Journal of Applied Geophysics 212 (2023) 104992

Contents lists available at ScienceDirect

Journal of Applied Geophysics

journal homepage: www.elsevier.com/locate/jappgeo

Using AI tools to fill an incomplete well log dataset: A workflow

H. Uzkeda [a,b,*], O. Vidal-Royo [a], A. Amilibia [a]

[a] Terractiva Consulting SL, Barcelona, Spain
[b] Departamento de Geodinámica, Estratigrafía y Paleontología, Universidad Complutense de Madrid, Madrid, Spain

ARTICLE INFO

Keywords:
Artificial intelligence
Petrophysical properties
Supervised learning
Well logs

ABSTRACT

One issue commonly found when working with well log data is the irregular abundance/availability of the different recorded parameters. This is especially applicable when working with datasets collected in different campaigns that may span through the years, even decades, or different companies. Artificial Intelligence may be useful to fill gaps in the original database, resulting in a more complete, standardised one. In this work we present a workflow that can be followed to fills gaps in a dataset using different AI techniques. It consists of four main steps: 1) feature combination selection; 2) hyperparameter tuning; 3) performance assessment and best option choice; 4) blind testing. The process can be performed iteratively, successively populating the database with missing parameters, starting with those for which there are more available training data and whose results are more reliable. In this work, we present an example in which we filled an incomplete dataset consisting of wells provided by the UK National Data Repository (NDR) of the Oil & Gas Authority (OGA). The performance of some of the most commonly used artificial intelligence methods (support vector machine, random forest, multi-layer perceptron) was tested varying their hyperparameters until reaching an adequate result.

1. Introduction

One common issue that Earth Scientists face when working with well log data is that these datasets tend to be incomplete; some parameters are extensively recorded, whereas others are only partially present in a few wells. In this manner, Artificial Intelligence (AI) techniques may be helpful to fill such gaps in a reliable and quantitative way. AI algorithms have been successfully applied to well log data in the geosciences for a variety of tasks, including: lithology and facies classification (Dubois et al., 2007; Tschannen et al., 2017; Imamverdiyev and Sukhostat, 2019; Sahu et al., 2019; Kumar et al., 2022), stratigraphic correlation (Wedge et al., 2019; Tokpanov et al., 2020), coal quality estimation (Zhou and O'Brien, 2016), fracture density (Zazoun, 2013), ore presence (Caté et al., 2017), or for the estimation of interesting parameters (Goutorbe et al., 2006; Konaté et al., 2015; Ważny et al., 2021). However, these studies tend to be focused on performing a certain task with the already available data and tend not to consider the issues of handling incomplete datasets. Commonly, they are restricted to well-studied areas and work only with those logs that are systematically recorded, which, as consequence, notably limits their applicability. Because of this, we present in this work a workflow thought to work with less-than-ideal datasets, aimed at filling gaps using different AI techniques. This workflow could be contemplated as an initial step for a larger-scale project, given that it may allow to have starting conditions closer to the ideal ones, i.e., full dataset.

It consists of four main steps, three for the training/validation and a last one for checking the results (Fig. 1): 1) feature combination selection; 2) hyperparameter tuning; 3) performance assessment and best option choice; 4) blind testing.

The first stage involves choosing the feature combination that will be used in the next steps to calculate the missing parameter. Then, the selected features are fed into different AI methods varying their hyperparameters for training and validation. During the third stage, we carry out a performance assessment of the results obtained in the previous phase to try to find the most suitable combination for the prediction of the missing well log(s). Finally, in the fourth stage we apply the best combination to the wells saved for blind testing in order to assess the ability of the trained regressors to work with data not used during the training validation process. After this last step, provided the results are acceptable, the regressor can be confidently applied to estimate the missing data.

Given that AI works with any number of input features, there is no need to restrict the selection to a group of parameters that seemingly have more control over the results (e.g., high porosity tends to be linked

* Corresponding author.
E-mail addresses: huzkeda@ucm.es, hodei@terractiva.net (H. Uzkeda), oskar@terractiva.net (O. Vidal-Royo), alejandro@terractiva.net (A. Amilibia).

https://doi.org/10.1016/j.jappgeo.2023.104992
Received 28 September 2022; Received in revised form 20 January 2023; Accepted 1 March 2023
Available online 5 March 2023

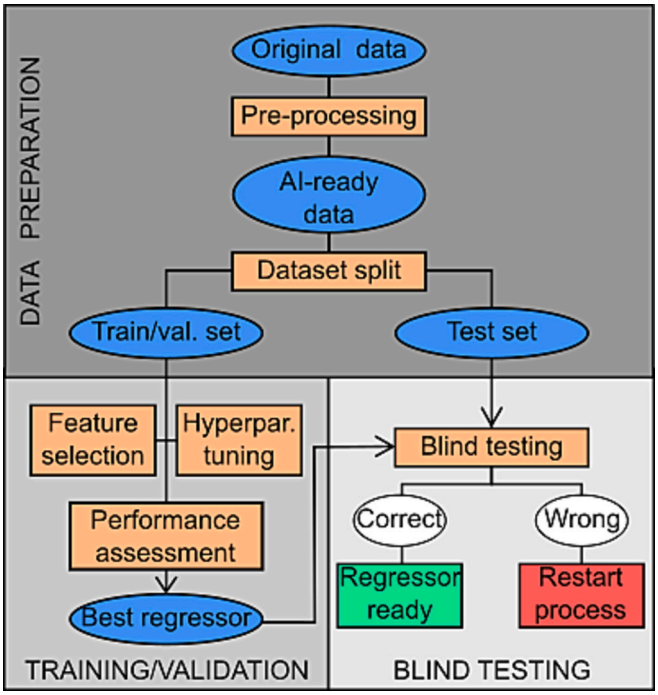


Fig. 1. Workflow followed in this work.

to lower density). Thus, no previous knowledge of the correlation between different attributes recorded in the well logs is required (although it may help, as the geoscientist's experience could narrow the grid search down to the best features). For this work we used the tools from the scikit-learn libraries (Pedregosa et al., 2011) implemented in Python (Python 3.8, ran in the Spyder software package). More precisely, we employed the *svm* (Support Vector Machine), *RandomForestRegressor* and *MLPRegressor* functions from the library mentioned above to carry out the training and prediction of missing values, as well as auxiliary ones during the data preparation (*MinMaxScaler*), results evaluation (*r2_score*, *np.polyfit*) and representation (*matplotlib.hist, matplotlib.scatter, matplotlib.plot*).

2. Dataset

In this work we used free, public data kindly provided by the Oil & Gas Authority (OGA, 2022) through the UK National Data Repository (NDR). The dataset consisted of 44 wells from quadrants 15, 16, 22, 30, and 31 (Fig. 2 and Table 1). These wells were drilled by seven different companies (Chevron, Harbour Energy, Hess, Ithaca Energy, Repsol, Shell, and Waldorf) between 1974 and 2017. The two wells from quadrant 30 were kept apart from the rest during the training/validation process and were saved to be used later during the blind testing stage.

2.1. Training/validation set preparation and analysis

We selected 42 of the 44 original wells, those from quadrants 15, 16, 22, and 31, to constitute the training/validation set (Table 1).

The well log data was compiled and converted into an AI-ready matrix of >800,000 rows. Because of source disparity, a consequence of having wells drilled by different companies and over a timespan of more than forty years, it was necessary to merge attributes recorded under different names (Table 2). Instead of working with all the available features (more than three hundred), we filtered the well logs, selecting only those attributes more frequently used and recorded in a reasonable number of observations, around 30% of the total number of observations. The selected attributes were: deep resistivity, density, gamma ray, medium resistivity, neutron porosity, shallow resistivity, and sonic DT.

We performed a statistical analysis before starting the regression process to assess the typical value ranges for the different attributes and to determine whether some sort of regularization could be required (Table 3). Additionally, we generated crossplots of the different attributes to make a visual assessment of the outliers (Fig. 3). These graphs are also helpful to detect data that required a certain conversion (e.g., neutron porosity may be given as a percentage (%) or as a fraction of one (*V*/V)). In this case, the percentages, present in four wells, were divided by 100 to have all the data in the same units. A similar situation was found with density: one well (16/26–23) it was expressed as kg/m^3, whereas the rest used g/cm^3. Regarding density, another well (31/26a-16) presented problems with empty cells that were interpreted incorrectly as actual, valid data. This situation was a result of inconsistency between the header and the data in the original LAS file. The script employed for data merging creates a report that informs on the attributes recorded in each well, as well as their units and description, hence facilitating the process of finding and solving inconsistencies. We performed an additional data filtering by removing the 0.5–1.0 (depending on the feature under consideration) upper and lower percentiles to avoid introducing extreme values that might impede training of the regressors. It is important to work with datasets that are as clean as possible, as any data introduced in the training will have an impact on the predictions being performed later. Table 4 and Fig. 4 illustrate a clean dataset, with all input parameters tidy, filtered and ready to start with the prediction of missing well logs. Furthermore, to avoid issues with features having excessively different value ranges (i.e., sonic DT might be over 200, whereas maximum neutron porosity is around 1), we applied a min-max scaler to bring all the features to within a [1−10] range, an order of magnitude similar to that of the features to be predicted (e.g., density, ranging approximately from 1 to 4). Because of the particular distribution of the resistivity values, we calculated their logarithms before applying the scaler. The scaler applied to the training/validation set was saved and applied later to the blind test set to ensure consistency in both the workflow and the results.

2.2. Blind test set preparation and analysis

Two wells from quadrant 30 (13–7 and 16–15) were selected to act as a blind test set after finding the most optimal feature and hyperparameter combination. The original data were transformed and made AI-ready following the same procedure described above. Fig. 5 shows the blind test data as initially received, whereas Fig. 6 corresponds to the crossplots once the dataset has been cleaned.

