Estimation of Missing Petrophysical Data using a Machine Learning Based Bagging Approach

Ibrahim Olawale[1] and Fatai Anifowose[2]
[1]Federal University of Technology, Akure, Nigeria
[2]EXPEC Advanced Research Center, Saudi Aramco, Saudi Arabia

ABSTRACT

Proper analysis of petrophysical parameters is very crucial in the reservoir formation evaluation process. Petrophysical parameters are typically calculated from well logs. Missing values and incomplete well data also contribute to the degrees of complexity that are associated with the effort to reduce uncertainties. This in turn reduces the quality of the interpretation done to evaluate petroleum reservoirs. A machine learning approach to reproduce the missing values using the bootstrap aggregation (bagging) learning paradigm is proposed. Bagging is an ensemble machine learning technique used to evolve a consensus output from multiple tree-based learners. Six wells from a field in the North Sea were used to optimize and validate the models. To further demonstrate the efficiency of this method, missing sections on the neutron and density logs corresponding to the reservoir section of the test wells were also estimated. The bagging models significantly outperformed the traditional decision tree model with correlation coefficient (R2) scores in the range of 0.77 to 0.93, and root mean square error values as low as 3.99 for the validation wells. The results demonstrate the increased accuracy and reliability of the bagging machine learning paradigm to solve missing well information problems compared to the traditional single decision tree method. Furthermore, the result shows how bagging as a machine learning technique can be used to increase the quality of petrophysical interpretation. Consequently, this approach helps to reduce uncertainties involved in the reservoir formation evaluation process.

Keywords:

INTRODUCTION

High quality well log data are essential for an efficient and effective reservoir characterization process. This is because well log data are routinely used for wavelet estimation, low frequency model building, seismic velocity calibration, and time-to-depth conversion (Kumar et al.,2018). Reservoir formation evaluation is done using petrophysical properties and parameters calculated and derived from well logs. Petrophysical logs that are acquired through wireline logging are subjected to various operational conditions arising from the logging process. In most cases, these logs are recorded with missing values or missing sections arising from cost maintenance, geology of the environment or logging instrument malfunction. Petrophysical parameters calculated from well logs determine the final results of the formation evaluation process. These petrophysical parameters are usually calculated with some degrees of uncertainties arising from a number of assumptions ranging from the depositional environment, through the reservoir fluid type, lithology, stress, to strain conditions of the well bore. Missing values or sections in well logs essentially represent sections of no data, which can introduce noise to the formation evaluation process.

Methods of Estimating Petrophysical Properties

The most conventional way to predict missing data in petrophysical properties is to consider the correlation between the porosity and permeability given the well log records. This may however, result in inaccurate prediction (Al-Mudhafar et al.,, 2014). Apart from using machine learning methods for the estimation of missing petrophysical data, statistical methods and algorithms have also been used. Al-Mudhafar et al.,(2014) proposed the mean substitution (MS), iterative robust model-based imputation (IRMI), multiple imputation of incomplete multivariate data (MIIMD), random imputation of missing data (RIMD) algorithms for solving missing values for petrophysical analysis. The RMID method,

along with other algorithms proposed were applied based on the deductive statistical inference to input incomplete data (Al-Mudhafar et al.,, 2014). Robust sequential imputation algorithm estimates the missing values in a dataset by minimizing the determinant of the covariance of the augmented data matrix (Al-Mudhafar et al.,, 2014).

Machine learning methods have been used extensively for the prediction of petrophysical properties and estimating missing values. Such methods range from linear models to tree models, artificial neural networks, and recurrent neural networks. Chijioke et al., 2018 employed a ResNet deep learning architecture for well logs missing values forecast and using Apache Spark's distributed machine learning stochastic linear regression model. Model performance using the latter approach was poor giving an accuracy of about 10%. A two-layer artificial neural network (ANN) with the relu activation functions were applied. Model performance was estimated with the root mean square error (RMSE) and MSE. Their RMSE was estimated to be 10.2 for the test data. The ResNet model applied with auto encoders failed to improve their model performance. Gradient boosting decision tree (GBDT) and neural network approaches was proposed by Vito et al., 2020. While both model approaches were within acceptable limits using the MSE as metric, it was shown that the GBDT model consistently outperformed the neural network model (Vito et al., 2020). Rui et al., 2017 used ANN, random forest, gradient boosting, and linear models for estimating missing gaps in wireline logs. Statistical inference was applied for model performances. The ensemble methods (random forests and gradient boosting) performed better than the linear approaches for every gap size quartile, with statistical significance (Rui et al., 2017).

Tree Based Models

Tree based machine learning models are a family of machine learning algorithms that operate based on certain decisions met or not met in the data set. They are subdivided into decision trees, decision forests, and boosting trees models. A decision tree can be used to represent and implement simple statements or "decisions" to be executed based on certain conditions met. By deciding on which of the features of a data set to split on and when to stop splitting, a final decision is made by a decision tree. Each node in a tree terminates with either a value or triggers another conditional statement to be evaluated. Decision tree algorithms are implemented to minimize the entropy (degree of randomness or disorderliness) and to maximize information gain of features within the dataset. This is achieved by minimizing a certain loss function. The Gini impurity metric is used to calculate the probability of making a correct probability. It also refers to how much entropy has been removed from the dataset. Decision forests on the other hand are constructed by sampling different portions of the dataset to build multiple decision trees which results are eventually averaged. This is referred to as bootstrap aggregation where "bagging" takes its name from.

$$E = -\sum_{i=1}^{C} \quad pilog2pi \text{ - - - - - - (1)}$$

where E in equation (1) represent the information entropy in a in a dataset

pi is the probability of randomly picking an element of class i

C represent the number of classes present in the dataset

$$G = -\sum_{i=1}^{C} \quad p(i) * \big(1 - p(i)\big) \text{ - - - - - (2)}$$

Equation (2) represent the Gini impurity of classifying of a data point where C represent the number of classes in the data and pi represent the probability of picking a data point with class I.

