A machine learning workflow for log data prediction at the basin scale

Keyla Gonzalez[1], Olga Brusova[1] and Alejandro Valenciano[1]* describe a machine learning workflow to predict missing data in well logs at the basin scale.

Summary

Log data recorded by wireline tools are incomplete in most well locations. Vital information often needs to be predicted to precisely characterise the Earth's subsurface. Here we describe a machine learning (ML) workflow to predict missing data in well logs at the basin scale. The ML models produce outstanding results when adequate quality data is provided for the model training and inference. Using examples from the Permian Basin in the US, we illustrate the use of the automated data clean-up pipeline and the clean-up impact on ML algorithm training and prediction. The ML models achieve a prediction quality of 90% to 95% in a blind test containing 679 wells if trained on clean data from the Permian Basin.

Introduction

Wireline logs are an integral component in characterising subsurface properties. For economic reasons, data from specific logs or depth intervals are not collected, resulting in incomplete information from the surface to the base of the well in most locations. An alternative that addresses the lack of data is to create synthetic curves. With the availability of millions of digitised wells with a wide spatial distribution and the recent advances of data science, the prediction of missing logs or missing log intervals with machine learning (ML) algorithms is now possible.

Data preparation and clean-up are fundamental steps before ML algorithms training and inference. In conventional petrophysical or geophysical workflows, log data clean-up is done manually, one well at a time. The clean-up tasks include, but are not limited to, verifying information in the LAS files, curve categorization, units' standardisation, log splicing, and log spatial normalisation. The manual approach to well log clean-up is not scalable, as training ML algorithms requires extensive data. To create practical ML tools for well log prediction, we first need to develop an automated pipeline for well log cleaning.

This paper presents an end-to-end machine learning workflow for well logs prediction at a basin scale. The first section describes the well log cleaning processing, which encompasses an extensive data clean-up before ML training and inference. The second part of the paper introduces the ML training, validation, and inference based on a gradient-boosted regression trees algorithm. We developed an ML pipeline to predict missing data in well logs for ten US onshore basins (Figure 1). The ML workflow is illustrated on data from the Permian Basin in the US.

Well log clean-up workflow

This section describes the well log cleaning pipeline, which encompasses extensive data clean-up before ML training and inference (Figure 2). The pipeline includes assigning each curve to a predefined category, verifying information in LAS and log headers, splicing, and merging logs from different runs, depth shifting for log depth alignment, normalising logs for tool/environmental effects, and removing or editing inferior quality

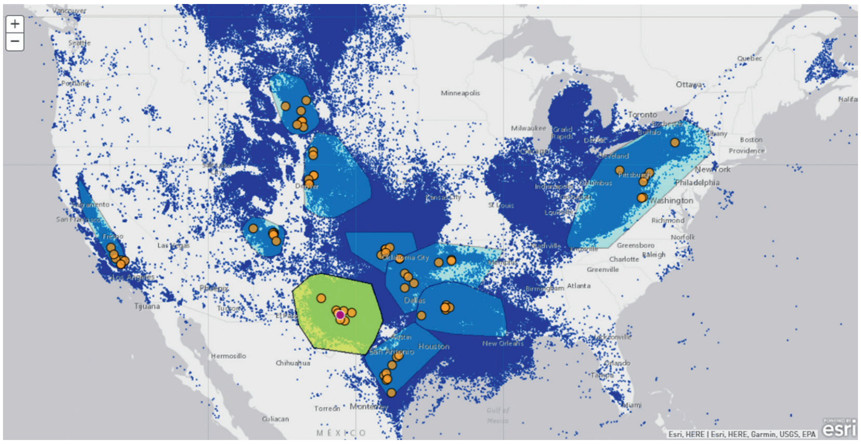

Figure 1 US onshore basins with existing ML models for predicting missing data in well logs. The polygons indicate areas where we created ML models. The yellow polygon encircles the data used for the Permian Basin model.