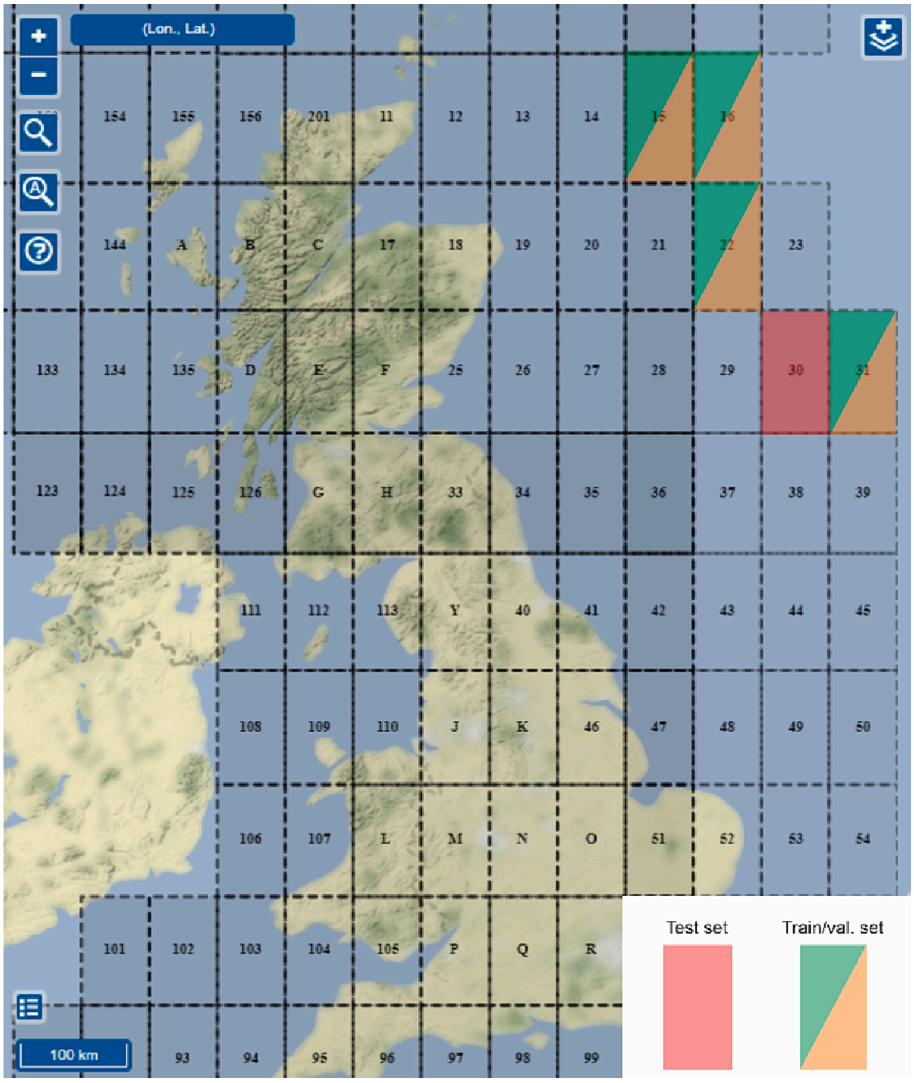


Fig. 2. Map illustrating the quadrants from which the wells were taken (modified from OGA, 2022).

Table 1
List of wells used; those from quadrant 30 were reserved for the blind testing.

Quadrant	Wells
15	29A-11, 30–2, 30–5, 30–6, 30–8, 30–9, 30–10, 30–12
16	26–2, 26–10, 26–11, 26–12, 26–13, 26–14, 26-14A, 26–16, 26–17, 26–19, 26–22, 26–23, 26–24, 26–25, 26–26, 26–27
22	30B-A14, 30B-A16
30	13–7, 16–15
31	26–3, 26a-A11, 26a-A12Z, 26a-A07, 26a-A09Z, 26a-11Z, 26a-16, 26b-18, 26c-13

Table 2
Feature equivalents.

Feature	Well logs
Deep resistivity	RESD, RESDEP, RES_DEEP, RIPD, ILD, LLDE, LLD
Density	DEEN, RHOB, RHOZ, RHOE
Depth	DEP, DEPT, DEPTH
Gamma ray	GR, GREC
Medium resistivity	RIPM, ILM, RESMED, RES_MED
Neutron porosity	NHPHI, NEU, NPHE
Shallow resistivity	RESS, RESSLW, RES_SLW, LLSE, LLS
Sonic DT	DT, SONIC_DT

Table 3
Descriptive statistics for the training/validation set before cleaning the dataset (outlier removal, error correction, extreme value exclusion).

	Depth	Deep res.	Density	Gamma ray	Medium res.	Neutron por.	Shallow res.	Sonic DT
Count	800,339	551,339	391,029	710,978	479,439	399,964	76,308	462,065
Mean	7126.602	3.560	−16.461	53.692	3.830	5.433	3.546	113.375
Std	4287.284	56.392	237.829	27.126	52.696	15.707	6.824	32.829
Min	0.000	0.000	−999.250	0.000	0.103	−999.250	0.180	−601.867
P25	3723.000	0.658	2.142	33.250	0.725	0.189	0.512	83.322
Median	6553.000	1.027	2.261	51.167	1.063	0.393	0.970	124.400
P75	10,123.000	1.907	2.534	70.666	2.036	0.470	3.103	140.250
Max	22,617.500	9993.018	2675.470	536.792	9993.018	121.080	151.692	1513.450

3. Regressor training

As explained above, this process consists of four stages: feature combination selection, hyperparameter tuning, performance assessment and blind testing. The procedure is described below taking density as an example of a predicted well log.

3.1. Feature combination selection

With 7 input features there are 127 possible combinations (7 × 1 feature; 21 × 2 features; 35 × 3 features; 35 × 4 features; 21 × 5 features; 7 × 6 features; 1 × 7 features). Although it would have been possible to use all of them, we filtered the data to reduce number of calculations and to speed up the process. First, we discarded regressions that used only one input feature as they would only work when there is a simple relation between the input and the estimated feature—such a scenario would be the case for resistivities (see Figs. 4 and 6 where the linear correlation between the three resistivities can be easily noticed). In addition, because of the varying availability of the input features, there were combinations that could not be used for either training/validation and testing nor for prediction, as there were not enough available observations. As a consequence, we selected only those combinations that could be trained and validated using at least 30% of the total observations. Additionally, because of the high correlation between the different resistivities (Fig. 7), combinations using two or more resistivities were discarded, hence avoiding issues derived from introducing redundant data and further reducing the number of combinations. In the end, we selected a total of 18 combinations for running the hyperparameter tuning, an easier number to handle than the original list of 127 (Table 5).

If the number of feature combinations had remained too large, making the following steps excessively time-consuming, an additional strategy could have been employed. Under these situations, we recommend to use a relatively simple, shallow neural network (i.e., only one hidden layer with 64 neurons or similar configuration) that can be trained and whose results may be validated fairly quickly to check which feature combinations seem to work better. Then, for example, the best 25 can be chosen to proceed with hyperparameter tuning. This technique should be used carefully and only when testing all the existing possibilities is not feasible because of the computation time it would imply, as some appropriate combinations might be overlooked, preventing getting to the optimal prediction.

3.2. Hyperparameter tuning

In this work we opted for comparing three different AI methods: support vector regressor, random forests and neural networks (multi-layer perceptron regressor) as provided by scikit-learn.

The support vector regressor (SVR) is a method proposed by Drucker et al. (1997) based on the support vector machine originally developed for classification problems by finding the best separating hyperplane, decision boundary (Vapnik, 1995). In the case of regression, the hyperplane is the n- dimensional surface containing the maximum number of points within a threshold value (epsilon). We tried different hyperparameter combinations until we found an acceptable balance between fitting and generalization (Table 6).

Random forests (RF) were proposed by Ho (1995) as a classification method based on combining several decision trees. The use of more than one tree makes it possible to overcome one of the main limitations of the decision trees, which is to reduce their overfitting tendency when their complexity grows. When facing regression problems, the values calculated by each individual tree are averaged to obtain the final result. As with the SVR, we tested diverse hyperparameters (such as the maximum depth for each tree or the number of decision trees to use in parallel) in order to find the best combination for our study (Table 6).

The third technique we evaluated was neural networks (multi-layer perceptron regressor, MLPR), a series of connected layers formed by neurons whose weights are iteratively adjusted until reaching an acceptable solution. We tested three different architectures varying the number of hidden layers (6, 9, and 12) and the number of neurons by layer (128, 64, and 32, respectively) (Table 6).

Three search grids were created (one for RF, one for SVR, and one for MLPR), varying some of the hyperparameters in order to find the best predictor for density in our dataset (Table 6). A total of 112 regressors (32 RF, 32 SVR, and 48 MLPR) were trained and validated with the 18 previously selected input feature combinations. We used k-fold cross validation, with each validation set formed by 4 wells (Fig. 8). k oscillated between 5 and 7 depending on the feature combination and the corresponding data availability, given that certain wells did not have some logs recorded.