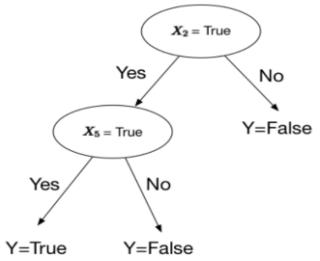


Figure 1: A decision tree method of operation.

Bagging is a machine learning ensemble method that averages the prediction of multiple single decision trees in an attempt to increase the prediction accuracy and metrics. This helps to reduce the bias and possible variance that could have been associated with a single decision tree and also prevent the variance that could have been associated with using gradient boosting decision trees when not properly regularized. Bagging algorithms have shown high accuracy on test datasets due to their ability to generalize well on unseen data sets. With sufficient numbers of trees, the chances of overfitting are greatly reduced. Tree based models have been put to use in the past for estimating petrophysical data. Vito et al., 2020 proposed a gradient boosting tree approach to resolving missing petrophysical data from well logs.

Machine Learning Ensemble Techniques

Ensemble method in machine learning allows for multiple weak (base) learners to improve the accuracy and reliability of machine learning models. Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a default or weighted vote of their predictions based on the classical bias–variance decomposition of the error. It has been shown that ensembles can reduce variance or both bias and variance (Matteo Re et al.,2012). While there is no unified theory underlying ensemble methods (Matteo

Re et al.,2012), they have been implemented and abstracted into algorithms to aid machine learning predictions and improve their accuracy. Ensemble techniques could be as simple as averaging the prediction of two different regressor algorithms in an attempt to get a more generalized result or building several base models with different subsets of the training data to using multiple algorithms with different hyperparameters to get diverse but similar results which a meta learner can train on to get even more efficient results. This is referred to as stacking.

DATA COLLECTION

Six different wells from the North Sea were used to build, optimize, and validate the proposed models. Two of the wells served as the test datasets while the other four were used simultaneously to create the training and validation data sets. The gamma ray, neutron porosity, density, resistivity, and Delta T (DT) logs were selected from the complete well logs suite to create the datasets. The choice of logs was based on occurrence in the majority of the wells with relatively little amount of missing values compared to other logs. The two wells (F15B and FT2) were chosen as the test datasets due to the unlogged DT in the log suite. Four training and validation datasets were created each from the other four wells (F1, F11A, F12, F15D). Each training dataset was created by using three of the well logs data while the remaining well served as the validation data set in which the trained model was evaluated on.

Exploratory Data Analysis

The primary aim of exploratory data analysis (EDA) is to examine the data for distribution, outliers, and anomalies to direct specific testing of the hypothesis (Komorowski et al.,2016). During EDA, the data is simply visualized, plotted, manipulated or transformed, without any assumptions, in order to help assess the quality of the data and building models (Komorowski et al.,2016). EDA helps to provide insights that can be useful during data modeling. When the distribution is skewed or the data structure obscures the pattern, the data could be rescaled in order to improve interpretability (Chong Ho Yu, 2010). The skewness of a data is the degree of distortion of the data from a normal distribution.

Bivariate Analysis

The bivariate analysis of the logs with respect to the target variable (DT log) is done to visualize the relationships and distributions between the feature (input) logs and the target log. This as well helps to capture information about outliers, and unreal and spurious log values. The combined logs instead of the individual logs were rather used for the bivariate analysis since this will be fed directly into the model.

Skewness of Target Variable (DT Log) in Each Validation Dataset

The degree of skewness of the target variable is from a normal distribution. This has the potential to affect the way the model is trained as it negates most algorithms' assumptions of a normal distribution for a regression problem This is similar to the class imbalance problem in the case of a classification problem. For a right-skewed target distribution, the model will be trained more on the portion of data that falls under the right part of the distribution and vice-versa for a left-skewed target distribution. This can also impair features importance analysis. Figure 5 shows the results of the skewness analysis of the training dataset when (a) Well F1 and (b) F11A are used for validation respectively.

The distribution plots of the validation targets all fall within range of a normal data distribution. In cases of a highly skewed distribution, the data is scaled down and used to train the model. Predictions made on the test set with the scaled model are then scaled back up using the inverse function used for downscaling the training target.

DATA PREPARATION

Missing values/sections in all datasets were removed to allow for a cleaner dataset devoid of biases from using other methods of dealing with missing values. A larger amount of these missing values account for unlogged data starting from most of the well tops. These parts were dropped off from both the training and validation datasets to allow the algorithms work on the dataset properly. Missing and infinite values are seen as ambiguous for algorithms like random forest and extra trees regressors.