[1] TGS
* Corresponding author, E-mail: alejandro.valenciano@tgs.com
DOI: 10.3997/1365-2397.fb2023015

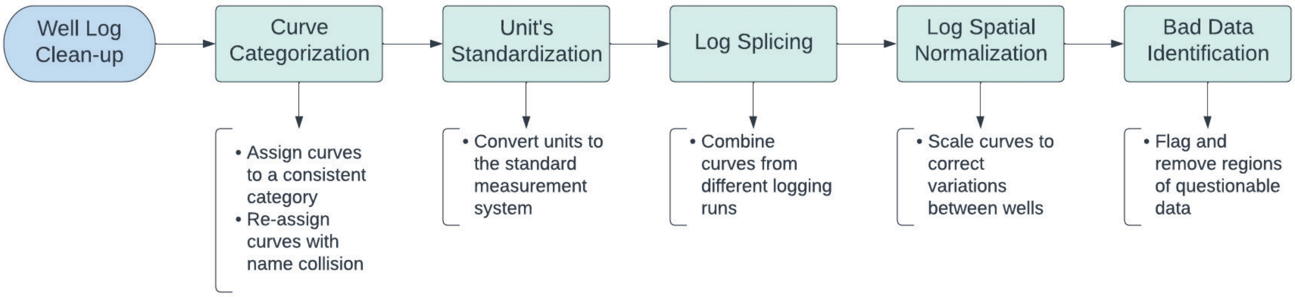
Well Log Clean-up
Curve Categorization
Unit's Standardization
Log Splicing
Log Spatial Normalization
Bad Data Identification
• Assign curves to a consistent category
• Re-assign curves with name collision
• Convert units to the standard measurement system
• Combine curves from different logging runs
• Scale curves to correct variations between wells
• Flag and remove regions of questionable data

Figure 2 Well log clean-up workflow.

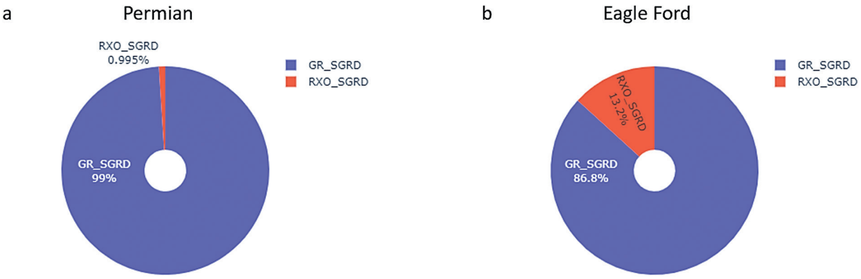

Figure 3 SGRD curves for different curve classes in Permian (a) and Eagle Ford (b) basins. RXO_SGRD is an SCGR mnemonic that was assigned to the shallow resistivity category based on curve analyses. GR_SGRD SGRD is an SCGR mnemonic assigned to the gamma ray category based on curve analyses.

data. A design principle observed during the clean-up was to minimise human interpretation as much as possible to allow for a basin-wide application.

Curve categorisation

There is no standard naming convention for well log curves. The name of the curve depends on the measurement type, logging tool, and the operator. Different operators have different naming conventions for the same measurement types. As a result, a basin-wide set of LAS files may include hundreds of varying curve mnemonics.

The first processing step in our log clean-up workflow is assigning a category to each curve in the dataset. We considered the following main curve categories: Gamma-ray, Neutron Porosity, Compressional Sonic, Bulk Density, and Resistivity. The Neutron Porosity is further classified in the flow based on the rock matrix set up for the measurements: limestone, sandstone, or dolomite. The resistivity curve is additionally categorised based on penetration depth (deep, medium, shallow, micro). We built a comprehensive list of mnemonics for each curve category that users can update if a new unknown mnemonic is encountered.

Sometimes the same mnemonics can be encountered for different log types in a LAS file. One example is SGRD, which can be found for either spectral gamma-ray curve run with density tool or shallow guard resistivity curve. To assign the curve to correct categories, we need to check the units of the curve, curve description, and/or curve values. SGRD mnemonic clash is a big problem for US onshore basins. Figure 3 shows the scale of the problem for the Permian and Eagle Ford basins. While for the Permian basin, not resolving the SGRD mnemonics issue results in 1% of miscategorised curves, for the Eagle Ford basin, this error reaches 13%.

Units standardisation

Unit standardisation is a necessary step in the log clean-up process. Although there are standard units for different curve types, curves are also reported in non-standard units, using other measurement systems, with typos and mistakes or sometimes with no units. In this step of the flow, the necessary unit conversions are performed. We also have a comprehensive list of unit aliases to cross-check typos in the unit names. Curves with nonstandard units are flagged and can be checked manually.

Curve categorisation and unit correction can provide additional data for further use. Figure 4 (a-b) presents the results of fixing the mnemonics and the units of gamma-ray logs for an arbitrary cross-section of 800 wells in the Permian basin. Figure 4a shows the cross-section with minimal mnemonic and unit processing using common mnemonics. The cross-section (Figure 4a) is missing data compared to the cross-section in Figure 4b, where we used a more comprehensive mnemonic table and automatic unit discovery with standard unit conversions. Due to thorough mnemonic and unit QC, several logs are added to the cross-section (Figure 4b).