3.3. Performance assessment

To evaluate the performance of the hyperparameter/feature combinations and select the best one, we created predicted versus observed

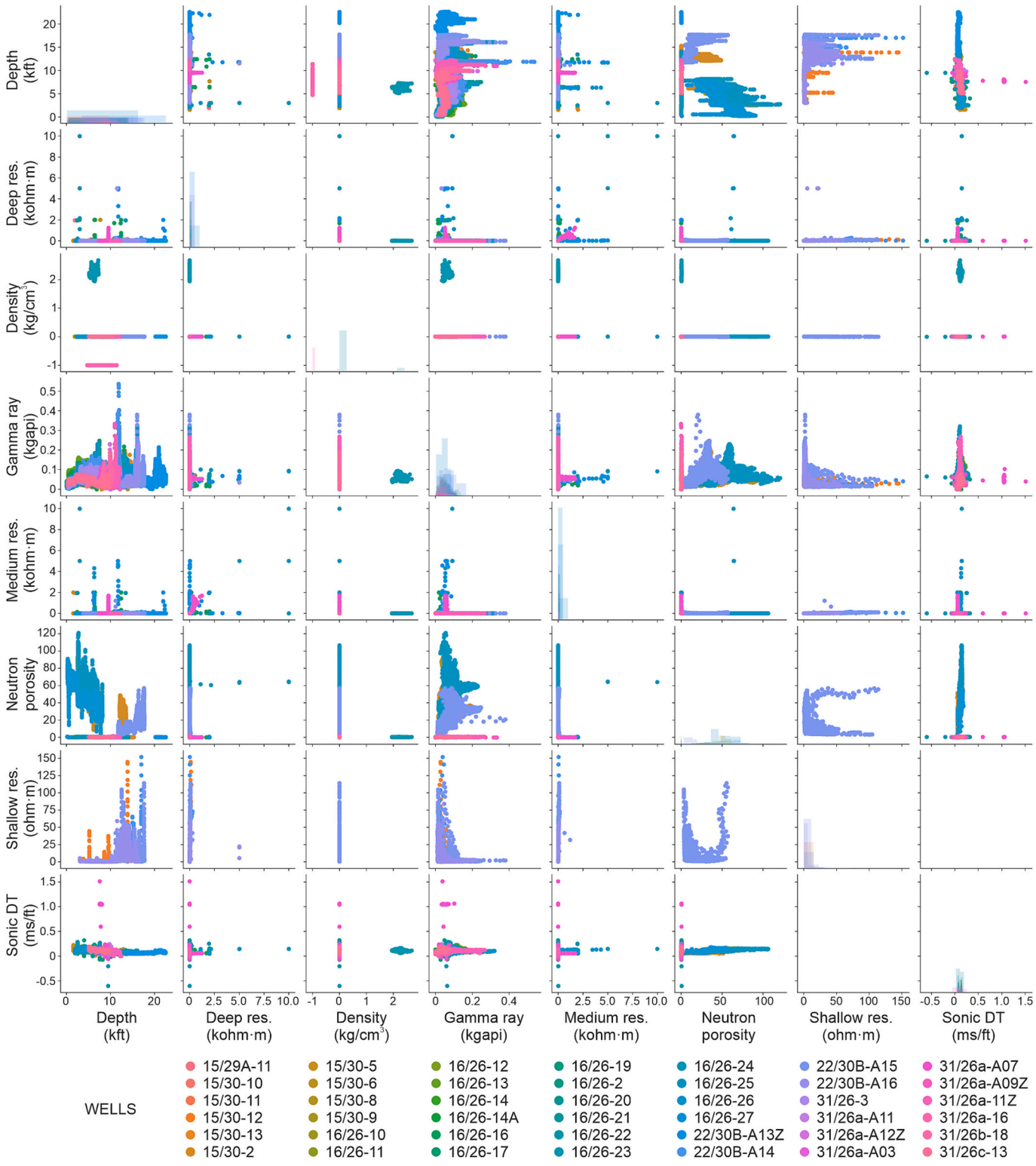


Fig. 3. Crossplots of selected well logs before tidying the training/validation dataset.

Table 4
Descriptive statistics for the training/validation set after cleaning the dataset (including log conversion of resistivities and removal of observations without any of the attributes considered).

	Depth	Deep res.	Density	Gamma ray	Medium res.	Neutron por.	Shallow res.	Sonic DT
Count	800,339	548,582	378,859	703,868	479,439	370,917	76,308	457,444
Mean	7126.602	0.551	2.305	53.615	0.583	0.336	0.550	113.349
Std	4287.284	1.751	0.231	26.741	1.722	0.158	0.834	32.000
Min	0.000	−1.481	1.083	0.233	−0.987	−0.345	−0.745	40.067
P25	3723.000	−0.182	2.154	33.264	−0.140	0.209	−0.291	83.400
Median	6553.000	0.012	2.267	51.162	0.027	0.386	−0.013	124.422
P75	10,123.000	0.280	2.537	70.633	0.309	0.454	0.492	140.250
Max	22,617.500	4.000	3.948	212.930	4.000	0.994	2.181	214.973

density plots for each run and, using a linear fit, extracted the corresponding slope and dispersion (R^2) (Fig. 9). Then, we calculated the average of these values for the different folds used during cross validation. In order to be able to compare the different options using only one parameter, we employed the harmonic mean of the slope and the R^2. This would be an approach similar to using the typical F1 score for classification problems that combines precision and recall at the same time. The reason for using the harmonic and not the arithmetic mean is that we want to find the best combination but with a balanced distribution of precision (R^2) and accuracy (slope). The harmonic mean helps to discard options with very low precision and high accuracy, or, low accuracy and high precision, and to identify those options with a better balance between the two primary performance indicators (Fig. 10). Before calculating this performance estimator, a transformation of the slope value may be needed if the slope is above 1 (e.g., consistently overestimated predictions). If this is the case, instead of directly using the slope, which may lead to a deceptively high value of the score; the harmonic mean should be calculated between the R^2 and the transformed slope (1-(Slope-1)). For instance, a result whose slope is 1.3 will have a transformed slope of 0.7, indicating that it is 0.3 units from a perfect fit (Fig. 11).

Among the three methods, the best results were obtained with MLPR (Table 7 displays the top 10 ranked by their pseudo-F1 score). Looking at these results, it is easy to spot that the main controlling factor is the feature combinations (IF in Tables 6 and 7). In most cases, the same input features in different network architectures lead to quite similar results. For instance, three of the four best combinations are obtained with features combination (IF) 13 (Depth, Sonic DT, Medium resistivity) (Table 7). Considering this, although more hidden layer configurations could have been tried, the validity of the fitting would probably not have improved significantly. Thus, it might be more useful to check as many input features as possible instead of as many architectures as possible.

It is also worth checking the difference between the variance (error for the validation set) and the bias (the training error should be lower than the variance). A notably higher variance than bias is usually indicative of overfitting to the training set. This can usually be solved with feature engineering (i.e., regularizing the input features), changing the hyperparameters (especially in a situation where a systematic search of the best combination with a grid is not used) and/or employing larger training sets with more data variability. Fortunately, in our case, there are no substantial differences between the training pseudo-F1 scores and the validation ones, so no action was required.

Instead of selecting just one set for the blind testing and, had it offered acceptable results, using it to make predictions and to fill empty observations, we took the best (IF 13) and third best (which had gamma ray added to the input features; IF 17: Depth, Gamma ray, Sonic DT, Medium resistivity) combinations given that their scores were very similar (F1 for IF 17 was 0.919 versus 0.926 for IF 13; see Table 7).

3.4. Blind testing

After finding the theoretically ideal combination through successive training/validation runs, we applied it to the blind test set (two wells from quadrant 30) that were left out during the training process. Because these wells were not used during the training/validation, they can be taken as a reference for the applicability of the regressors to other areas or datasets.

Significantly higher errors may arise during blind testing than during the validation phase. They are usually a consequence of not having a validation set reflecting the variability that the regressor faces when applied to data not seen during the training/validation process. The best way to deal with this issue is to use a larger training/validation set with a similar distribution of observations to those that are to be predicted.

The regressors were trained with the whole set used before and applied to the test set in order to make predictions. When the best hyperparameter/feature combination pair was found, predicted versus observed plots were created and linearly fitted to get the slope and dispersion (e.g., Fig. 12 illustrates the results for the combination with highest pseudo-F1 score during the validation). In addition, we calculated the absolute errors for each observation in order to check the maximum, average and normalized values (Table 8).

Attending to the errors of the best combination (first row of Table 8), in comparison to the values obtained during the validation process, the best combination shows slightly greater errors (the mean absolute error grows from 0.044 for the validation to 0.094, ≈2% to ≈4% normalized values) and lower pseudo-F1 score (0.926 to 0.793). The latter is a consequence of a higher dispersion ($R^2 = 0.675$), given that the slope is close to 1 (0.961). Most of the identified issues arise from observations in well 30/13–7 at depths ranging from 4553.0 to 5674.5 ft. (~1388 to 17,296 m, denoted by a green circle in Fig. 12a and 13). This can be seen in the analysis where it is disaggregated by depth as shown in Table 9. Whereas observations out of this range have a mean error of 0.069

