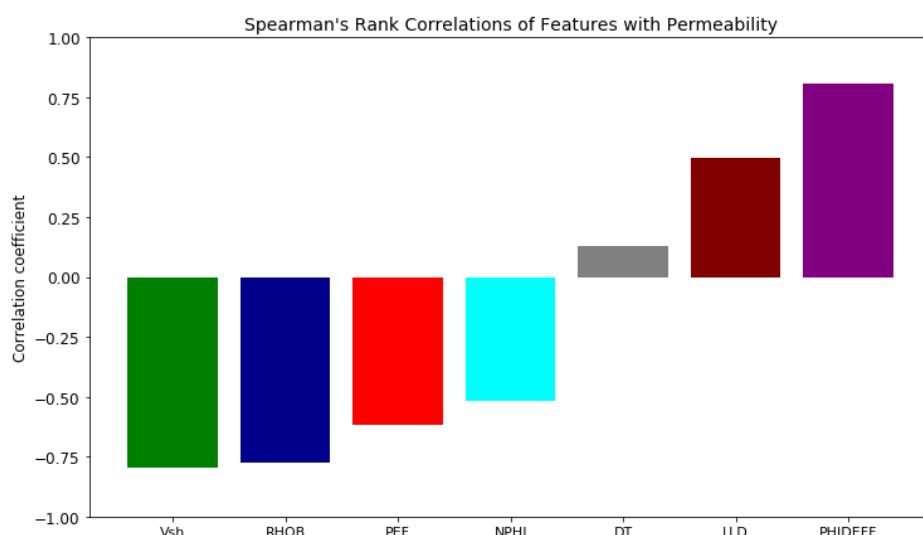


Figure 2. A plot of the relative importance of all features in predicting permeability.

3. Permeability Prediction with Machine Learning

One of the most important steps in predictive modelling with machine learning is model selection. Selecting the best of many algorithms for the problem at hand often requires training multiple algorithms with the same dataset and comparing their performances in both training and validation phases. In this study, four different ML algorithms—Random Forest (RF) regressor [15], Multi-Layer Perceptron/Artificial neural network (ANN) regressor [16], Gradient Boosting Regressor (GBR) [17], and Support Vector Regressor (SVR) [18]—were independently trained with a wide range of hyper-parameters to ensure that the best model is selected, and also that the right combination of model parameters is selected. The scikit-learn library [19] provides a good Python implementation for each of these algorithms.

Random Forest Regressor (RF): In this study, we applied the Python implementation of the random forest regression algorithm [15] provided by the scikit-learn’s ensemble class. This function uses multiple hyper-parameters to fit the model to the input data. A complete list and meanings of the required hyper-parameters are available on the official webpage of scikit-learn. However, the following selected hyper-parameters that were considered most important in optimizing the performance of the RF regressor model used were the number of estimators (`n_estimators`), maximum tree depth (`max_depth`), minimum samples split (`min_samples_split`), and minimum samples leaf (`min_samples_leaf`). The number of estimators represents the number of independent decision trees in the ensemble. Increasing this parameter typically leads to the improved overall performance of the RF model but could also increase the computational time significantly [20]. Maximum tree depth is the maximum depth of each tree in the ensemble. It controls the number of splits, and hence the complexity of each tree [20]. Larger depths tend to improve model performance but increase the computational time exponentially. Larger tree depths could also lead to lower prediction stability, making the model prone to overfitting [20]. Minimum samples split is the lowest number of samples required to split a node. Combined with the `n_estimators`, `max_depth`, and `min_samples_leaf`, this parameter has a significant effect on the size of the model [19] and consequently its complexity and computational time requirement. Lastly, minimum samples leaf is the lowest number of samples required to initiate a split point at any depth. It is also the minimum number of samples that must be present at a leaf node [19].

Multi-Layer Perceptron/Artificial Neural Network (MLP/ANN): This is the most used ML algorithm in predicting petrophysical properties from wireline logs. In its simplest form, an ANN consists of a fully connected architecture composed of an input layer, a hidden layer, and an output layer (Figure 3). Each layer is composed of neurons that are sometimes referred to as nodes. The nodes in the input layer are connected to those in the hidden layer, which are in turn connected to those in the output layer. A node-to-node connection is controlled by an assigned weight, which is adjusted after successive training iterations through a technique known as back-propagation [21]. The type of ANN described above is sometimes referred to as a shallow neural network because it contains only one hidden layer. If the number of hidden layers is greater than one, the ANN is referred to as a deep neural network.

This study used a shallow neural network built-in Python using the scikit-learn multi-layer perception regression function [19]. A neural network has several hyper-parameters that can be tuned to improve its predictive performance. In this study, the following hyper-parameters tuned for the neural network model were the activation function, penalization factor, number of nodes in the hidden layer, and maximum number of iterations. The activation function provides a means of transforming the calculated weighted sum of the input signal into an output signal to be fed as input for the next layer [22].

Gradient Boosting Regressor (GBR): GBR is an example of boosting ensemble ML models. GBR minimizes a loss function using gradient descent. Again, the Scikit-learn implementation of the GBR was adopted in this study. Due to its similarity to the RF regressor, the GBR uses similar hyper-parameters as those already defined for RF regressor. The only additional hyper-parameter considered for optimization in this study is the

“min_impurity_decrease” which is the minimum decrease in node impurity required for the node to split [19]. Typically, the tree branch in the ensemble is homogenous, and thus, a parameter to measure the level of contamination of the branch is known as the impurity measure [23].

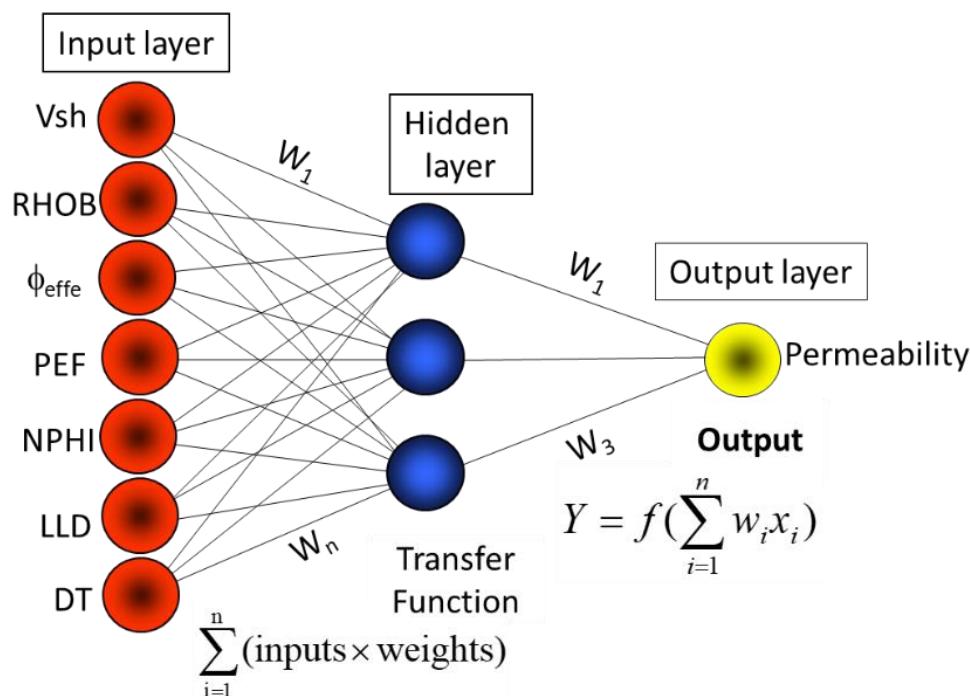


Figure 3. A typical neural network is composed of three layers: input, middle (hidden), and output.