Log splicing

Log splicing (merging) is a critical step in the log cleaning workflow. This process combines similar log curve measurements from different logging runs to construct a single curve. Overall, log splicing is hard to automate, especially if there is no metadata information on multiple logging runs. But after making reasonable assumptions and simplifications, we can achieve basin-wide reliable results.

The log splicing starts by defining the primary and secondary curve(s) in all available logging runs. The primary curve is selected to be the longest valid (with correct units and values in a predetermined range) curve that overlaps with most of the critical logs in the well. The quality of secondary curves is rated. Later, they are merged into the primary curve in the order of importance. The primary and secondary curves are compared in overlapping intervals during the splicing process. We fixed the depth mismatch by shifting the secondary curve to be depth-aligned with the primary. After confirming that curves are 'tied' in-depth, we also need to ensure that curves are on a similar scale as logs must be calibrated relative to each other before merging. If required, the calibration happens in the overlap interval after removing artifacts at the end of the curves (e.g., casing points, stacked tools). After calibration, the logs are on the same scale and can be merged.

Examples of this processing step are shown in Figure 5. Figure 5a shows the importance of calibration before log splicing. The first track shows two gamma-ray curves from the same LAS file. The black curve (primary) has a more significant variance than the red curve (secondary). It is also offset and reads higher values for the same depths. After normalisation, the second track shows the same curves; the curves are brought to the same scale and show consistent reading in the overlap interval. Finally, the third track shows the spliced curve ready for further processing.

The depth alignment before log splicing using two gamma-ray curves from the same LAS file is shown in Figure 5b. There is an apparent depth mismatch between the two curves in the first track. In the second track, the curves are depth aligned. The red curve (secondary) is shifted upwards to match the black curve (primary). The third track shows the spliced curve.

The effects of splicing are shown in the cross-section in Figure 4c. After taking a closer look at the differences between Figures 4b and 4c, we notice that more data is added to the cross-section.

Logs spatial normalisation

After unit standardisation and log splicing, a log curve basin-scale spatial normalisation is necessary. Basin-scale well log datasets contain wells of different vintages, logging tool manufacturing, calibration variations, inconsistent log units, changes in borehole environment (size, drilling fluid, etc.), and various environmental corrections. By basin-scale spatial normalisation, we renormalise logs with obvious scale problems to have the same overall trends based on the neighboring wells.

The normalisation process described here is similar to a recalibration of the curves in the log splicing tool discussed previously. Calibration is applied to the curves in the same well before merging, while normalisation is the process that standardises logs from different wells across the dataset. Basin-scale spatial normalisation is commonly applied to statistical logs like gamma-ray and neutron porosity. Other logs like resistivity, bulk