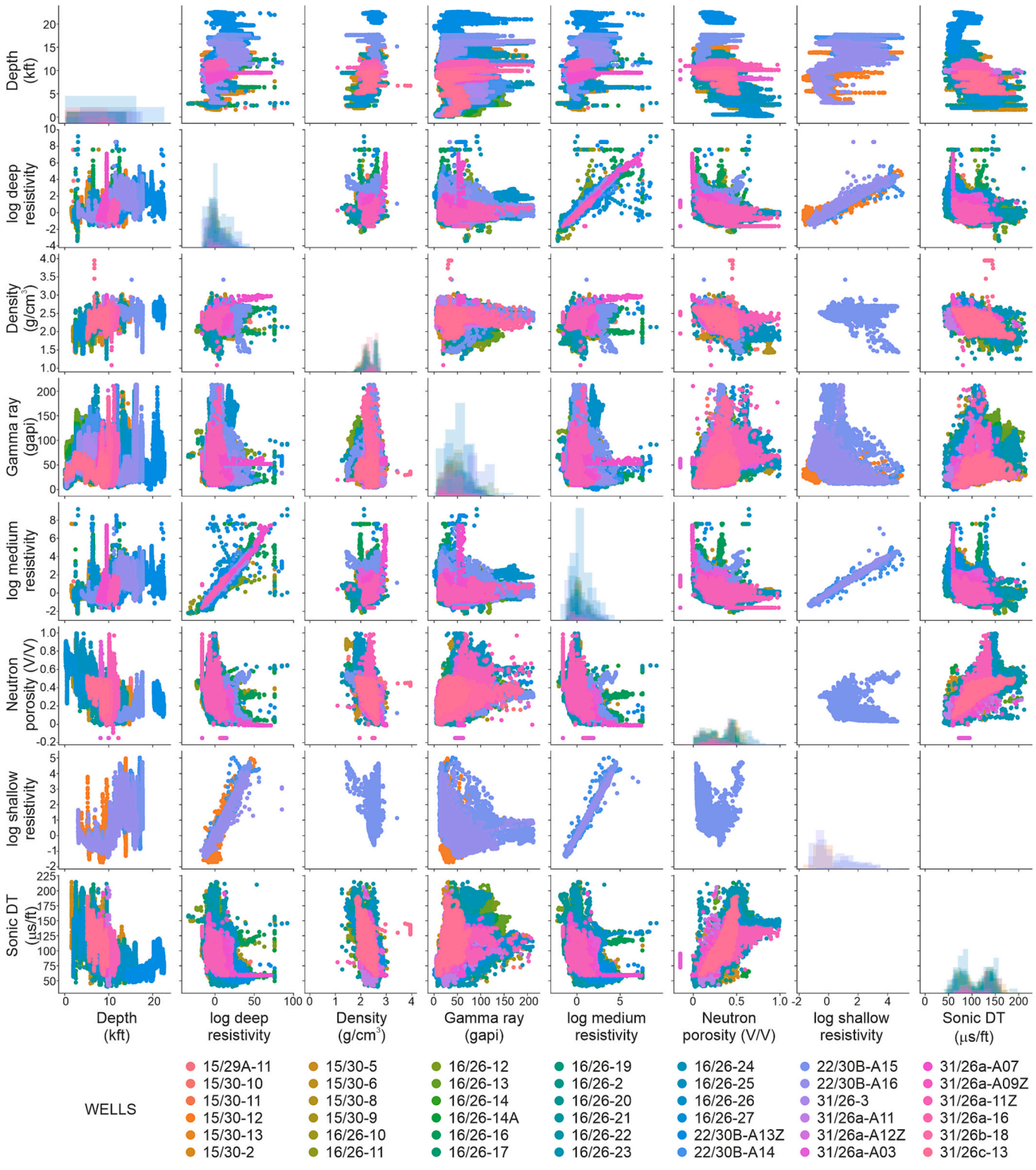


Fig. 4. Crossplots of selected well logs of the training/validation set after outlier removal, unit correction, logarithm calculations of resistivities.

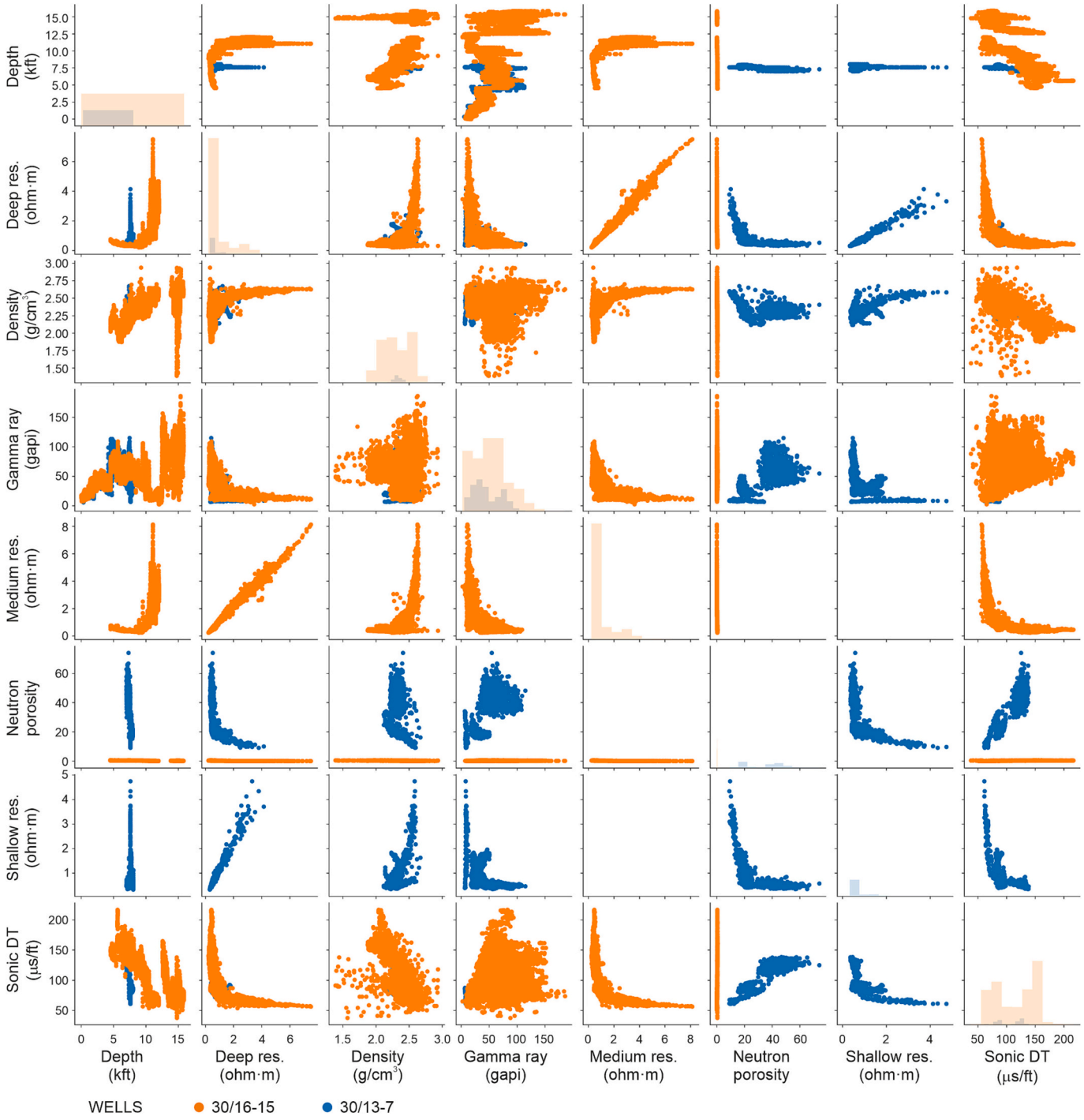


Fig. 5. Crossplots of selected well logs before tidying the blind test dataset.

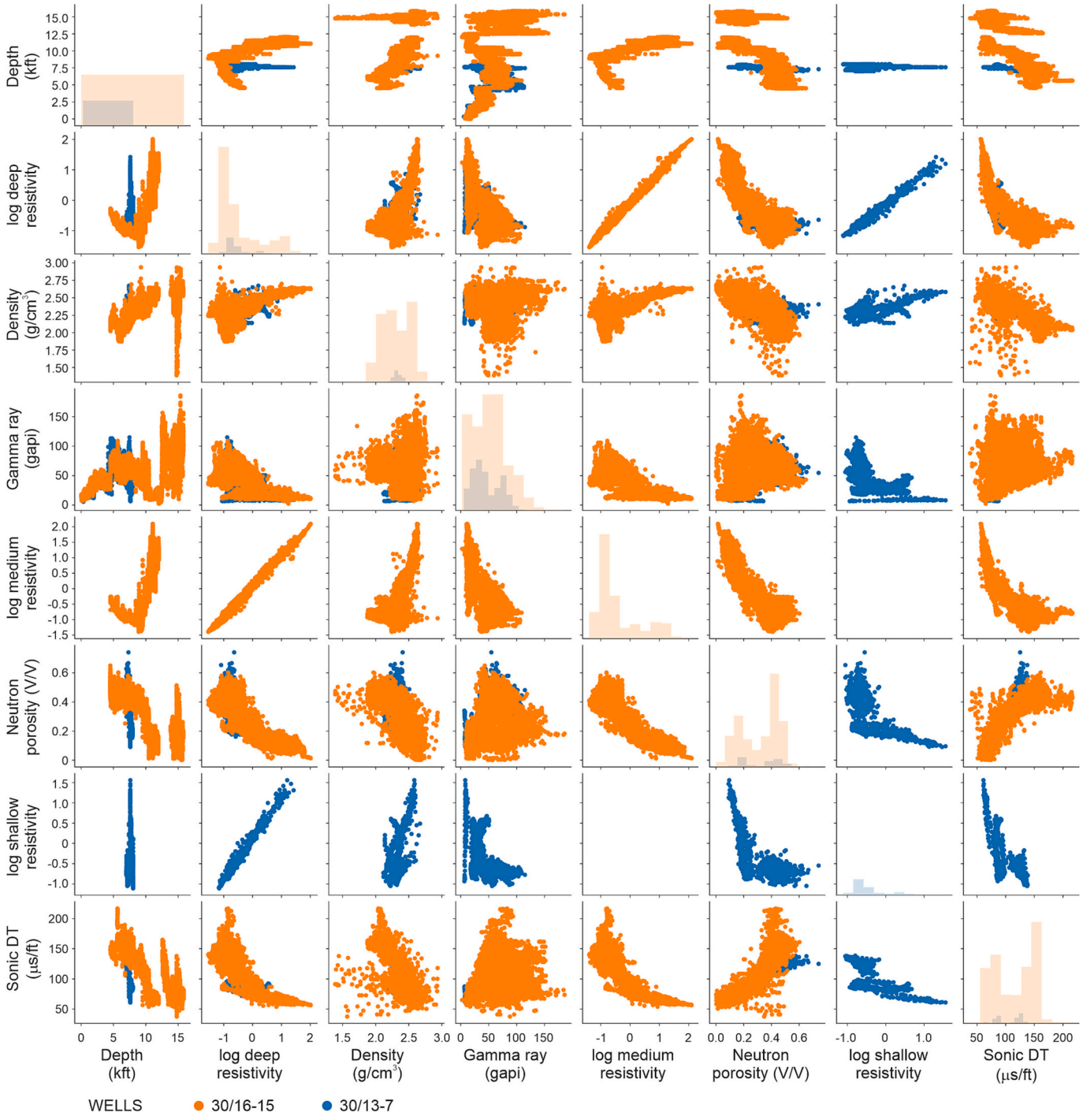


Fig. 6. Crossplots of selected well logs of the blind test dataset after outlier removal, unit correction, logarithm calculations of resistivities.