Support Vector Regressor (SVR): SVR is the regression version of the support vector machine (SVM) often used in classification problems. It approximates a given data to a continuous function through a non-linear transformation that maps the data to a high-dimensional space [24]. It solves a convex optimization problem that minimizes an ε -insensitive loss function [24]. The model performance and complexity are controlled by hyper-parameters such as the kernel function used and its associated parameters, and the regularization factor (C) and the most commonly used are radial basis function (RBF) or gamma function, sigmoid, linear, or non-linear such as polynomial [19].

Each of the ML models were trained for all the different combinations of well logs shown in Table 1. This step was taken to ensure the applicability of this approach for wells with insufficient well log data.

The best combination of hyper-parameters was selected by searching through a range of values (search space) of each using a brute-force search algorithm known as the grid search, which is an exhaustive search through all the possible combinations of values in the search space. This method, although computationally intensive, is the most widely used hyper-parameter optimization technique [25]. In this study, grid search was performed for each training scenario, on the range of values shown in Table 2, using the GridSearchCV function in scikit-learn [19].

To ensure the selected model is generalizable, the grid search was done with cross-validation [26–28]. K-Fold cross-validation is particularly useful when developing ML models for small datasets [29,30]. It involves splitting the dataset into K different sets [31] and training the model with each set for every combination of hyper-parameters. The model performance is the average performance of all the K folds trained separately [32]. It is generally believed that as the value of K increases, the model's performance and stability also increase. This may not be true for small datasets where each split may not be large enough to produce a stable model, resulting in an unstable ensemble [32]. Even

for large datasets, the gain in model stability or performance may not offset the increased computational load required to train the additional models introduced by the increase in K. In this study, we applied 5-fold cross-validation to all the models. Our choice of K = 5 was guided by the evidence from the literature [31,32].

Table 1. Different combinations of input logs used as features.

Case ID	Features
case_1	RHOB, Vsh
case_2	RHOB, Vsh, PHIDEFF
case_3	RHOB, Vsh, PEF
case_4	RHOB, Vsh, PHIDEFF, PEF
case_5	RHOB, Vsh, NPHI, PHIDEFF
case_6	RHOB, Vsh, LLD, PHIDEFF
case_7	RHOB, Vsh, DT, PHIDEFF
case_8	RHOB, Vsh, NPHI, PEF
case_9	RHOB, Vsh, LLD, PEF
case_10	RHOB, Vsh, DT, PEF
case_11	RHOB, Vsh, NPHI, DT, PHIDEFF
case_12	RHOB, Vsh, LLD, DT, PHIDEFF
case_13	RHOB, Vsh, NPHI, PHIDEFF, PEF
case_14	RHOB, Vsh, LLD, PHIDEFF, PEF
case_15	RHOB, Vsh, NPHI, LLD, PHIDEFF
case_16	RHOB, Vsh, NPHI, LLD, PHIDEFF, PEF
case_17	RHOB, Vsh, NPHI, DT, PHIDEFF, PEF
case_18	RHOB, Vsh, LLD, DT, PHIDEFF, PEF
case_19	RHOB, Vsh, NPHI, LLD, DT, PHIDEFF
case_20	RHOB, Vsh, NPHI, LLD, DT, PHIDEFF, PEF

Table 2. Hyper-parameter search spaces and optimum values.

Model	Parameter	Search Space
RF	max_depth	80, 90, 100
	min_samples_split	8, 10, 12, 15
	min_samples_leaf	3, 4, 5
	n_estimators	300, 400
NN	activation	identity, logistic, tanh, relu
	alpha	0.000001–0.001
	hidden_layer_sizes	7–16
	max_iter	200, 400, 1000, 1500
GBR	min_impurity_decrease	0.0, 0.001, 0.00001
	learning_rate	0.0001, 0.001, 0.01, 0.1
	min_samples_split	8, 10, 20
	min_samples_leaf	1, 2, 5
	n_estimators	300, 500
SVR	kernel	RBF, Sigmoid
	C	1, 3, 5, 7, 9
	degree	3–6
	coef0	0.01, 0.5, 10
	gamma	0.001, 0.01, 0.1

In this study, the model's performances in the following four phases—training, validation, combined training and validation, and blind testing—were assessed using a coefficient of determination (R^2) and root mean square error (RMSE) as metrics. However, the final prediction model was selected based on the performance in the blind testing phase.

3.1. Results and Discussion

Building a robust and generalizable machine learning model for regression requires several iterations to ensure the best model and the right combination of the model's hyper-parameter are selected. In this study, eighty different models were developed representing the twenty different combinations of input logs (as shown in Table 1) for each of the four ML algorithms listed in Table 2. Each of these models represents the best combination of hyper-parameters for that algorithm, following the extensive grid search. This "best estimator" corresponds to the combination of hyper-parameters that gave the maximum average test performance from the cross-validated samples. A comparison of the performances (R^2 values) for all the 80 models in three different phases—training, validation, and (blind) testing—is shown in Figure 4. It should be noted that these R^2 values were calculated by comparing the actual permeability values with those predicted by the best estimators. This was done to ensure a fair comparison among the models, and more importantly, between the validation and blind testing phases.