Data Scaling and Normalization

The data was passed through a data scaling or normalization process. Normalization is a scaling technique or a mapping technique or a pre-processing stage where we can find a new range from an existing one range (Patro et al.,2015). Data normalization is the process of casting the data to the specific range, like between 0 and 1 or between -1 and +1 (Ali et al., 2014). This transforms the data into a normal distribution with a standard deviation of zero and a mean value of 1. Normalization is required when there are big differences in the ranges of different features (Ali et al.,2014). The effects of data scaling and normalization are largely seen on linear models like linear regressors and SVM. Tree based models like decision trees and forests have however been observed not to require scaling as the method of making predictions does not require any scaling. Neural networks on the other hand however require scaling and normalization of extreme values (outliers). Data scaling was done to reduce the magnitude of the petrophysical properties passing through the algorithms to speed up training time.

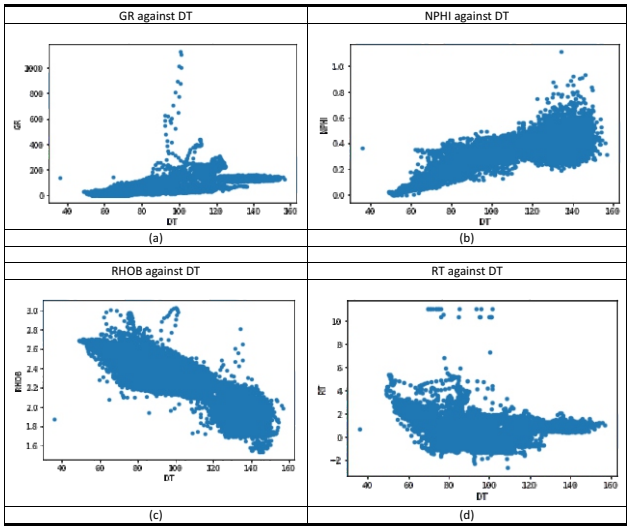


Figure: 2 (a - d) showing the petrophysical logs (features) against the training label (DT log) for Well F1.

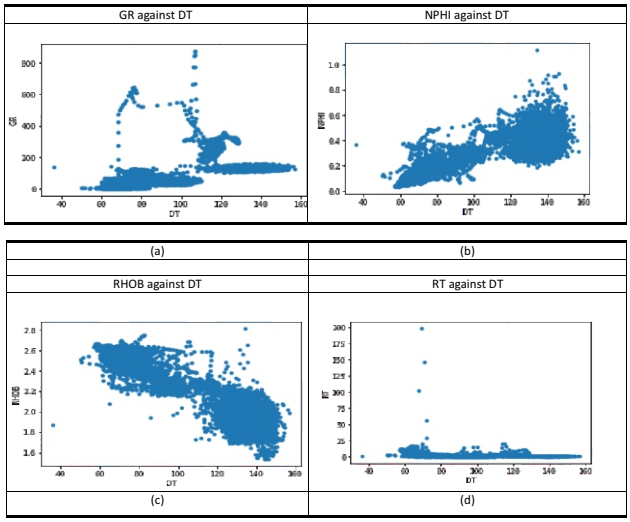


Figure 3: (a - d) showing the petrophysical logs (features) against the training label (DT log) for Well F11A.

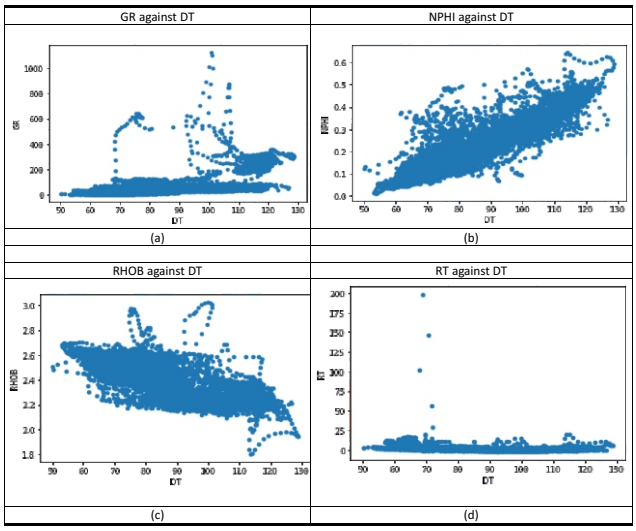


Figure 4: (a - d) showing the petrophysical logs (features) against the training label (DT log) for Well FT2.