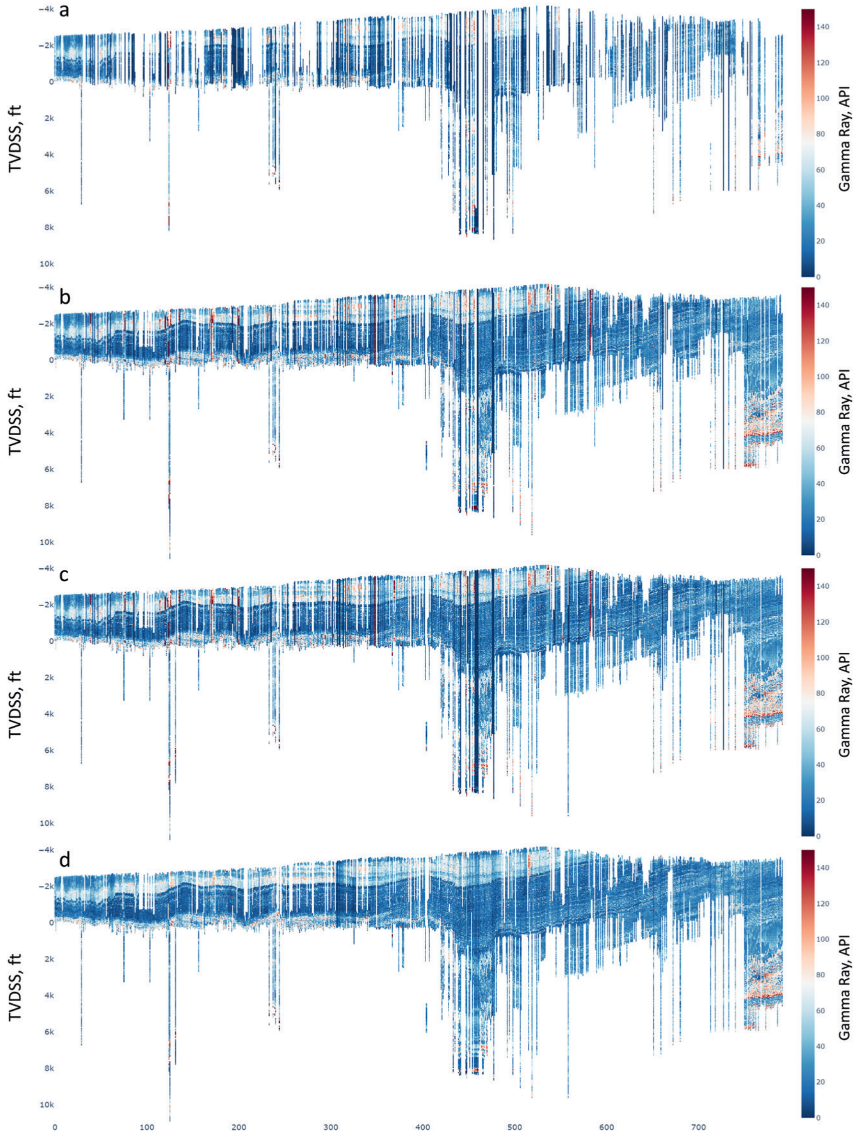

Figure 4 Change in the state of an arbitrary gamma-ray cross section in the Permian Basin during the clean-up workflow. (a) Shows data with common mnemonics and standard units. (b) Shows the same data after more rigorous mnemonic analyses, fixing mnemonic clashes, checking, and fixing unit issues. Due to thorough mnemonic and unit quality control, several logs are added to the cross-section. (c) Shows the effect of log merging from different runs. The wells have better depth coverage. (d) Shows the effect of basin-scale log normalisation. Most artificially high and low gamma-ray values present in 4c are fixed in 4d.