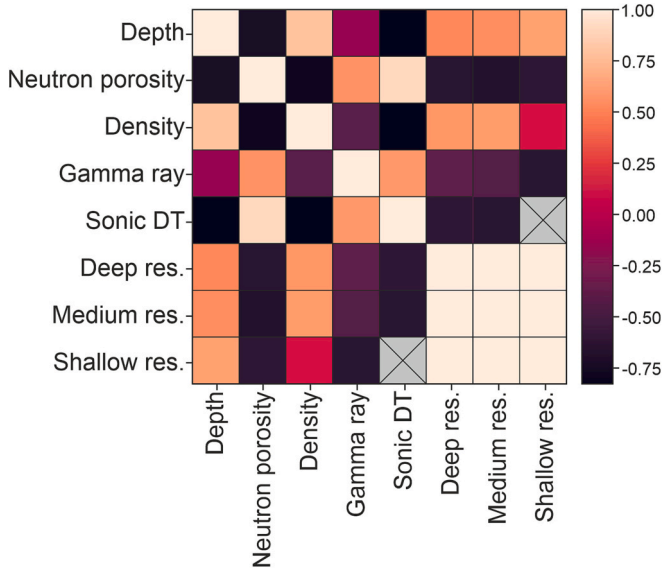


Fig. 7. Feature correlation heat-map.

(normalized value of ≈3.0%), those within the problematic region display an average absolute error of 0.249 (≈10.8%). This situation can be due to the presence of a stratigraphic unit or geological body with anomalous characteristics that prevent the regressor from working as expected. This is one of the main limitations of the method: the trained regressor would only work properly under conditions similar to those of the training process. The inconsistently larger error in well 30/13–7 corresponds approximately to the first 900 ft. of Cenozoic succession above the Mid Miocene unconformity and the underlying 200 ft. (Hart et al., 1997). The well report describes an intensely fractured zone towards the lower part of this succession and a rapid increment of the pore pressure from ≈4500 to ≈6000 ft. These circumstances could have also hindered the logging, making it less reliable and, consequently, impeding the regressor to make predictions as accurately as for the rest of the recorded section. Despite this, the overall fitting between the predicted and observed values was good enough, making this regressor applicable with enough confidence.

3.5. Regressor application

The blind testing results showed that the trained regressor was able to adequately predict density, even for wells outside the quadrants used for training and validation. Consequently, the regressor was successfully

Table 5
Feature combinations tested for Density estimation showing the number of observations that can be used for the training/validation, blind testing and that may be predicted.

	Feature combination for density prediction	Train/Val.	Test	Pred.
1	Depth, Gamma ray	330,168	20,763	406,386
2	Depth, Sonic DT	326,272	20,771	137,866
3	Depth, Deep resistivity	316,644	16,850	234,939
4	Depth, Medium resistivity	304,512	14,870	175,042
5	Depth, Gamma ray, Deep resistivity	292,917	16,800	220,262
6	Gamma ray, Deep resistivity	292,917	16,800	220,262
7	Depth, Gamma ray, Sonic DT	292,798	20,721	136,937
8	Gamma ray, Sonic DT	292,798	20,721	136,937
9	Depth, Sonic Dt, Deep resistivity	291,990	16,814	134,613
10	Sonic DT, Deep resistivity	291,990	16,814	134,613
11	Depth, gamma ray, Medium resistivity	280,863	14,870	160,673
12	Gamma ray, Medium resistivity	280,863	14,870	160,673
13	Depth, Sonic DT, Medium resistivity	279,906	14,834	105,956
14	Sonic DT, Medium resistivity	279,906	14,834	105,956
15	Depth, Gamma ray, Sonic DT, Deep resistivity	269,505	16,764	133,938
16	Gamma ray, Sonic DT, Deep resistivity	269,505	16,764	133,938
17	Depth, Gamma ray, Sonic DT, Medium resistivity	257,499	14,834	105,536
18	Gamma ray, Sonic DT, Medium resistivity	257,499	14,834	105,536

Table 6
Hyperparameter grids used for the systematic search.

RF		SVR		MLPR	
Kernel	Rbf, Sigmoid	Max depth	None, 10, 25, 50	Tolerance	0.001,0.0001
C	0.1, 1, 10, 100	Min. samples split	2, 4	Solver	Adam, lbfgs
Epsilon	0.1, 0.2	Estimators	10, 25, 50, 100	Alpha	0.0001, 0.00001
Tolerance	0.001, 0.0001			Initial learning rate	0.0001, 0.00001
				Hidden layers	(32 × 12), (64 × 9), (128 × 6)

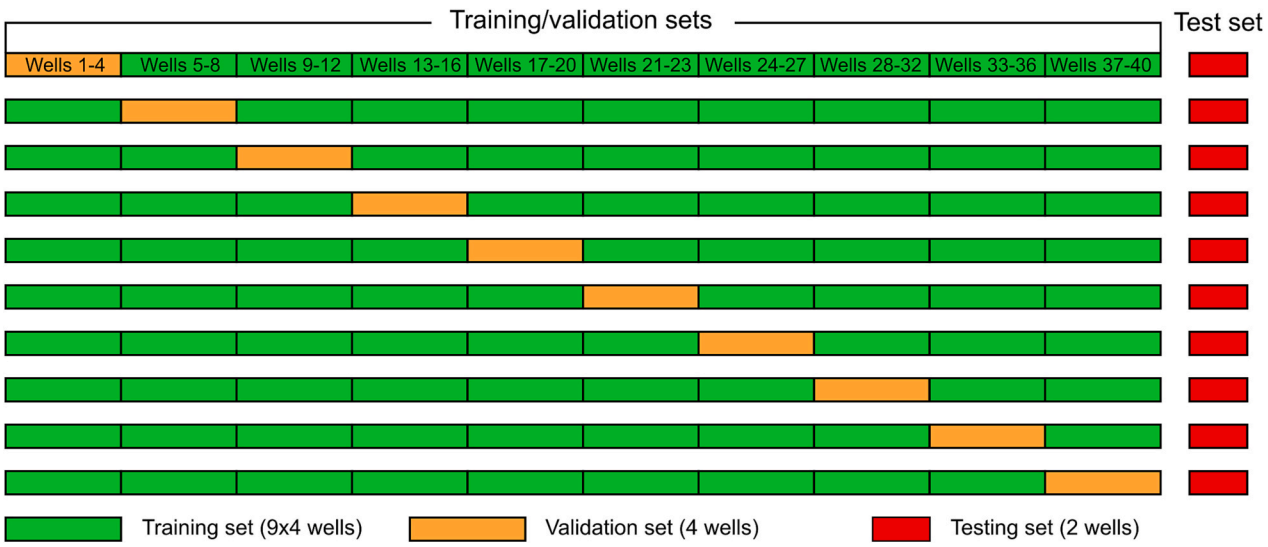


Fig. 8. Example of training/validation sets distribution for a 10-fold cross correlation with 40 wells and 2 wells left out for later blind testing.

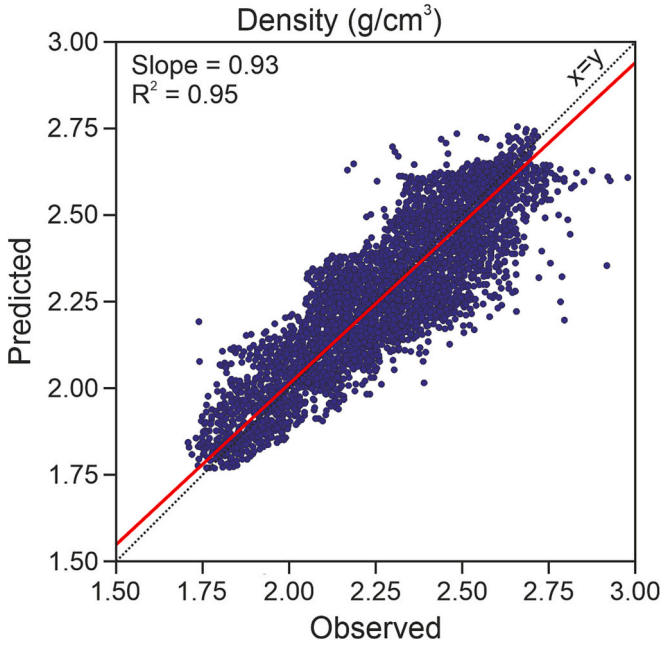


Fig. 9. Example of predicted versus observed density values for one run.

applied to fill gaps in our original dataset. After this process, the number of observations with available density data rose from 380,000 to 484,395, nearly a 30% increase. Fig. 14 shows the density logs of five selected wells before and after the regression process. In these examples the increment is particularly remarkable, doubling (well 15/30–5) or near quadruplicating (22/30B-A13Z) the number of density values. The AI prediction provided density values for a wider range of depths, in contrast to the measured data that was restricted to the lower ~2000–4000 ft. We generated LAS files for each well, including the predicted density (Fig. 15), and including a newly created column identifying each density value as either originally recorded or predicted with AI tools.

Once properly trained, these regressors may also be applied to other areas where not enough observations are available (pre-trained regressors). Although the quality will probably be lower than those for regressors trained with data from the same area (that may be called ad-hoc regressors), they may provide an approximate estimation.

4. Conclusions and recommendations

The workflow proposed in this study culminates with the attainment of two types of outcomes: more complete well log datasets and ready-to-use/pre-trained regressors. Regarding the first point, our density dataset grew in number of observations from 380,000 to 484,395. This number could be further increased, although at the cost of presumably greater errors, by employing sub-optimal regressors to fill additional gaps. Should this become necessary, we recommend to progressively fill gaps starting with the best solution (highest pseudo-F1 score) and continuing with the less apt feature, method and hyperparameter combinations (lower pseudo-F1 scores). This would allow one to fill more gaps than if only the most optimal solution is employed but, at the same time, ensuring that they are filled with the best available regressor for each iteration.