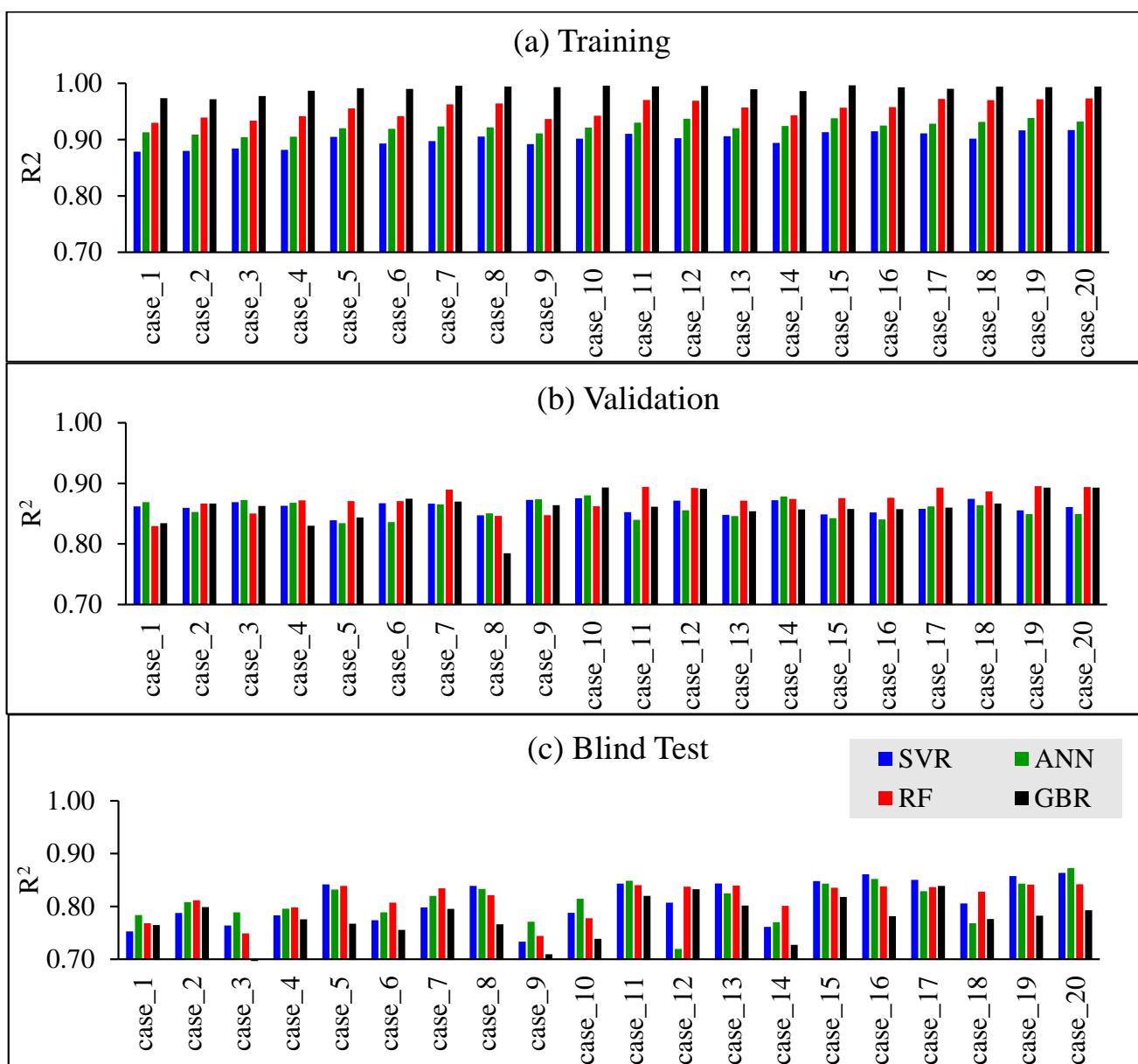


Figure 4. Comparing the performances of all 80 models in (a) training, (b) validation, and (c) testing.

All models achieved $R^2 > 0.9$ in training except SVR, which had $R^2 < 0.9$ for eight of the twenty cases. Based on training alone, the models can be ranked in order of performance as GBR > RF > ANN > SVR. Thus, the GBR and RF achieved the best performances in training compared to ANN and SVR. This is expected given that both GBR and RF are ensembles of multiple decision trees. Compared to the training phase, all the models achieved similar but lower performances in the validation phase, with the GBR producing the largest drops in performance despite having the largest training performances across all cases. A similar observation is made in testing R^2 values, where the R^2 -value for GBR fell to below 0.70 in case 3, representing an almost 29% reduction in performance relative to training. For most of the cases, RF still performed better than ANN and SVR in validation, and its performances appeared to fall to the same level as the duo during testing. This observation suggested that tree-based ensemble models (RF and GBR) may not be truly generalizable for the type and size of the dataset used in this study.

Despite its low training performances, SVR shows good generalizability, maintaining similar performances in validation and testing. As discussed, the penalty factor, the insensitive loss function, and the radial basis function are the major parameters controlling the performance of SVR [24]. Thus, higher training performance may be achieved without jeopardizing the generalizability by improving the search space for these parameters [24]. However, considering the large computation time requirement of SVR, compared to ANN, the potential gain in performance may not be worthwhile. ANN showed good performance in training while also achieving a similar level of robustness as SVR.

The Base Case Model

From the above discussion, it is clear that three (RF, SVM, and ANN) of the four algorithms tested in this study have comparable performances when blind tested with unseen data. However, as shown in Figure 5, ANN requires the least time to run all twenty cases, and as such has been adopted as the base algorithm to be used for the remaining modelling work conducted in this study. ANN appears to be relatively more robust, producing comparable performances (in training as well as in validation) for all twenty cases. However, case 20 gave the best performance in the testing phase, and as such has been adopted as the base scenario for the rest of this study. Thus, the base case model is an ANN with all the logs shown in Figure 2 as inputs. The network parameters for the ANN are identified below:

- Number of layers = 3
- Number of hidden layers = 1
- Number of neurons in hidden layer = 12
- Activation function = Rectified Linear Unit (ReLU)
- Cross-validation type = K-fold (K = 5)
- Learning rate = 0.001
- Solver = with Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS)
- Alpha = 0.001

Note that the number of nodes in the input layer is equal to the number of variables in the feature (seven for this base case) and the number of nodes in the output layer is the number of target variables (one in this case).

It should be noted that the model described above did not discriminate between the Upper and Lower Precipice Sandstone but treated it as a single “uniform” formation. Figure 6 shows a plot of the actual model against predicted permeability for the training, validation, and testing phases. As previously stated, the R^2 values were calculated by comparing the actual permeability values with those predicted by the model. An alternative would have been to report the average R^2 value for the cross-validation splits. However, this value is only available for the training and validation set, since the testing set was not exposed to the model.