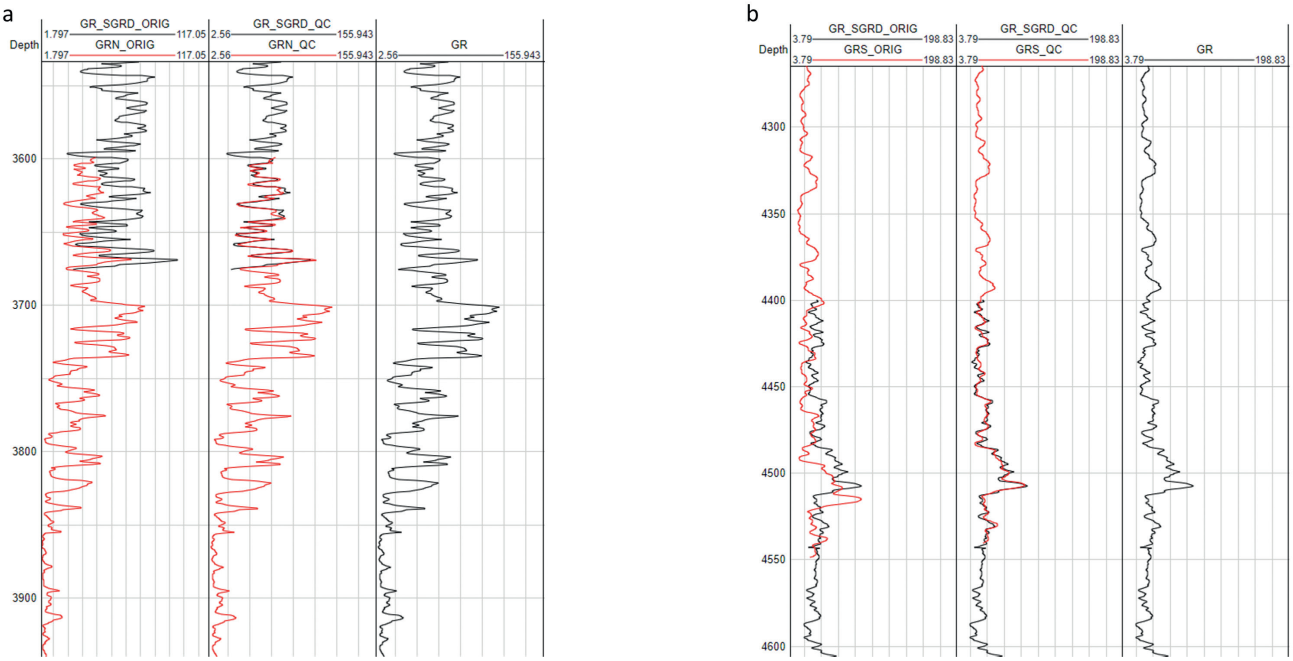

Figure 5 Problems solved in the log splicing process. (a) Calibration before log splicing. (b) Depth alignment before splicing. The first track on each panel shows two curves in the same well: the primary curve is black, and the secondary curve is red. The second track shows the same curves after shifting. The last track shows the final merged curve after log splicing.

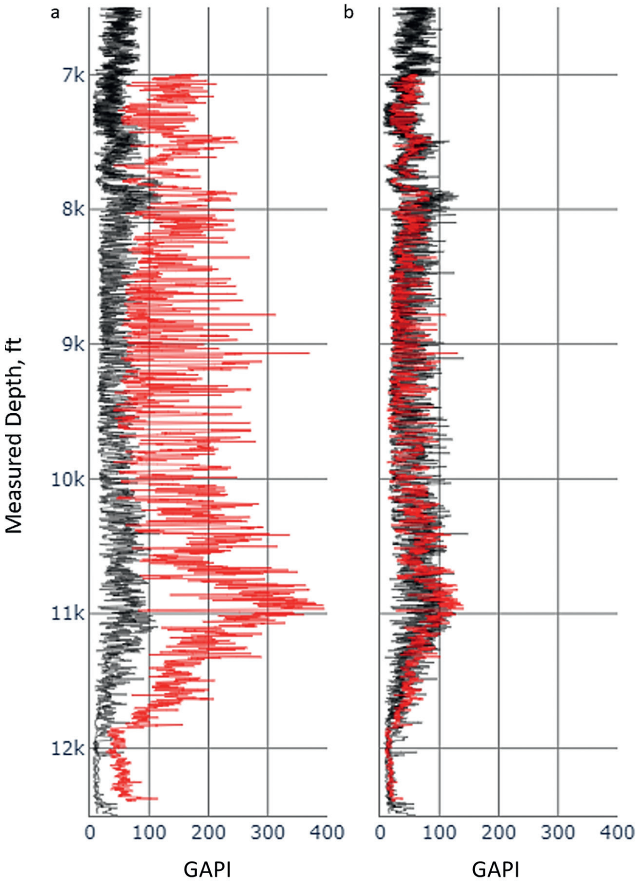

Figure 6 Gamma-ray curve (red) before (a) and after (b) normalisation flow as compared to the reference curve (black).

density, and sonic are not normalised unless there is a good reason, as normalisation can result in artificial changes in the tool's response to rock properties or geological variations Shier (2004).

In our workflow, we apply normalisation to gamma-ray logs only. Gamma-ray logs are used to determine the percentage of shale/sand in the lithology column. This is a relative measurement where we need to define sand and shale baselines. Absolute measurements are not crucial for this calculation. Normalisation, if done correctly, will not affect rock properties derived from gamma-ray. Normalised gamma-ray logs are easier to use in batch processing, where a set of interpretation parameters is applied across the whole field. Equation 1 was used for normalising gamma-ray logs to a reference curve so that the range of the values after normalisation becomes comparable to the reference curve:

$$GR_{norm} = GR_{ref_{low}} + \left(GR_{ref_{high}} - GR_{ref_{low}}\right) * \left(\frac{GR - GR_{low}}{GR_{high} - GR_{low}}\right) \quad (1),$$

where GR is the gamma-ray log value before normalisation, GR_{norm} is the gamma-ray log value after normalisation, GR_{low} is the gamma-ray log 'sand' point, GR_{high} is the gamma-ray log 'shale' point, $GR_{ref_{low}}$ is the reference gamma-ray 'sand' point, $GR_{ref_{high}}$ is the reference gamma-ray 'shale' point. Gamma-ray low value ('sand' point) is the average gamma-ray count for clean sandstone of local geology. Gamma-ray high value ('shale' point) is the average gamma-ray counts for clean shale. Low sand and high gamma-ray constants can be approximated by the 5th and 95th percentile curve statistics correspondingly.