For example, the best solution might require four input features, but some of the observations without the log to be estimated only have three. The proposed procedure is to use first the regressor with four input features and, afterwards, apply one that uses only three for the remaining gaps. The second regressor could have been applied first and it would have filled more observations at once. However, the prediction for those observations where four input features were available would

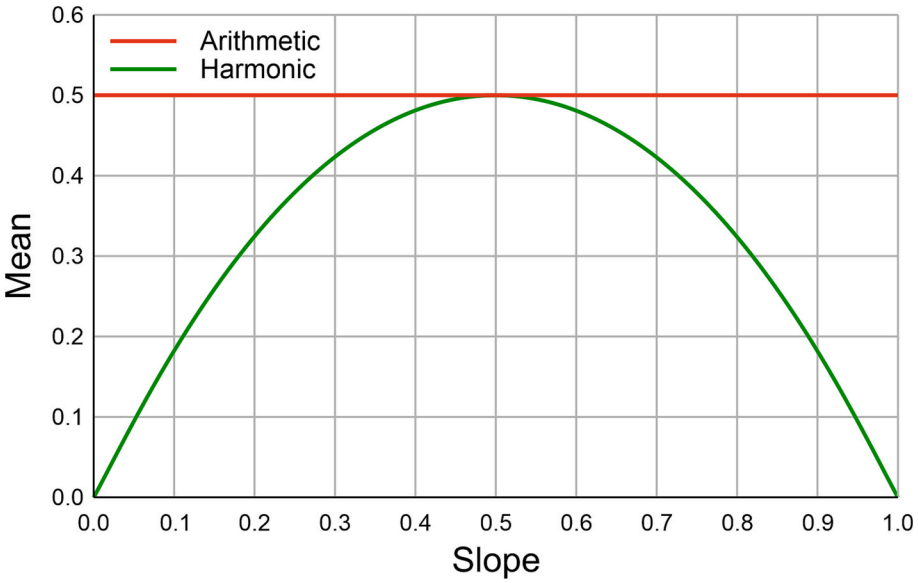


Fig. 10. Graph illustrating the difference between the arithmetic (red) and the harmonic (green) means using as an example the situation where the slope plus R^2 equals 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

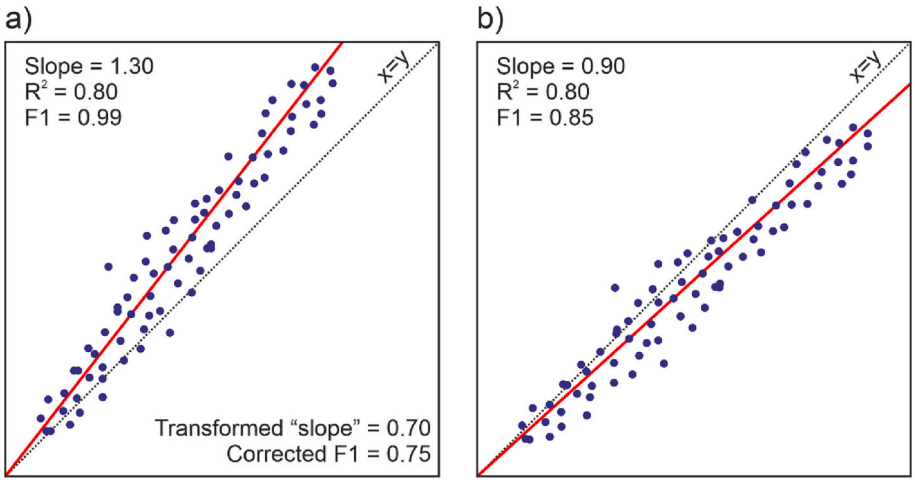


Fig. 11. Graph illustrating the situation where the value of the slope may lead to an incorrect selection of the best feature combination. The solution in a) seems better than b) before applying the transformation to the slope and the recalculation of the pseudo-F1 score.

Table 7
Error summary of the 10 best hyperparameter/feature combinations using MLPR; the first and the third ones were used for the blind testing. NN: neural network architecture (1 = (32 × 12); 2 = (64 × 9); 3 = (128 × 6)), in all the cases they use a tolerance of 0.0001, adam solver, an alpha of 0.0001 and an initial learning rate of 0.0001; IF: input feature combination, see numerical correspondence in Table 6; MaxE: maximum error; MAE: mean absolute error; NMAE: normalized mean absolute error (% of the mean value for the property to be predicted); Slope: slope of the linear fit for the predicted versus observed graph; R^2: dispersion for the predicted versus observed graph; F1(val): pseudo-F1 score for the validation set; F1(tr): pseudo-F1 score for the training set.

NN	IF	MaxE	MAE	NMAE	Slope	R^2	F1(val)	F1(tr)
2	13	0.825	0.044	1.944	0.933	0.920	0.926	0.939
3	13	0.805	0.044	1.940	0.932	0.920	0.926	0.942
1	17	0.707	0.045	1.975	0.919	0.919	0.919	0.946
1	13	0.867	0.045	2.007	0.917	0.916	0.917	0.927
2	17	0.795	0.049	2.181	0.925	0.901	0.913	0.955
3	17	0.757	0.045	2.011	0.912	0.914	0.913	0.946
1	9	0.819	0.043	1.899	0.910	0.885	0.897	0.934
2	9	0.709	0.044	1.940	0.910	0.884	0.897	0.935
3	15	0.810	0.049	2.180	0.936	0.858	0.896	0.952
3	9	0.682	0.043	1.909	0.899	0.889	0.894	0.937

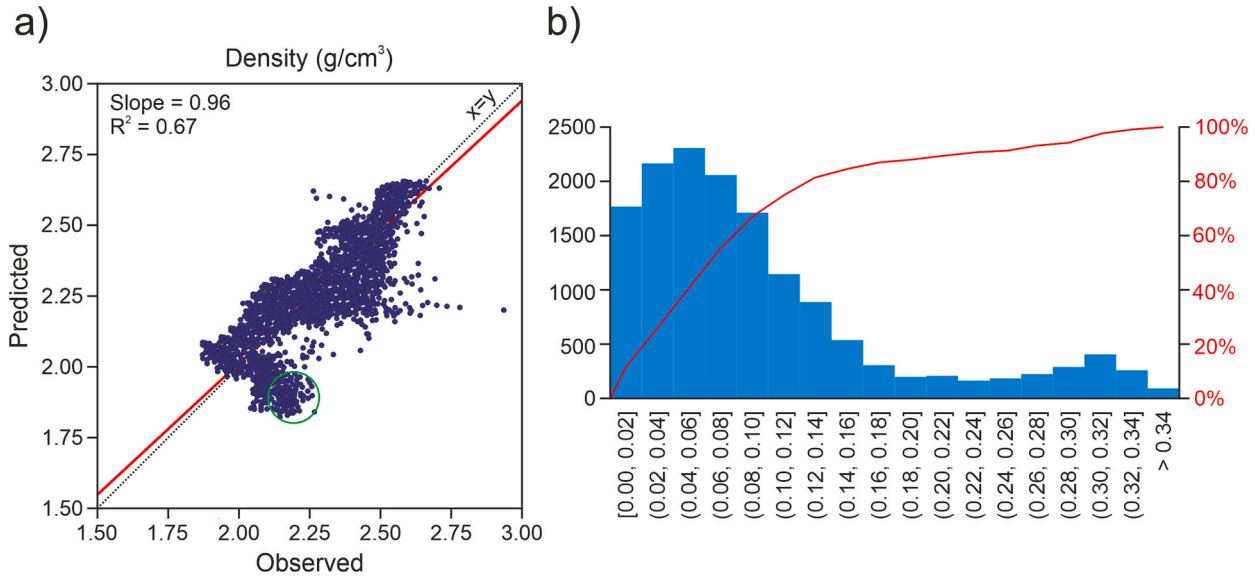


Fig. 12. a) Predicted versus observed density values for the blind test with the best input features and hyperparameter combination (red line: best linear fit of the points; dotted black line: x = y reference line; green circle: problematic points); b) histogram showing its absolute errors (red line: cumulative histogram). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 8
Error summary for the blind testing. NN: neural network architecture (1 = (32 × 12); 2 = (64 × 9)); IF: input feature combination, see numerical correspondence in Table 6; MaxE: maximum error; MAE: mean absolute error; NMAE: normalized mean absolute error (% of the mean value for the property to be predicted); Slope: slope of the linear fit for the predicted versus observed graph; R^2: dispersion for the predicted versus observed graph; F1(test): pseudo-F1 score for the test set; F1(tr): pseudo-F1 score for the training set.