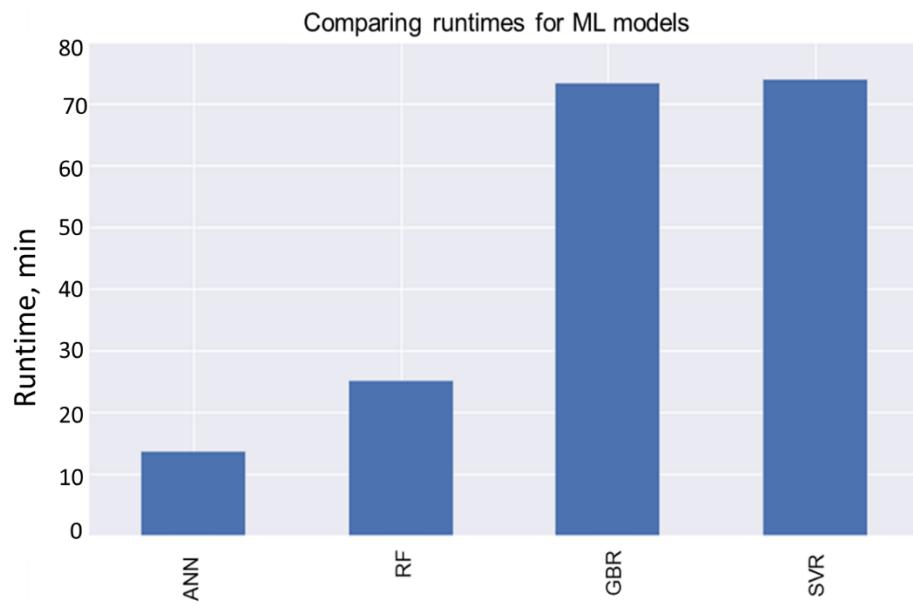


Figure 5. Comparison of cumulative runtime for different ML algorithms.

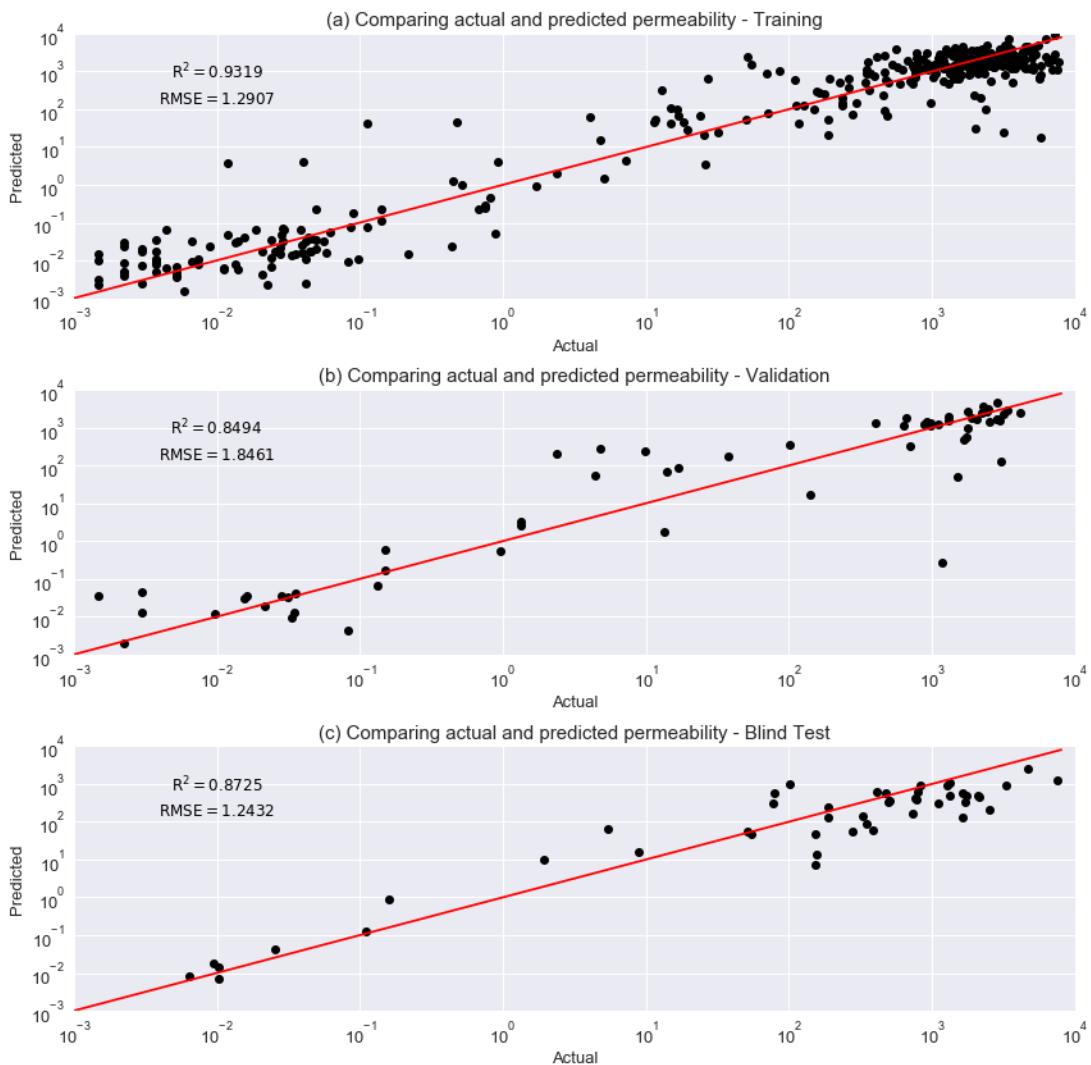


Figure 6. ANN performance for the training (a), validation (b), and blind testing (c) for the base case model where all logs were used as inputs.

The overall match between the actual and calculated permeability values is shown in Figure 7, for combined training and validation, and the testing datasets. The horizontal axes in these plots represent the index location of the data point in the dataset. Although a few mismatches (around 300–400 index) are obvious, the overall trend was well matched across the whole datasets in training and validation, as well as in the blind testing phase, and as such, the model is considered fit for the purpose of this study.