Figure 6 shows the effect of spatial normalisation in two adjacent wells. Figure 6a shows two gamma-ray curves from different wells before normalisation. The black curve is used as a reference. The red curve reads remarkably high gamma-ray values and needs to be normalised. Figure 6b shows the same curves after normalising the red curve using the black curve as a reference. After normalisation, the gamma-ray log range is similar for both wells.

Reference curve selection is vital for the log normalisation process. We can elect to normalise to a single ‘key’ reference curve if we work with a limited number of wells. Normalising the curve on the basin scale to a single reference curve will no longer be adequate as lithology may change dramatically over considerable distances. Our approach is to select multiple ‘key’ reference wells so we can build basin-wide grids using statistics from these wells. The grids then can be used to normalise the rest of the wells in the basin.

Figure 4d shows the effect of basin-scale log normalisation on a cross-section view. Artefacts shown in cross-section 4c are fixed after normalisation (4d). As a result, the cross-section becomes less disrupted by outliers.

Bad data identification

Well-log curves can report erroneous values related to borehole conditions or equipment failures. The borehole errors can be due to cave-ins, significant mud cakes, and sub-optimal mud types. Another type of error may result from equipment failures, recording errors, or bad tool calibration. They are easy to spot by a trained petrophysicist, but it is harder to train a machine and automate the identification in thousands of wells. In our automated workflow, the user sets basin-wide limits on minimum and maximum values to flag intervals of bad data quality. Outliers are then removed from those intervals, and a flag is set for each curve. Bulk density curve corrections can also rely on complementary curves such as caliper and density correction readings to flag zones where the log may be compromised.

Another way to flag zones of questionable data is by cross-correlating different curves. In most cases different curves (e.g., density and sonic) show strong correlation or anticorrelation as they respond to changes in the same rock properties (e.g., porosity, saturation, the volume of clay, etc.) or combinations of them. Two cross plots in Figure 7 show the effect of data clean-up by removing zones with questionable data. After clean-up, the cross plot has a better-defined trend and less ‘noise’.

Machine learning logs modelling

A complete well log suite with no data gaps and maximum data coverage can provide insight into various subsurface intelligence problems, such as lithofacies picking, velocity model building and production prediction. Analytics Ready LAS (ARLAS) is a basin-scale machine learning pipeline that can predict missing logs. The target logs for prediction are the bulk density, gamma-ray, neutron porosity, deep resistivity, and compressional sonic. For each target curve separate models are trained from different combinations of available measured curves. The underlying algorithm for curve prediction is gradient-boosted regression trees.

The gradient-boosting tree algorithm

Gradient boosted trees belong to a class of tree-based methods (Breiman 2001; Ranka and Singh 1998; Machado 2003) with the capacity for modelling data-driven piece-wise target-feature interactions. A single-decision tree ML model is built, asking questions to partition data recursively based on the feature values before reaching a solution. Figure 8 shows the flowchart diagram of a decision tree. The main advantage of a decision tree model is that it is easy to understand and interpret.

One of the biggest problems with decision trees is overfitting (inferior performance on the data not used during training). Ensemble methods were developed to overcome this problem. They produce several different models and use them to reach the final solution. The gradient-boosting (GB) algorithm (Chen and Guestrin 2016; Li *et al.* 2007) uses an ensemble of short decision trees. It predicts a target value by building weak (only slightly better than random choice) prediction models where each model tries to predict the error left over by the previous model. Another aspect of GB is that the algorithm concentrates on the data that does not give a good result by weighing them

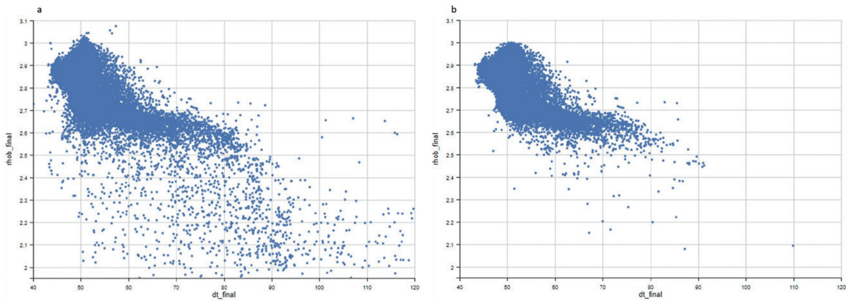

Figure 7 Cross plot of the sonic and density data for a well in the Permian Basin. (a) Shows data before clean-up. (b) Shows data after clean-up. The effect of data clean-up is a better-defined correlation between sonic and density with less ‘noise’.