NN	IF	MaxE	MAE	NMAE	Slope	R^2	F1(test)	F1(tr)
2	13	0.738	0.094	4.171	0.961	0.675	0.793	0.941
1	17	0.800	0.084	3.714	0.888	0.687	0.775	0.949

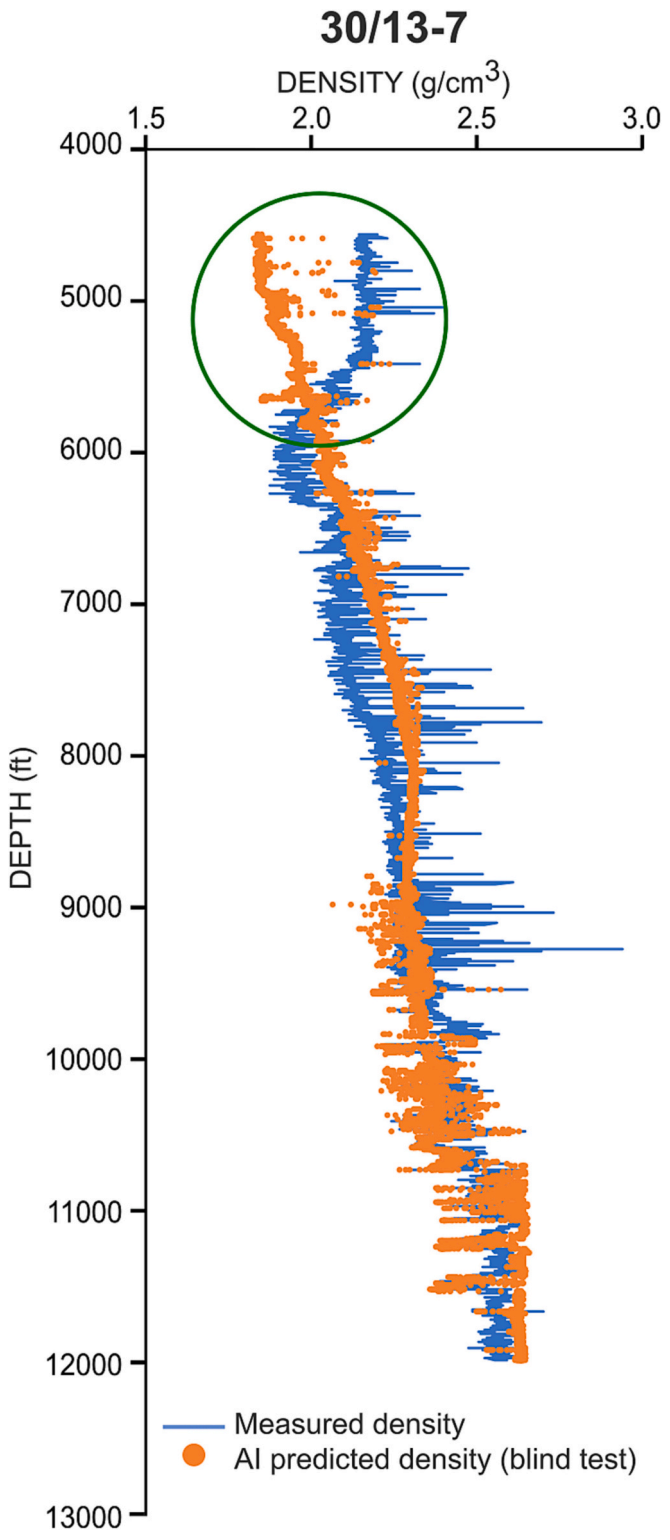


Fig. 13. Graph showing the results of the blind test for well 30/13–7, comparing the density values measured in the well (blue line) with the AI predicted values (orange dots). The green circle highlights the problematic depth interval between ~4500 and ~ 6000 ft. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

be worse than if it were done in two steps. Because of that, we suggest the use of progressively less adequate regressors.

The second key outcome is a collection of ready-to-use regressors. Commonly, it is better to train the regressor with the same (or, at least, similar) dataset in which it will be applied, ensuring that the training and validation sets reflect the variability of the observations to be predicted. However, the use of pre-trained estimators is of great help when, for instance, there are not enough observations recorded to run the training phase. Pre-trained regressors may also be useful for first-pass, quick estimations, as their application is much less laborious than the whole process of training, validation and testing.

One important step to do when adding predictions to the resulting well log files (typically exported as LAS files) is to add a column denoting what observations were initially recorded and what observations were predicted using the AI methods (Fig. 15). A simple binary field may be enough (e.g., 0 = observed, 1 = predicted). However, a more complex system informing the regressor applied and the confidence in the value may also be implemented (i.e., 0 = observed, 1 = predicted with the best regressor, 2 = predicted with the second best regressor and so on). In the header of the resulting file, or in a separate report, information regarding the accuracy and dispersion of the blind test for each regressor could be included. One drawback of this, it is that it may introduce an unnecessary complexity, with information that may be hard to digest depending on the user's familiarity with AI/mathematical concepts. Tracking what has been estimated is especially relevant when, for example, the predicted values are to be included in the prediction of a second incomplete well log or to be used for other tasks such as classification problems. In this case, if possible, the training of new regressors/classifiers should be done taking only the observed values and leaving out the predicted ones. Should the use of predicted values be necessary, the geoscientist must keep in mind that the results will probably be worse, as the errors will accumulate during consecutive predictions.

Funding

This work was partially funded by the National Research Agency (Agencia Estatal de Investigación, Ministerio de Ciencia, Innovación y Universidades of Spain) through their Torres Quevedo program (grant reference: PTQ2018–009983).

CRediT authorship contribution statement

H. Uzkeda: Conceptualization, Methodology, Software, Validation, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization. **O. Vidal-Royo:** Conceptualization, Methodology, Validation, Investigation, Resources, Writing – review & editing, Visualization, Funding acquisition. **A. Amilibia:** Conceptualization, Methodology, Validation, Investigation, Resources, Writing – review & editing, Visualization, Funding acquisition.

Declaration of Competing Interest

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

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Hodei Uzkeda reports financial support was provided by Agencia Estatal de Investigación, Ministerio de Ciencia, Innovación y Universidades of Spain.

Table 9
Table comparing the errors in well 30/13–7, separated by depth.

Depth	Observations	Mean	Std	Min	P25	P50	P75	Max
4553.0–5674.5	2117	0.249	0.073	0.052	0.205	0.270	0.309	0.373
>5674.5	12,717	0.069	0.048	0.000	0.034	0.061	0.094	0.738

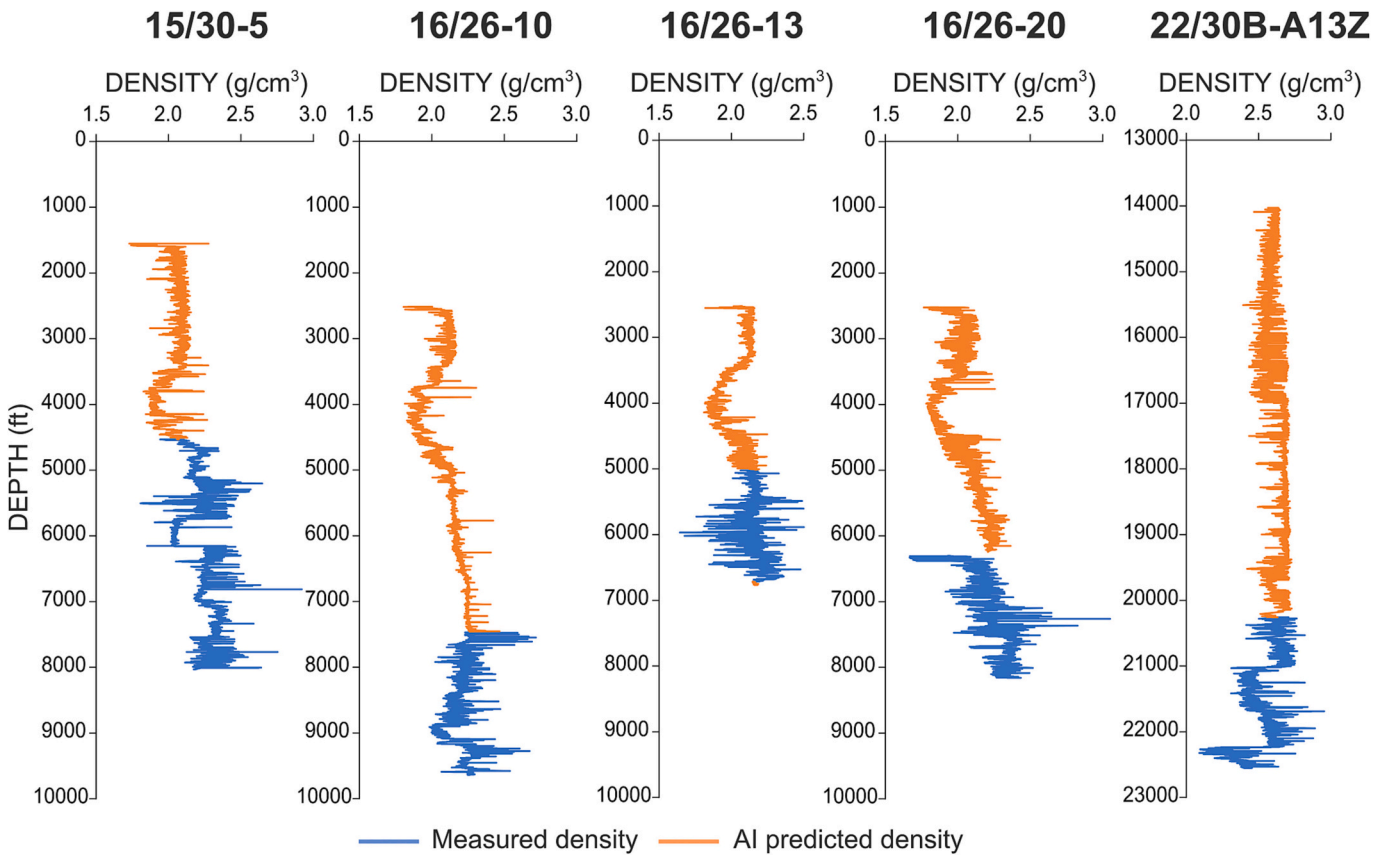
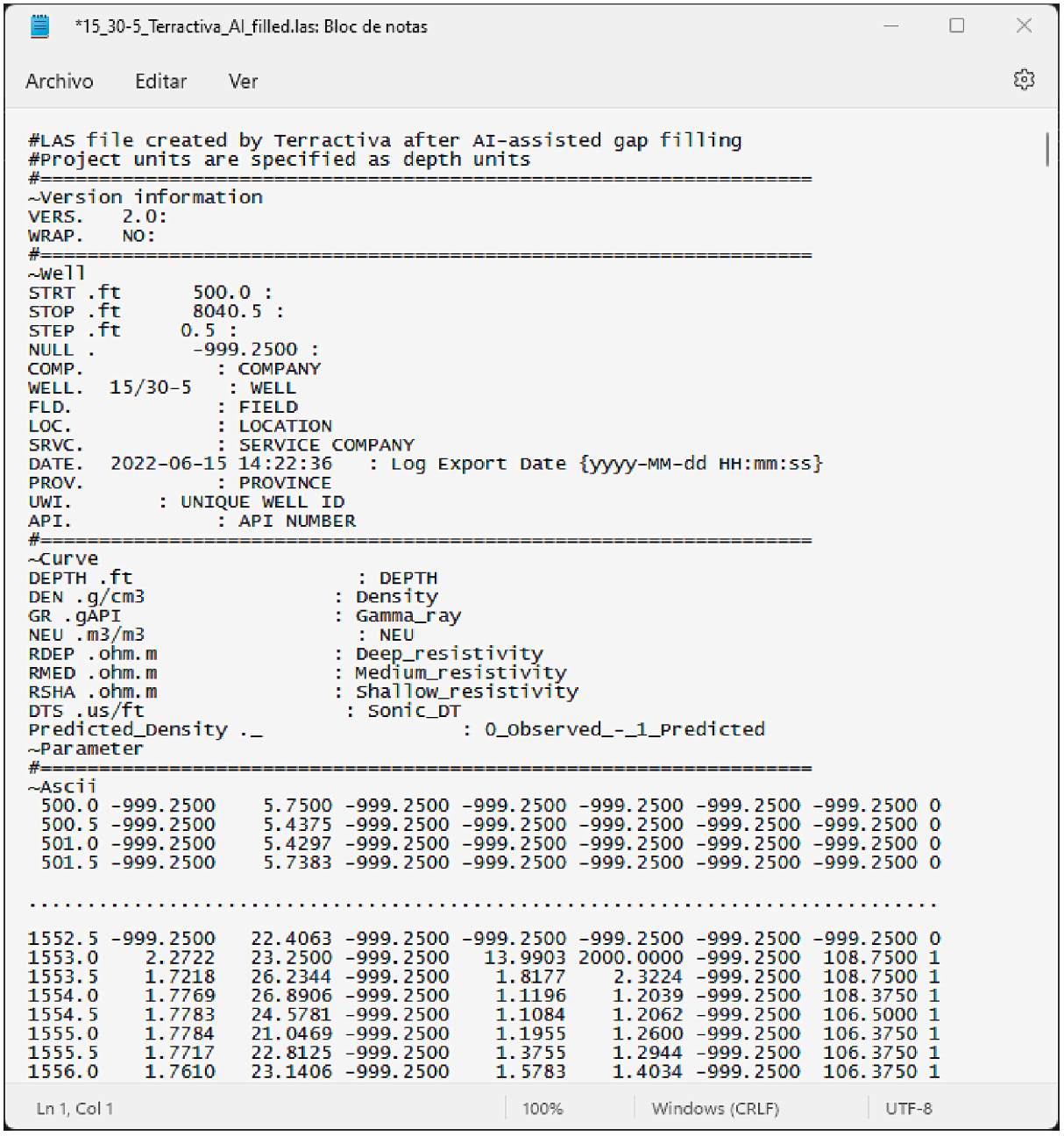