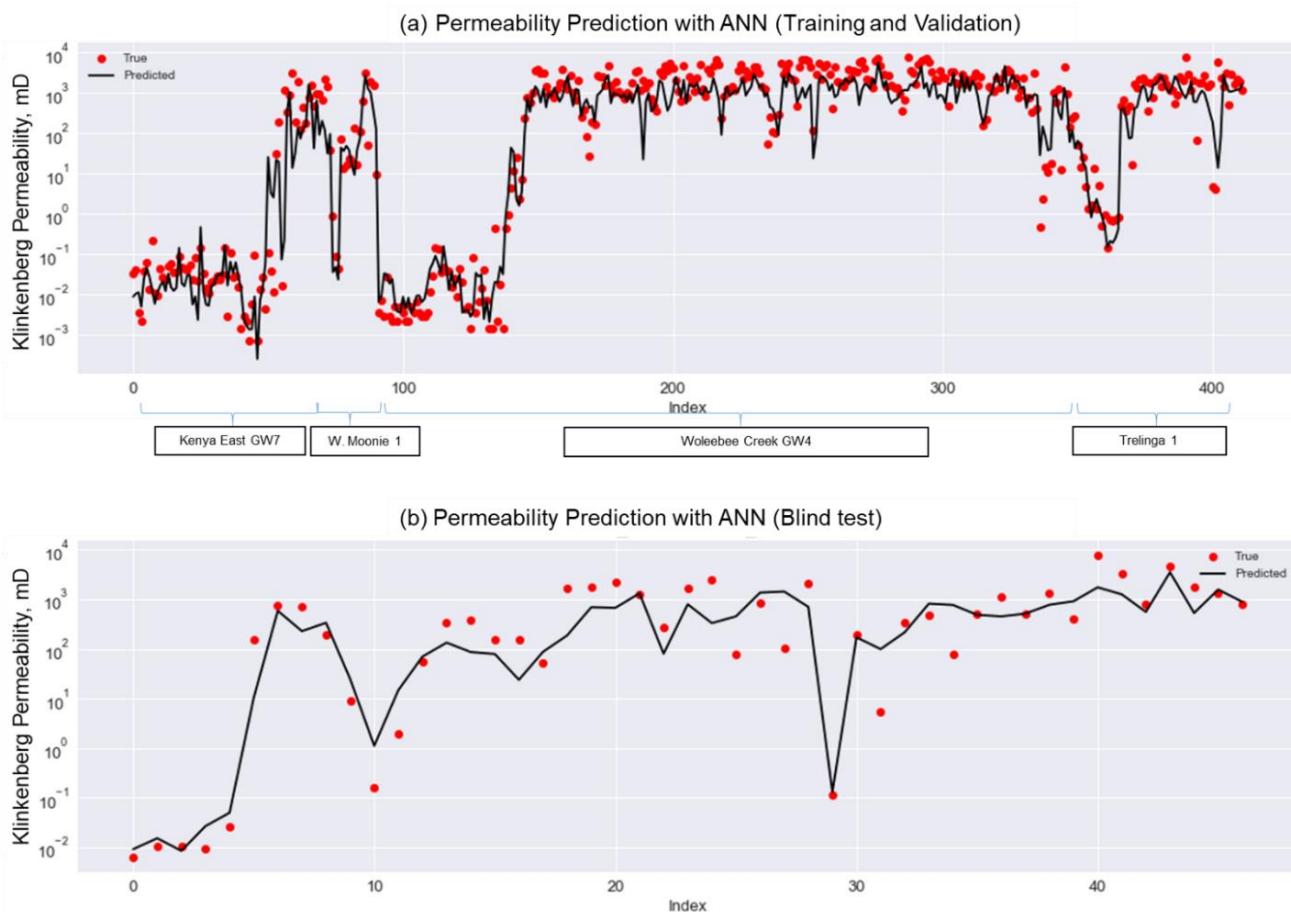


Figure 7. Comparing the match between the calculated and actual permeability for (a) combined training and validation dataset (b) testing dataset for West Wandoan 1.

4. Uncertainty Quantification

As discussed, a total of 20 different scenarios (Table 1) were modelled purposely to capture various possible combinations of the input well logs, but more importantly to provide another means of quantifying the range of uncertainties associated with using the base case model with such different combinations of the input well logs. To quantify these uncertainties, the base case model was used to train each of the other cases 1–19 and the mean permeability of each model was compared with that obtained from the base case. Figure 8 shows the percentage deviations of these mean permeability values (arranged in increasing order) calculated for cases 1–19 relative to case 20 (the base case). Cases 11, 13, 17, and 19 gave the lowest deviations from the base case value, while cases 1 and 6 deviated the most. The mean permeability calculated from case 1 was the lowest, while that calculated from case 6 was the highest of all the cases. Although cases with more than 4 logs as inputs appeared to deviate less from the base case (which used all 7 well logs), there is no discernible trend with the number of input logs. Obviously, the model performance is dependent not only on the number of logs, but also on the type of logs used as inputs.

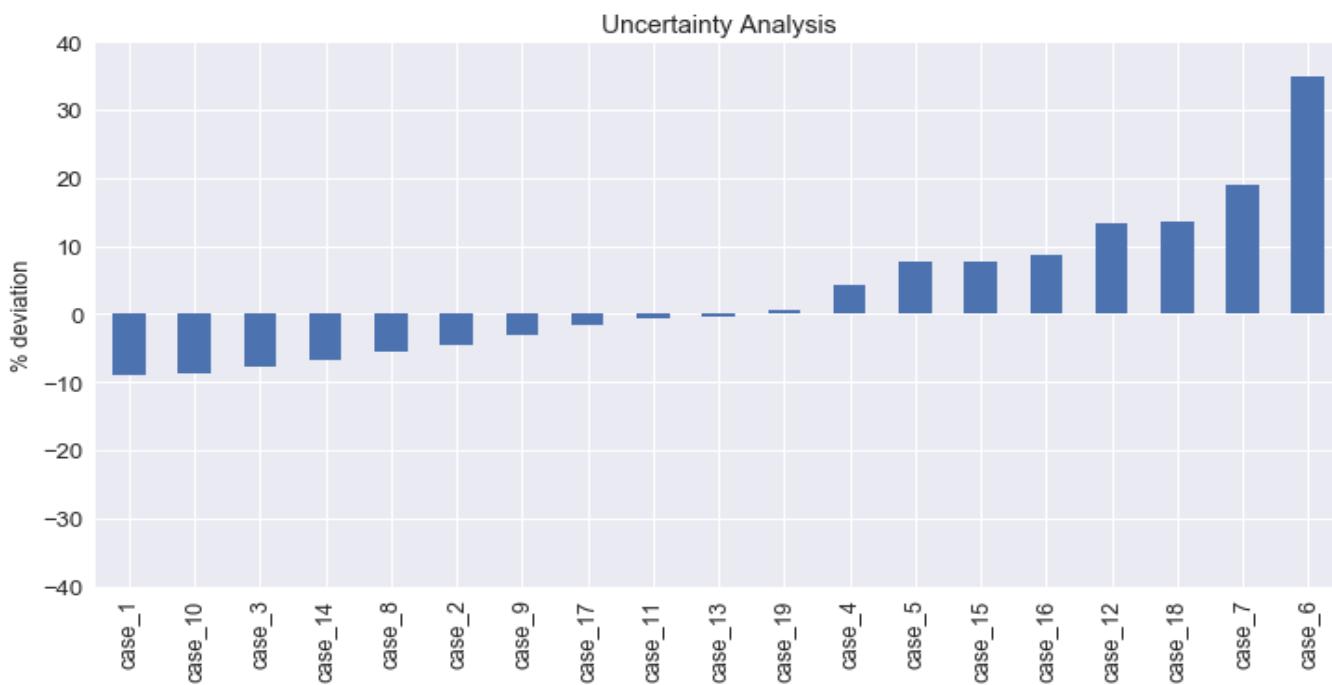


Figure 8. Uncertainty in average permeability based on different combinations of input well logs.