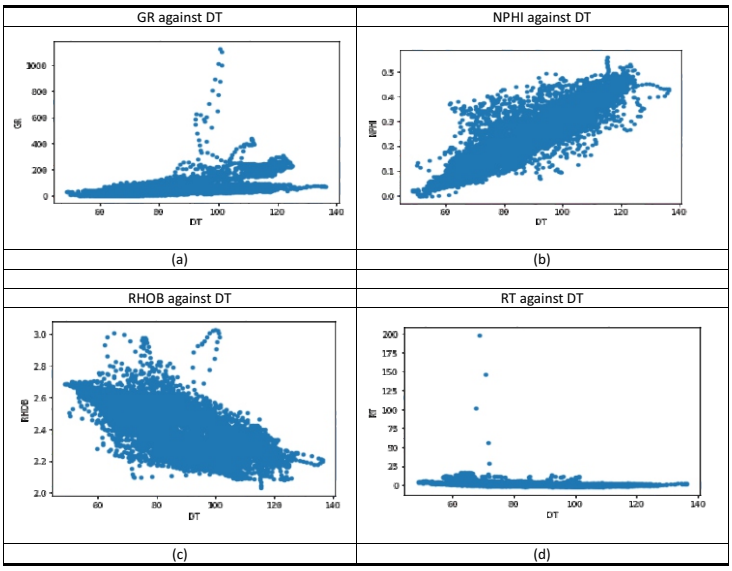
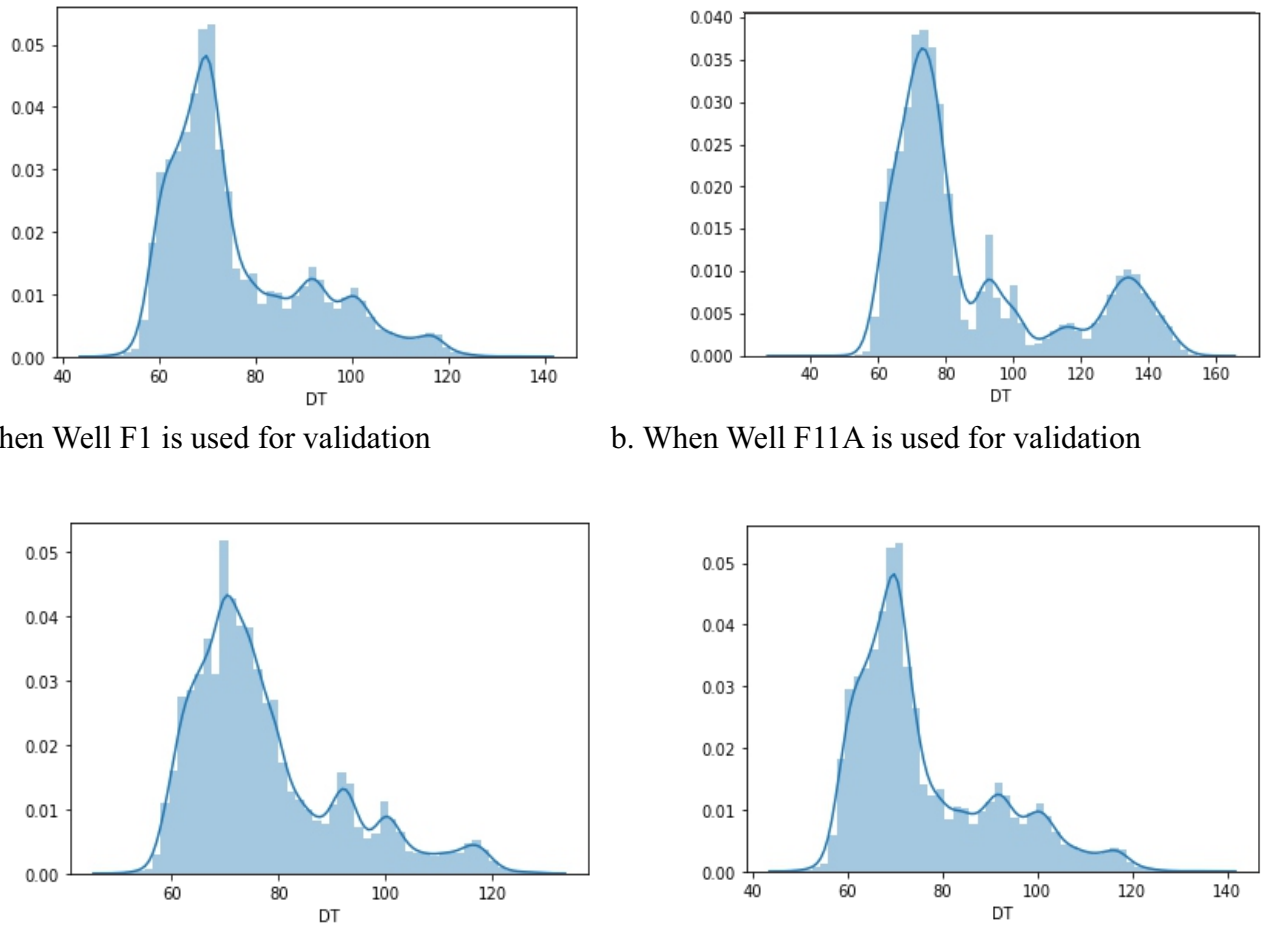


Figure 5: (a - d) showing the petrophysical logs (features) against the training label (DT log) for Well F12.



a. When Well F1 is used for validation

b. When Well F11A is used for validation

c. When Well FT2 is used for validation

d. When Well F12 is used for validation

Figure 6: (a) - (d) show the histogram distribution of the target log in wells F1, F11, F12, and FT2 respectively.