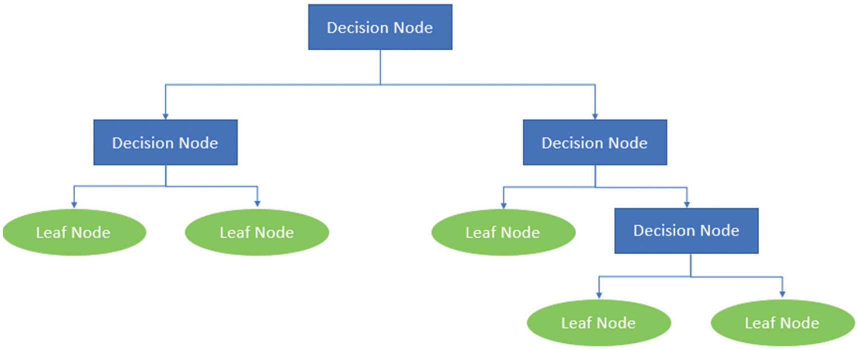

Figure 8 Diagram of a decision tree ML model. The decision nodes partition the input data based on the value(s) of their feature(s). Leaf nodes store the outcome of the prediction.

higher as new learners are trained. GB also uses a learning rate and a loss function to ensure the learning is done optimally. The learning rate and the number of learners employed in the model are usually determined during the model hyperparameter tuning stage. In our work, we use the Light Gradient Boosting (LGB) Tree Algorithm (Ke *et al.* 2017), an optimised version of the GB algorithm.

Features

The underlying tree structure in LGB is invariant to scaling. We can use features in their raw form without normalisation and rescaling. Our pipeline uses the following input features to predict desired logs: compressional sonic log, gamma-ray log, the logarithm of deep resistivity, logarithm of neutron porosity in a limestone matrix, cube of bulk density, true vertical depth below sea level (TVDSS), well latitude, and well longitude. The depth feature helps the model to learn how the log properties change with depth. Spatial features help to tune the model to variations in local geology.

We train models for different combinations of input features and target curves since we can expect variations in feature availability in each well and for different depths within the same well. When predicting sonic from gamma-ray and resistivity the feature vector will consist of gamma-ray, the logarithm of resistivity, spatial locations (latitude and longitude), and TVDSS. We train 15 models for each target curve in a basin to predict the target when different combinations of the five main curves are present.

Some logs are run more often than others and have overall better depth coverage. We can expect gamma-ray and resistivity logs to be available in any basin with the most depth coverage from the top of the well to its base. Sonic and density logs are expensive to run and are not as readily available. This impacts the number of samples available for each model during training.