Figure 9 shows the predicted permeability for the base case compared to a couple of other cases representing the varying number of input logs. The shaded regions represent the uncertainty bounds resulting from training the base case model with each of the comparison cases. The effect of the number of input logs on the predicted permeability can be seen from these plots. The uncertainty bound appears to reduce as the number of input logs increases.

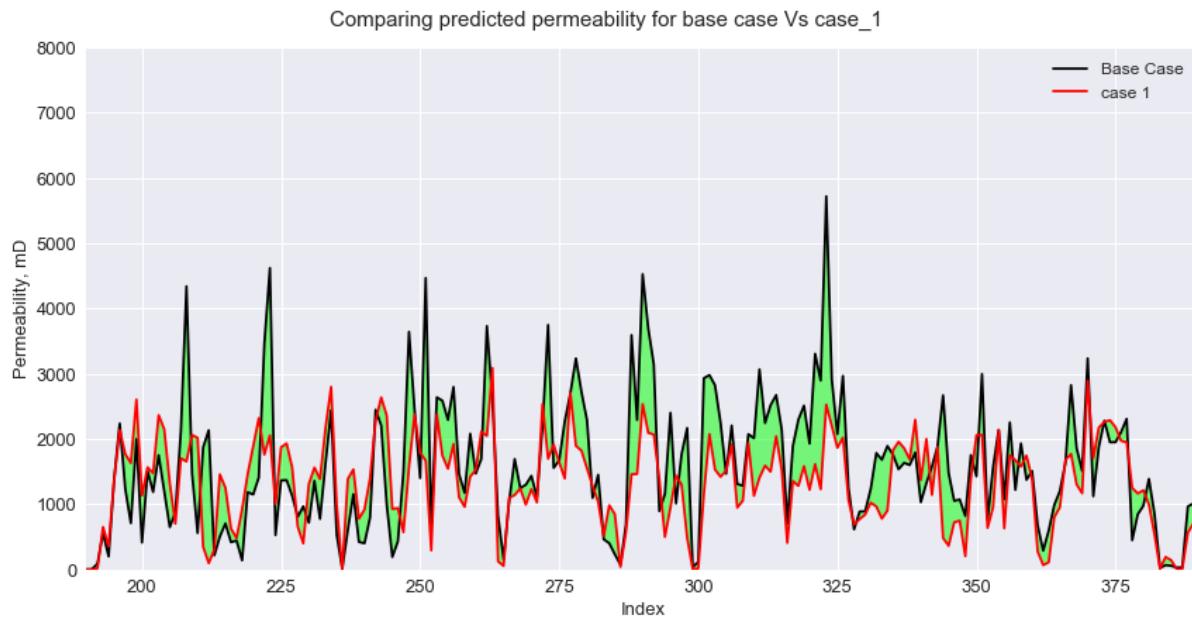


Figure 9. Cont.

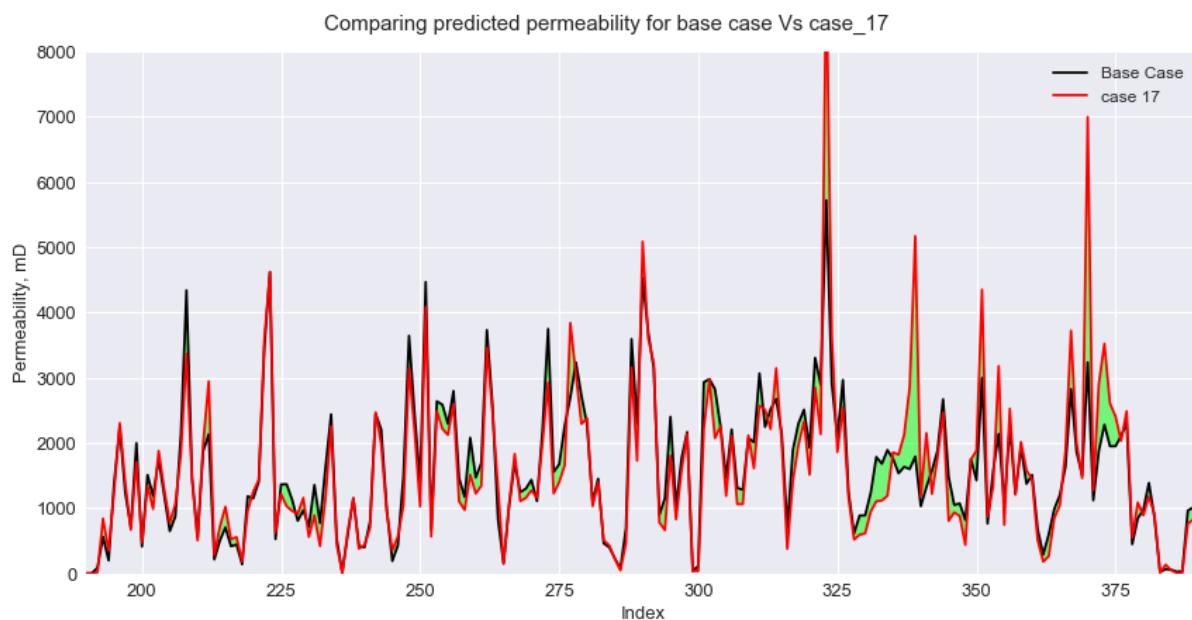


Figure 9. Uncertainty regions introduced by cases 1 (2 logs input, above) and 17 (6 logs input, below).

5. Multi-Regression Analysis

Since ANN is considered a black-box model that does not provide a tangible tool for its output prediction, the multi-regression analysis was also performed for the available dataset. Multi-regression is an extension of the regression analysis that incorporates additional independent variables in the predictive equation. The main reason for multi-regression is to find the relationship between several independent or predictor variables. The regression analysis has been used frequently as a predictive tool to find the relationship among the petrophysical properties of rock, including porosity and permeability [33,34].

To evaluate the applicability of the regression analysis to predict permeability from well logs for the Precipice Sandstone, the same dataset of this study was used utilizing IBM SPSS Statistics. Table 3 lists developed equations from the multi-regression analysis ranked based on their coefficient of determination (R^2). These empirical equations help calculate reservoir permeability in wells with different combinations of available well logs. Based on this approach, it appears that RHOB, Vsh, and effective porosity with the highest weight are the most influential inputs, respectively.

Table 3. Empirical equations developed from regression analysis using all data. R^2 values show the coefficient of determination between measured and calculated permeability.