Table 1: Summary of Predictions

WELLS	Decision Tree		Random Forest		Extra Trees	
	RMSE	R^2 Score	RMSE	R^2 Score	RMSE	R^2 Score
F11A	6.795	0.809	6.474	0.827	4.873	0.902
F1	6.311	0.766	5.739	0.807	5.280	0.836
FT2	5.481	0.865	4.938	0.890	3.990	0.928
F12	20.826	0.446	20.723	0.452	19.992	0.490

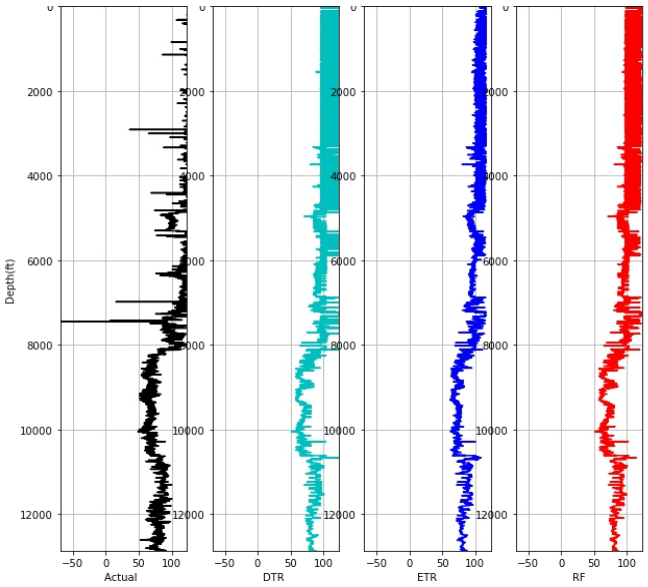


Figure 7: Showing the predicted DT logs from the three models for validation Well F12 against depth (m).

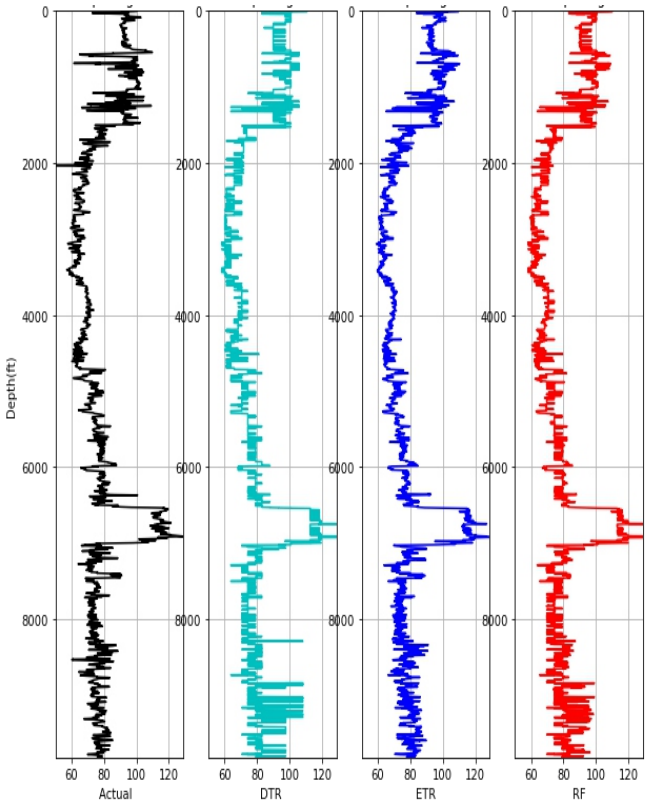


Figure 8: Showing the predicted DT logs from the three models for validation well F1A against depth (m).

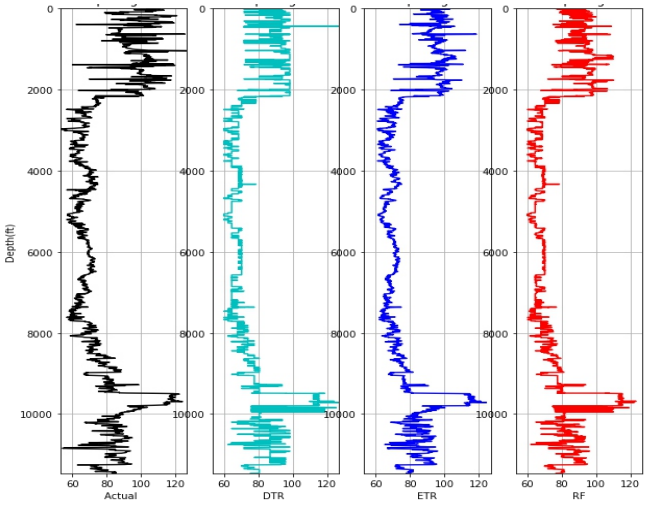


Figure 9: Showing the predicted DT logs from the three models for validation well F1A against depth (m).