Fig. 14. Density logs for five wells showing the originally available data (blue lines) and the new values predicted (orange lines) using the best input features and hyperparameter combination. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



*15_30-5_Terractiva_AI_filled.las: Bloc de notas

Archivo Editar Ver

```
#LAS file created by Terractiva after AI-assisted gap filling
#Project units are specified as depth units
#================================================================
~Version information
VERS.    2.0:
WRAP.    NO:
#================================================================
~Well
STRT .ft       500.0 :
STOP .ft       8040.5 :
STEP .ft      0.5 :
NULL .          -999.2500 :
COMP.             : COMPANY
WELL.  15/30-5    : WELL
FLD.              : FIELD
LOC.              : LOCATION
SRVC.             : SERVICE COMPANY
DATE.  2022-06-15 14:22:36   : Log Export Date {yyyy-MM-dd HH:mm:ss}
PROV.             : PROVINCE
UWI.        : UNIQUE WELL ID
API.              : API NUMBER
#================================================================
~Curve
DEPTH .ft                    : DEPTH
DEN .g/cm3                 : Density
GR .gAPI                   : Gamma_ray
NEU .m3/m3                   : NEU
RDEP .ohm.m                : Deep_resistivity
RMED .ohm.m                : Medium_resistivity
RSHA .ohm.m                : Shallow_resistivity
DTS .us/ft                  : Sonic_DT
Predicted_Density ._                  : 0_Observed_-_1_Predicted
~Parameter
#================================================================
~Ascii
 500.0 -999.2500    5.7500 -999.2500 -999.2500 -999.2500 -999.2500 -999.2500 0
 500.5 -999.2500    5.4375 -999.2500 -999.2500 -999.2500 -999.2500 -999.2500 0
 501.0 -999.2500    5.4297 -999.2500 -999.2500 -999.2500 -999.2500 -999.2500 0
 501.5 -999.2500    5.7383 -999.2500 -999.2500 -999.2500 -999.2500 -999.2500 0

.....................................................................................

1552.5 -999.2500   22.4063 -999.2500 -999.2500 -999.2500 -999.2500 -999.2500 0
1553.0    2.2722   23.2500 -999.2500   13.9903 2000.0000 -999.2500  108.7500 1
1553.5    1.7218   26.2344 -999.2500    1.8177    2.3224 -999.2500  108.7500 1
1554.0    1.7769   26.8906 -999.2500    1.1196    1.2039 -999.2500  108.3750 1
1554.5    1.7783   24.5781 -999.2500    1.1084    1.2062 -999.2500  106.5000 1
1555.0    1.7784   21.0469 -999.2500    1.1955    1.2600 -999.2500  106.3750 1
1555.5    1.7717   22.8125 -999.2500    1.3755    1.2944 -999.2500  106.3750 1
1556.0    1.7610   23.1406 -999.2500    1.5783    1.4034 -999.2500  106.3750 1
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Fig. 15. Example of AI filled LAS file, showing the additional column indicating whether the value was originally recorded or has been predicted.

Data availability

The data used in this work is freely accessible through the UK National Data Repository (NDR), website: https://ndr.nstauthority.co.uk/

Acknowledgements

We would like to acknowledge the OGA for providing the well data and the scikit-learn project for the AI tools employed. This work was partially funded by the National Research Agency (Agencia Estatal de Investigación, Ministerio de Ciencia, Innovación y Universidades of Spain) through their Torres Quevedo program (grant reference: PTQ2018-009983). Our manuscript benefited from the constructive comments made by two anonymous reviewers. Also, we would like to thank Grant George Buffett for reviewing the English.

References

Caté, A., Perozzi, L., Gloaguen, E., Blouin, M., 2017. Machine learning as a tool for geologists. Lead. Edge 36 (3), 215–219.

Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V., 1997. Support vector regression machines. Adv. Neural Inf. Proces. Syst. 9, 155–161.

Dubois, M.K., Bohling, G.C., Chakrabarti, S., 2007. Comparison of four approaches to a rock facies classification problem. Comput. Geosci. 33, 599–617.

Goutorbe, B., Lucazeau, F., Bonneville, A., 2006. Using neural networks to predict thermal conductivity from geophysical well logs. Geophys. J. Int. 166 (1), 115–125.

Hart, N., Palmer, I., Pine, K., 1997. Geological Well Report 30/13–7. Internal report. Phillips Petroleum (UK) LTD, 64 p.

Ho, T.K., 1995. Random decision forests. In: In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, pp. 278–282.

Imamverdiyev, Y., Sukhostat, L., 2019. Lithological facies classification using deep convolutional neural network. J. Pet. Sci. Eng. 174, 216–228.

Konaté, A.A., Pan, H., Khan, N., Yang, J.H., 2015. Generalized regression and feed-forward back propagation neural networks in modelling porosity from geophysical well logs. J. Pet. Explor. Prod. Technol. 5, 157–166.

Kumar, T., Seelam, N.K., Rao, G.S., 2022. Lithology prediction from well log data using machine learning techniques: a case study from Talcher coalfield, Eastern India. J. Appl. Geophys. 199, 104605.

OGA, 2022. Web. https://ndr.ogauthority.co.uk/ accessed June 2022.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Sahu, S.K., Sing, S.K., Nair, H.K.P., Ranjan, M., Negi, I.S., Tandon, R., 2019. Facies classification from well logs using a scalable SVM based approach. In: Proceedings of the 13th Biennial International Conference and Exhibition.

Tokpanov, Y., Smith, J., Ma, Z., Deng, L., Benhallam, W., Salehi, A., Zhai, X., Darabi, H., Castineira, D., 2020. Deep-learning-based automated stratigraphic correlation. In: Proceedings of the SPE Annual Technical Conference and Exhibition, SPE-201459-MS.

Tschannen, V., Delescluse, M., Rodriguez, M., Keuper, J., 2017. Facies classification from well logs using an inception convolutional network. In: Computer Vision and Pattern Recognition. arXiv:1706.00613.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York, p. 334.

Ważny, J., Stefaniuk, M., Cygal, A., 2021. Estimation of electrical resistivity using artificial neural networks: a case study from Lublin Basin, SE Poland. Acta Geophys. 69, 631–642.

Wedge, D., Hartley, O., McMickan, A., Green, T., Holden, E.-J., 2019. Machine learning assisted geological interpretation of drillhole data: examples from the Pilbara Region, Western Australia. Ore Geol. Rev. 114, 103118.

Zazoun, R.S., 2013. Fracture density estimation from core and conventional well logs data using artificial neural networks: the Cambro-Ordovician reservoir of Mesdar oil field, Algeria. J. Afr. Earth Sci. 83, 55–73.

Zhou, B., O'Brien, G., 2016. Improving coal quality estimation through multiple geophysical log analysis. Int. J. Coal Geol. 167, 75–92.