Models	Equations	R^2
1	$\text{Log } k = 50.47 + (-18.266 \times \text{RHOB}) + (-1.443 \times \text{Vsh}) + (-1.74 \times \text{PHIDeffe}) + (-0.041 \times \text{NPHI}) + (0.001 \times \text{LLD}) + (-0.195 \times \text{PEF}) + (-0.051 \times \text{DT})$	0.902
2	$\text{Log } k = 52.202 + (-18.931 \times \text{RHOB}) + (-1.519 \times \text{Vsh}) + (-0.689 \times \text{PHIDeffe}) + (-0.039 \times \text{NPHI}) + (-0.181 \times \text{PEF}) + (-0.053 \times \text{DT})$	0.902
3	$\text{Log } k = 51.667 + (-18.896 \times \text{RHOB}) + (-1.43 \times \text{Vsh}) + (-0.042 \times \text{NPHI}) + (-0.053 \times \text{DT})$	0.902
4	$\text{Log } k = 56.659 + (-20.123 \times \text{RHOB}) + (-1.701 \times \text{Vsh}) + (-1.298 \times \text{PHIDeffe}) + (-0.233 \times \text{PEF}) + (-0.081 \times \text{DT})$	0.900
5	$\text{Log } k = 56.472 + (-20.225 \times \text{RHOB}) + (-1.55 \times \text{Vsh}) + (-0.083 \times \text{DT})$	0.899
6	$\text{Log } k = 43.026 + (-16.671 \times \text{RHOB}) + (-1.809 \times \text{Vsh}) + (-0.069 \times \text{NPHI})$	0.898
7	$\text{Log } k = 22.109 + (-8.669 \times \text{RHOB}) + (-1.311 \times \text{Vsh}) + (9.959 \times \text{PHIDeffe}) + (-0.454 \times \text{PEF})$	0.886
8	$\text{Log } k = 26.461 + (-10.794 \times \text{RHOB}) + (-1.582 \times \text{Vsh}) + (8.739 \times \text{PHIDeffe})$	0.883
9	$\text{Log } k = 9.825 + (-4.605 \times \text{RHOB}) + (20.043 \times \text{PHIDeffe})$	0.883
10	$\text{Log } k = 39.739 + (-15.763 \times \text{RHOB}) + (-2.735 \times \text{Vsh})$	0.883
11	$\text{Log } k = -1.738 + (24.746 \times \text{PHIDeffe})$	0.881
12	$\text{Log } k = 56.329 + (-23.06 \times \text{RHOB})$	0.848

Figure 10 shows the plot of measured permeability versus those calculated from the equations shown in Table 3. As can be seen in Table 3 and Figure 10, most of the models with different inputs are successful in predicting permeability. Clearly, those models with a higher number of suitable inputs (Models 1 to 3) are more successful (R^2 more than 0.9) in predicting permeability.

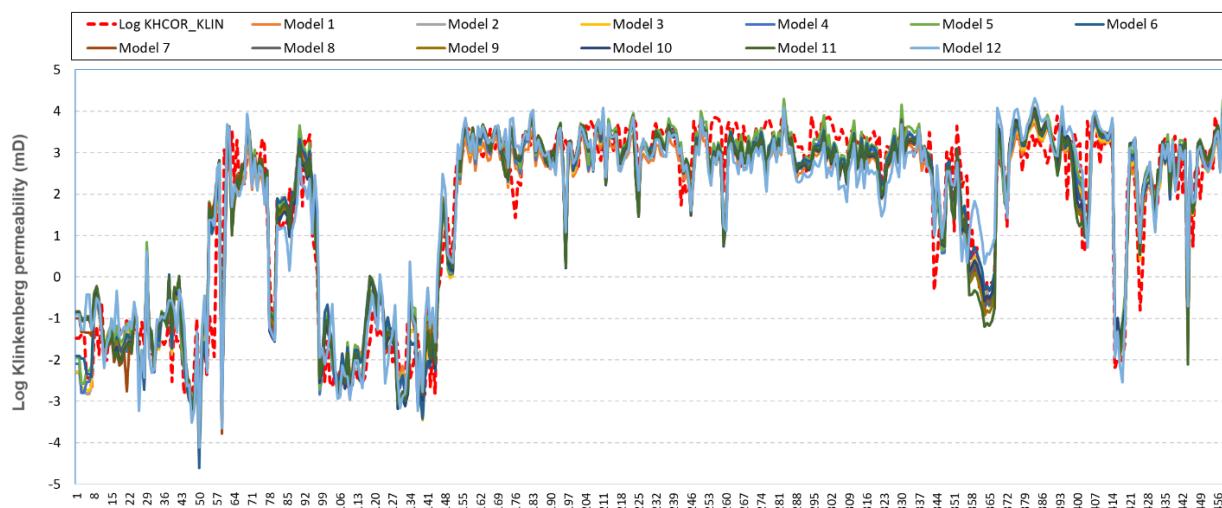


Figure 10. A comparison between permeability calculated from multi-regression analysis and measured Klinkenberg corrected permeability (KHCOR_KLIN).

6. Conclusions

This study used ML methods and uncertainty analysis to provide a robust tool for permeability estimation for Precipice Sandstone using conventional well logs. The required data for this study is collected from 5 wells in Surat Basin, where extensive core data and complete sets of well logs exist for the Precipice Sandstone. All well logs were quality controlled, and surface core GR scan and core mini-perm data were used for depth matching between core and log data. Overburdened core porosity and Klinkenberg corrected permeability were calculated for all cored sections from the correlations established in wells with special core analysis measurements.

Four different ML algorithms (RF regressor, GBR, SVR, and ANN) were developed with cross-validation for 20 different combinations of the seven available input well logs. Based on the performances in the validation and (especially in the) blind testing phases, the ANN was found to be the best model for our purpose, mainly because it requires the lowest runtime and more importantly because it was relatively more robust, producing comparable performances for all scenarios tested. The case with all seven logs (case 20) used as input was found to give the best performance in the blind testing phase, and as such, was chosen as the base case. Thus, the base model was ANN trained with case 20. This model was found to be useful in predicting permeability for the Precipice Sandstone.

Multi-regression analysis also appears to be a successful approach to calculate reservoir permeability for Precipice Sandstone. Models with a complete set of typical well logs can generate reservoir permeability with R^2 of more than 90%.

Author Contributions: Conceptualisation, R.R.; methodology, R.R. and J.E.; software, J.E. and R.R.; data curation and validation, R.R. and J.E.; writing—original draft, J.E.; writing—review and editing, R.R.; supervision, R.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Australian National Low Emissions Coal Research and Development. ANLEC R&D is supported by Low Emission Technology Australia (LETA) and the Australian Government through the Department of Industry, Science, Energy, and Resources.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors wish to acknowledge the financial assistance provided through Australian National Low Emissions Coal Research and Development. ANLEC R&D is supported by Low Emission Technology Australia (LETA) and the Australian Government through the Department of Industry, Science, Energy, and Resources.