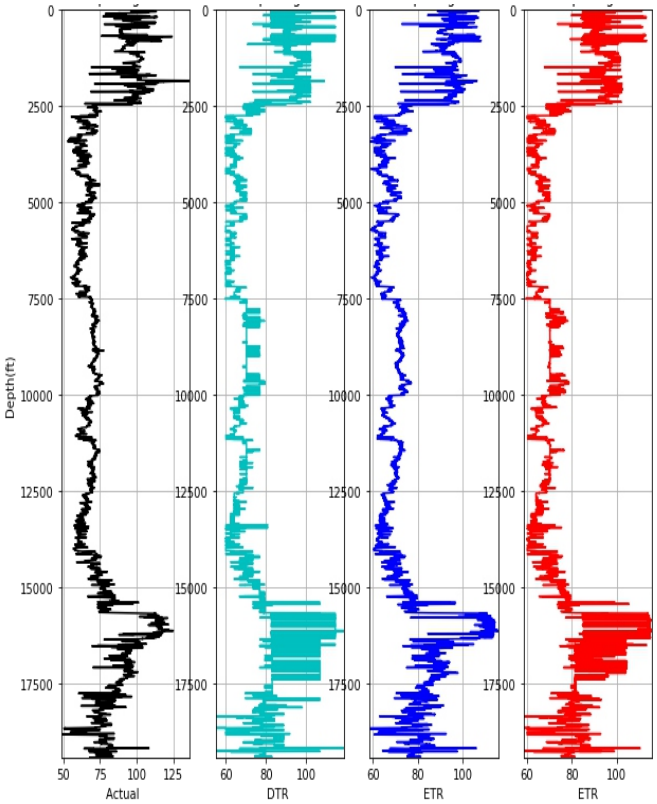


Figure 10: Showing the predicted DT logs from the three models for validation well F1A against depth (m).

The StandardScaler module from sklearn preprocessing package was used. It standardizes features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

$z = (x - u) / s$

where *u* is the mean of the training samples scores and s is the standard deviation of the training samples.

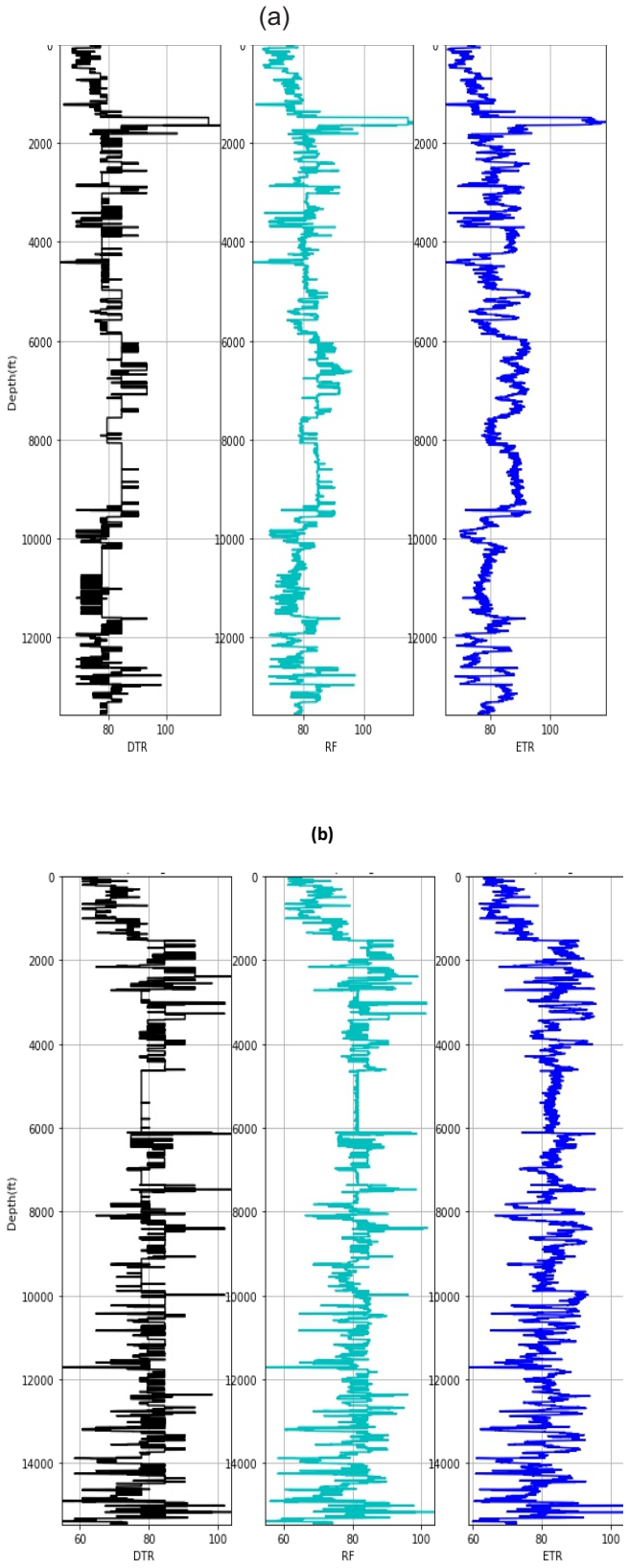


Figure 11: (a) and (b) Show the predicted DT logs from the three models for test wells F15 and F11B against depth (m).

MODEL OPTIMIZATION

Two bagging algorithms, a random forest and extra trees regressors and the conventional single decision tree regressor all from the Python scikit-learn library were used. A 10-fold cross-validation (CV) was used with each algorithm to get a more generalized result and to estimate the overall performance of the models on unseen data. This is also a data resampling technique. In k-fold cross-validation, the available training set is partitioned into k disjoint subsets of approximately equal size (Berrar 2018). The model is then trained with k-1 of the folds and evaluated on the last one.

Hyperparameter tuning with GridSearchCV was done to obtain optimal hyperparameters for the training of each model. It aims at finding a tuple of hyperparameters that yields an optimal model that minimizes a predefined loss function on a given independent data (Ghawi et al., 2019). A three-fold cross-validation was used to conduct a grid search on the hyperparameters. The number of estimators (for both proposed regressors), minimum split at each tree leaf node, minimum samples at each leaf node, minimum impurity split were tuned to obtain each model's optimal hyperparameters. A maximum depth of 6 was chosen at discretion for all three algorithms. This was done to prevent overfitting of the models on the train dataset. Overfitting maps out noise as interested relationships rather than the real data signature. The tree depths were chosen as 6. This is because the GridSearch hyperparameter tuning could only give a maximum depth that was fit for the training datasets. 400 trees were obtained as the optimal estimators for each bagging algorithm. Above 400 trees, model improvement was not significant.

MODEL PERFORMANCE MEASURES

The model performance was optimized and evaluated using the root mean square error (RMSE) and the R2 score (coefficient of determination). The RMSE is the standard deviation of all the prediction errors. It measures the deviation of the predicted values from the actual values. A very high value indicates a more error prone prediction while a lower value shows more prediction accuracy. R2 score is a measure of how fit the predicted values are to the actual values. R2 scores range between -1 to +1. A R2 score of +1 indicates a perfect positive fit while -1 indicates a perfect negative fit.

RESULTS & DISCUSSION

The comparative results of the performances using the optimized models to predict the missing values in the testing dataset are shown in Table 1.

Both bagging algorithms outperformed the single decision tree algorithm. In all validation cases however, the extra trees regressor outperformed the random forest regressor considering both the RMSE scores and

correlation coefficient scores. All three algorithms were used to make predictions on the test logs to reproduce the DT log which were missing in wells F11B and F15D. RMSE and R^2 validation scores serve as a guide into choosing the most accurate algorithm or model to use for replacing the missing logs. At depth interval 3050m - 3100m in Well F12, the NPHI, RHOB and RT logs recorded a missing section altogether. The extra Trees regressor was used in reproducing the missing section of the well.

PREDICTED LOGS COMPARISON

Comparative plots of the actual DT log and the predictions based on random forest and extra tree models for Wells F12, F11A, F1, T2, F15, and F11B are presented in Figures 6 through 10. The predicted DT values were plotted against their corresponding depths. A uniform x-axis scale was shown for all log types to properly display the variations and similarities among the curves produced. Spike readings on the actual log plots failed to appear on the prediction logs. The extra trees regressor log (ETR) showed a more distinct log signature compared to the other predicted logs; decision tree (DTR) and random forest (RF) logs. This is due to its better prediction accuracy hence showing more similarity to the actual log plot compared to the other two models.

CONCLUSION

Bagging models have proved to offer better prediction accuracy and generalization on unseen wells compared to single decision trees models. They can be used to save logging and drilling costs in the field by estimating log records from other recorded logs. It also offers the benefit of manpower and time conservation. Estimation of missing well logs will provide a better quality data leading to a better formation evaluation and reservoir characterization.

Machine learning models are not just dependent on how sophisticated the algorithms are but also how robust (in terms of quantity and quality) the training data fed into the model is. We suggested that the model performances can be improved through the following ways:

Using more logs from the wells.

Feeding the machine learning models with more data (feature generation and creation from existing data).

Calculating more petrophysical properties like porosity, permeability etc.

Feature selection: Checking for feature importance and dropping features likely to cause overfitting.

Feeding the models with better data (feature engineering).

Carrying out log transformations, trigonometric transformations, etc.

Stacking is done by making an ensemble of weak (base) learners with a meta algorithm to improve the predictions of machine learning models.

REFERENCES CITED

Al-Mudhafar W.J., Al-Mudhafar A., 2014. Comparative Statistical Algorithms for Imputation of Missing Measurements in Petrophysical Data.5 th Basra Oil & Gas International Conference and Exhibition (2014). Basra Iraq, 4 – 7 December 2014.

Berrar D., 2018. Cross-validation. Encyclopedia of Bioinformatics and Computational Biology, Volume 1, Elsevier, pp. 542–545. https://doi.org/10.1016/B978-0-12-809633-8.20349-X.

Breiman L., 1996. Bagging predictors. Machine Learning, 24(2):123–140, 1996.

Chijioke E., Emenike E., 2018. Using Deep Learning and Distributed Machine Learning Algorithms to Forecast Missing Well Log Data. 2018 Pacific Section AAPG Convention, Bakersfield, California, April 22-25, 2018.

Chon H.Y., 2010. Exploratory data analysis in the context of data mining and resampling. International Journal of Psychological Research, 3(1), 9-22.

Dietterich T.G., 2000. Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45014-9_1

Ghawi R., Pfeffe J., 2019. Efficient Hyperparameter Tuning with Grid Search for Text Categorization using KNN Approach with BM25 Similarity. Open Comput. Sci. 2019; 9:160-180.

Kumar M., Dasgupta R., Dip K.S., Singh N. P., 2018. Petrophysical evaluation of well log data and rock physics modeling for characterization of Eocene reservoir in Chandmari oil field of Assam-Arakan basin, India. J Petrol Explor Prod Technol (2018) 8:323–340.

Lam. L., Sue C., 1997. Application of majority voting to pattern recognition: an analysis of its behavior and performance. IEEE Transactions on Systems, Man and Cybernetics, 27(5):553–568, 1997.