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

References

1. Bradshaw, J. Regional Scale Assessment-Results & Methodology Queensland CO₂ Storage Atlas. In *Second EAGE CO₂ Geological Storage Workshop 2010*; European Association of Geoscientists & Engineers: Houten, The Netherlands, 2010.
2. Hodgkinson, J.; Grigorescu, M. Background research for selection of potential geostorage targets—Case studies from the Surat Basin, Queensland. *Aust. J. Earth Sci.* **2013**, *60*, 71–89. [\[CrossRef\]](#)
3. He, J.; La Croix, A.D.; Wang, J.; Ding, W.; Undershultz, J. Using neural networks and the Markov Chain approach for facies analysis and prediction from well logs in the Precipice Sandstone and Evergreen Formation, Surat Basin, Australia. *Mar. Pet. Geol.* **2019**, *101*, 410–427. [\[CrossRef\]](#)
4. Bianchi, M.; Kearsey, T.; Kingdon, A. Integrating deterministic lithostratigraphic models in stochastic realizations of subsurface heterogeneity. Impact on predictions of lithology, hydraulic heads and groundwater fluxes. *J. Hydrol.* **2015**, *531*, 557–573. [\[CrossRef\]](#)
5. Philip, R.; Mark, B. Reservoir Model Design: A Practitioner’s Guide. In *Reservoir Model Design*, 2015th ed.; Springer: Berlin/Heidelberg, Germany, 2015.
6. La Croix, A.D.; Harfoush, A.; Rodger, I.; Gonzalez, S.; Undershultz, J.R.; Hayes, P.; Garnett, A. Reservoir modelling notional CO₂ injection into the Precipice Sandstone and Evergreen Formation in the Surat Basin, Australia. *Pet. Geosci.* **2019**, *26*, 127–140. [\[CrossRef\]](#)
7. Cinar, Y.; Riaz, A.; Tchelepi, H.A. Experimental study of CO₂ injection into saline formations. *Spe J.* **2009**, *14*, 588–594. [\[CrossRef\]](#)
8. Sundal, A.; Miri, R.; Nystuen, J.P.; Dypvik, H.; Agaard, P. Modeling CO₂ distribution in a heterogeneous sandstone reservoir: The Johansen Formation, northern North Sea. In Proceedings of the EGU General Assembly Conference, Vienna, Austria, 7–12 April 2013.
9. Bachu, S. Review of CO₂ storage efficiency in deep saline aquifers. *Int. J. Greenh. Gas Control.* **2015**, *40*, 188–202. [\[CrossRef\]](#)
10. Wang, Y.; Zhang, K.; Wu, N. Numerical investigation of the storage efficiency factor for CO₂ geological sequestration in saline formations. *Energy Procedia* **2013**, *37*, 5267–5274. [\[CrossRef\]](#)
11. Mohaghegh, S.; Balan, B.; Ameri, S. State-of-the-art in permeability determination from well log data: Part 2-verifiable, accurate permeability predictions, the touch-stone of all models. In *SPE Eastern Regional Meeting*; Society of Petroleum Engineers: Dallas, TX, USA, 1995.
12. Chehrazi, A.; Rezaee, R. A systematic method for permeability prediction, a Petro-Facies approach. *J. Pet. Sci. Eng.* **2012**, *82*, 1–16. [\[CrossRef\]](#)
13. Wills, E. AI vs. MACHINE LEARNING: The Devil Is in the Details. *Mach. Des.* **2019**, *91*, 56–60.
14. Jakhar, D.; Kaur, I. Artificial intelligence, machine learning and deep learning: Definitions and differences. *Clin. Exp. Dermatol.* **2020**, *45*, 131–132. [\[CrossRef\]](#)
15. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
16. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [\[CrossRef\]](#)
17. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [\[CrossRef\]](#)
18. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [\[CrossRef\]](#)
19. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
20. Liu, C.H.B.; Chamberlain, B.P.; Little, D.A.; Cardoso, Á. Generalising Random Forest Parameter Optimisation to Include Stability and Cost. In *Machine Learning and Knowledge Discovery in Databases*; Springer International Publishing: Cham, Switzerland, 2017.
21. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)
22. Jain, R.; Chotani, A.; Anuradha, G. 9-Disease diagnosis using machine learning: A comparative study. In *Data Analytics in Biomedical Engineering and Healthcare*; Lee, K.C., Ed.; Academic Press: Cambridge, MA, USA, 2021; pp. 145–161.
23. Laber, E.; Molinarom, M.; Pereira, F.M. Binary Partitions with Approximate Minimum Impurity. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 2854–2862.
24. Liu, Q.; Liu, W.; Mei, J.; Si, G.; Xia, T.; Quan, J. A New Support Vector Regression Model for Equipment Health Diagnosis with Small Sample Data Missing and Its Application. *Shock. Vib.* **2021**, *2021*, 6675078. [\[CrossRef\]](#)
25. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
26. Allen, D.M. The Relationship Between Variable Selection and Data Augmentation and a Method for Prediction. *Technometrics* **1974**, *16*, 125–127. [\[CrossRef\]](#)

27. Stone, M. Cross-Validatory Choice and Assessment of Statistical Predictions. *J. R. Stat. Soc. Ser. B* **1974**, *36*, 111–133. [[CrossRef](#)]
28. Stone, M. An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike’s Criterion. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 44–47. [[CrossRef](#)]
29. Vabalas, A.; Gowen, E.; Poliakoff, E.; Casson, A.J. Machine learning algorithm validation with a limited sample size. *PLoS ONE* **2019**, *14*, e0224365. [[CrossRef](#)] [[PubMed](#)]
30. Erofeev, A.; Orlov, D.; Ryzhov, A.; Koroteev, D. Prediction of Porosity and Permeability Alteration Based on Machine Learning Algorithms. *Transp. Porous Media* **2019**, *128*, 677–700. [[CrossRef](#)]
31. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence—Volume 2, Montreal, QC, Canada, 20–25 August 1995; Morgan Kaufmann Publishers Inc.: Montreal, QC, Canada, 1995; pp. 1137–1143.
32. Moss, H.; Leslie, D.; Rayson, P. Using J-K fold Cross Validation to Reduce Variance When Tuning NLP Models. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018.
33. Eskandari, H.; Rezaee, M.; Mohammadnia, M. Application of multiple regression and artificial neural network techniques to predict shear wave velocity from wireline log data for a carbonate reservoir South-West Iran. *CSEG Rec.* **2004**, *42*, 48.
34. Rezaee, M.R.; Jafari, A.; Kazemzadeh, E. Relationships between permeability, porosity and pore throat size in carbonate rocks using regression analysis and neural networks. *J. Geophys. Eng.* **2006**, *3*, 370–376. [[CrossRef](#)]