Lopes R.L.,, Jorge A.M.,, 2017. Assessment of predictive learning methods for the completion of gaps in well log data. Journal of Petroleum Science and Engineering xxx (2017) 1–14.

Matthieu K., Dominic C. Marshall, Justin D.S., Yves C., 2016. Exploratory Data Analysis. Chapter 15 2016.

Mohammad A.S., Motafakkerfard R., Mohammad A.R., Siyamak M., Sabety N., 2014. Support vector machine method, a new technique for lithology prediction in an Iranian heterogeneous carbonate reservoir using petrophysical well logs. Carbonates and Evaporites 30(1):1-10 (2014).

Nordloh V.A., Roub´ıckov˘a´ A., Brown N., 2020. Machine Learning for Gas and Oil Exploration. 9th International Conference on Prestigious Applications of Intelligent Systems – PAIS@ECAI2020.

Re M., Valentini G., 2012. Ensemble methods: A review. (2012). page 5.

APPENDIX

WELL F1

	DEPTH	DT	GR	NPHI	RHOB	RT
count	9822.0000	9822.0000	9822.0000	9822.0000	9822.0000	9822.00
mean	3107.8500	77.4106	48.6247	0.1668	2.4809	2.7971
std	283.5511	13.0543	66.3325	0.0919	0.1358	3.5342
min	2616.8000	50.1782	1.2393	0.0332	1.8012	0.0933
25%	2862.3250	69.1022	11.1478	0.1081	2.4259	1.2510
50%	3107.8500	74.3363	34.5119	0.1449	2.5332	2.2546
75%	3353.3750	80.8687	53.7058	0.2034	2.5756	3.3496
max	3598.9000	128.7630	873.71940	0.6454	2.7486	197.7940

WELL F15D

	DEPTH	GR	NPHI	RHOB	RT
count	13566.0000	13566.0000	13566.0000	13566.0000	13566.0000
mean	3980.1500	36.7610	0.1848	2.3768	2275.3537
std	391.6311	28.8643	0.0542	0.1648	11331.0795
min	3301.9000	5.9773	0.0679	2.1319	0.2898
25%	3641.0250	20.2835	0.1521	2.2292	3.3612
50%	3980.1500	28.1454	0.1799	2.3296	6.3215
75%	4319.2750	46.7791	0.2005	2.5316	20.3874
max	4658.4000	269.9139	0.4843	3.0644	62290.7695

WELL F11B

	DEPTH	GR	NPHI	RHOB	RT
count	15466.0000	15466.0000	15466.0000	15466.0000	15466.0000
mean	3971.4500	33.9404	0.1833	2.3971	2255.3653
std	446.4793	19.7533	0.0590	0.1812	11450.4350
min	3198.2000	2.6870	0.0240	1.6270	0.1400
25%	3584.8250	17.9550	0.1550	2.2300	3.2672
50%	3971.4500	28.7230	0.1760	2.3740	7.8340
75%	4358.0750	43.5857	0.2090	2.5600	52.6257
max	4744.7000	123.3620	0.5410	3.0900	62290.7700

WELL F11A

	DEPTH	DT	GR	NPHI	RHOB	RT
count	11464.0000	11464.0000	11464.0000	11464.0000	11464.0000	11464.0000
mean	3150.1500	77.7336	32.9029	0.1657	2.4683	103.3138
std	330.9516	15.5498	51.0344	0.0996	0.1532	2334.4001
min	2577.0000	53.1650	0.8520	0.0100	2.0330	0.1030
25%	2863.5750	66.1840	8.5892	0.0930	2.3370	1.8230
50%	3150.1500	71.5790	19.7135	0.1300	2.5290	3.0810
75%	3436.7250	87.6747	38.3400	0.2230	2.5810	4.9732
max	3723.3000	126.8270	1124.4030	0.5590	3.0250	62290.7700

WELL FT2

	DEPTH	DT	GR	NPHI	RHOB	RT
count	19433.0000	19433.0000	19433.0000	19433.0000	19433.0000	19433.0000
mean	3549.8795	75.4962	32.9475	0.1533	2.4951	9.1746
std	561.8629	14.9014	45.2533	0.0981	0.1299	502.0105
min	2577.0000	48.9280	0.8380	-0.0030	2.0900	0.0710
25%	3062.8000	65.4020	8.4350	0.0780	2.4600	1.9270
50%	3550.7000	70.6800	13.1590	0.1280	2.5290	3.0500
75%	4036.5000	82.8270	33.6110	0.2030	2.5840	5.7780
max	4522.3000	136.2530	437.8230	0.4860	3.0040	62253.9570

WELL F12

	DEPTH	DT	GR	NPHI	RHOB	RT
count	12860.0000	12860.0000	12860.0000	12860.0000	12860.0000	12860.0000
mean	2404.3192	106.7495	90.1551	0.2788	2.1224	33.4969
std	604.8413	27.9914	42.2897	0.1420	0.2275	327.1620
min	1368.7044	-68.8277	12.7895	0.0282	1.3568	0.1000
25%	1869.1479	79.5656	49.4770	0.1827	1.9388	0.5898
50%	2408.3010	115.4020	106.0727	0.2631	2.1211	0.8917
75%	2919.8697	131.9252	126.9516	0.3716	2.2978	2.4949
max	3442.5636	161.0148	179.8782	1.1138	2.9392	5000.0000