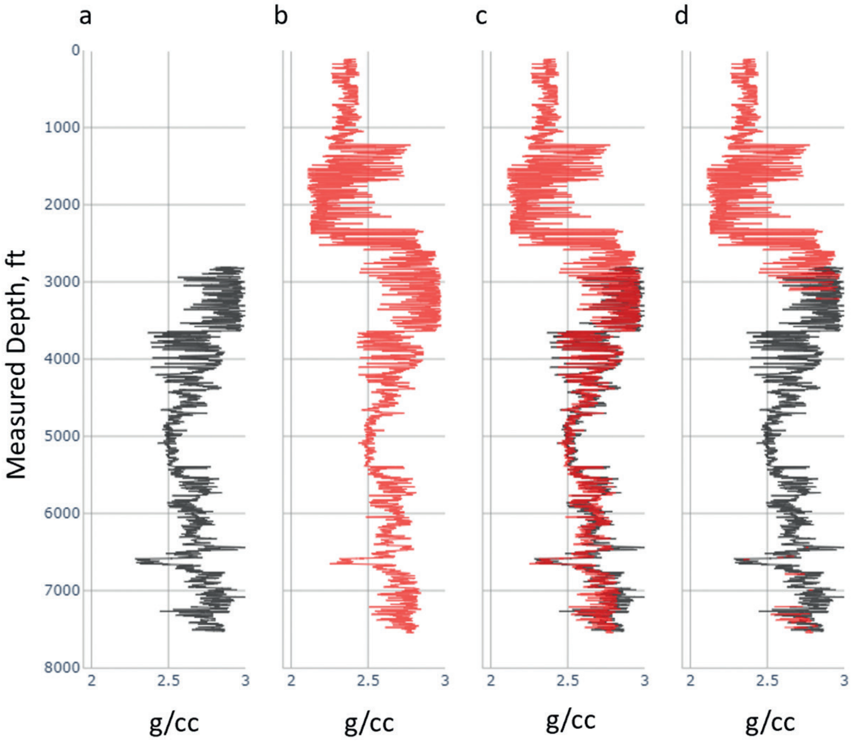

Figure 9 Application of ARLAS modelling workflow to a bulk density log of one of the wells in the Permian Basin. The first track (a) shows the recorded data in the well (black). The second track (b) displays the ARLAS prediction (red). The third track (c) shows the measured data (black) and predictions (red). The last track (d) shows the final product where original data (black) is combined with the predictions (red) to extend the log depth coverage and fill data gaps.

a.		Model Combinations for Analytic Ready LAS																													
Predicted Logs	Density (D)		14.5%	11.7%	13.1%	11.6%					11.2%	12.5%	11.0%	10.8%	10.0%	10.7%							10.4%	9.6%	10.4%	9.6%					9.3%
	Gamma (G)	9.7%		9.2%	9.7%	9.2%		9.2%	9.3%	8.9%				9.1%	8.9%	8.9%				8.9%	8.7%	8.6%				8.7%				8.5%	
	Neutron (N)	8.3%	13.2%		8.2%	8.2%	7.2%		6.8%	6.9%		7.6%	7.7%			7.1%		6.4%	6.5%			6.3%			6.8%				6.1%		
	Resistivity (R)	13.2%	13.5%	11.6%		13.2%	12.9%	11.4%		12.7%	11.4%		12.8%		11.3%		11.3%		12.6%		11.2%			11.1%				11.0%			
	Sonic (P)	10.4%	12.3%	10.3%	10.6%		9.4%	9.3%	9.1%		9.8%	9.9%		9.5%			8.9%	8.6%		8.8%			9.1%				8.4%				
	Normalized RMSE	D	G	N	R	P	D+ G	D+ N	D+ R	D+ P	G+ N	G+ R	G+ P	N+ R	N+ P	R+ P	D+ G+ N	D+ G+ R	D+ G+ P	D+ N+ R	D+ N+ P	D+ R+ P	G+ N+ R	G+ N+ P	G+ R+ P	N+ R+ P	D+ G+ N+ R	D+ G+ N+ P	D+ G+ R+ P	D+ N+ R+ P	G+ N+ R+ P
		Feature Combinations																													

b.		Model Combinations for Analytic Ready LAS																													
Predicted Logs	Density (D)		10.1%	7.9%	9.0%	7.7%					7.4%	8.3%	7.1%	7.1%	6.4%	6.9%							7.0%	6.0%	6.6%	6.1%					5.9%
	Gamma (G)	9.3%		9.0%	9.6%	9.0%		8.7%	8.9%	8.4%				9.1%	8.8%	8.7%				8.5%	8.3%	8.3%				8.6%				8.2%	
	Neutron (N)	6.9%	12.3%		8.2%	7.8%	6.2%		5.9%	5.8%		7.5%	7.3%			6.9%		5.4%	5.5%			5.4%			6.6%				5.2%		
	Resistivity (R)	13.3%	13.4%	11.4%		12.9%	12.7%	11.2%		12.4%	11.3%		12.7%		11.0%		11.1%		12.2%		10.9%			10.9%				10.8%			
	Sonic (P)	7.3%	9.8%	8.1%	8.6%		6.4%	6.3%	6.2%		7.6%	7.7%		7.2%			6.0%	5.7%		5.8%			6.9%				5.5%				
	Normalized RMSE	D	G	N	R	P	D+ G	D+ N	D+ R	D+ P	G+ N	G+ R	G+ P	N+ R	N+ P	R+ P	D+ G+ N	D+ G+ R	D+ G+ P	D+ N+ R	D+ N+ P	D+ R+ P	G+ N+ R	G+ N+ P	G+ R+ P	N+ R+ P	D+ G+ N+ R	D+ G+ N+ P	D+ G+ R+ P	D+ N+ R+ P	G+ N+ R+ P
		Feature Combinations																													

Figure 10 Results of evaluating ARLAS models on wells in the test dataset in the Permian Basin. Each row represents normalised average RMSE to predict different logs colour-coded by the error values. High relative errors are shown in red; low error values are shown in green. (a) Shows the error matrix for raw well logs (without clean-up). (b) Shows the error matrix for the same wells after the clean-up processing.

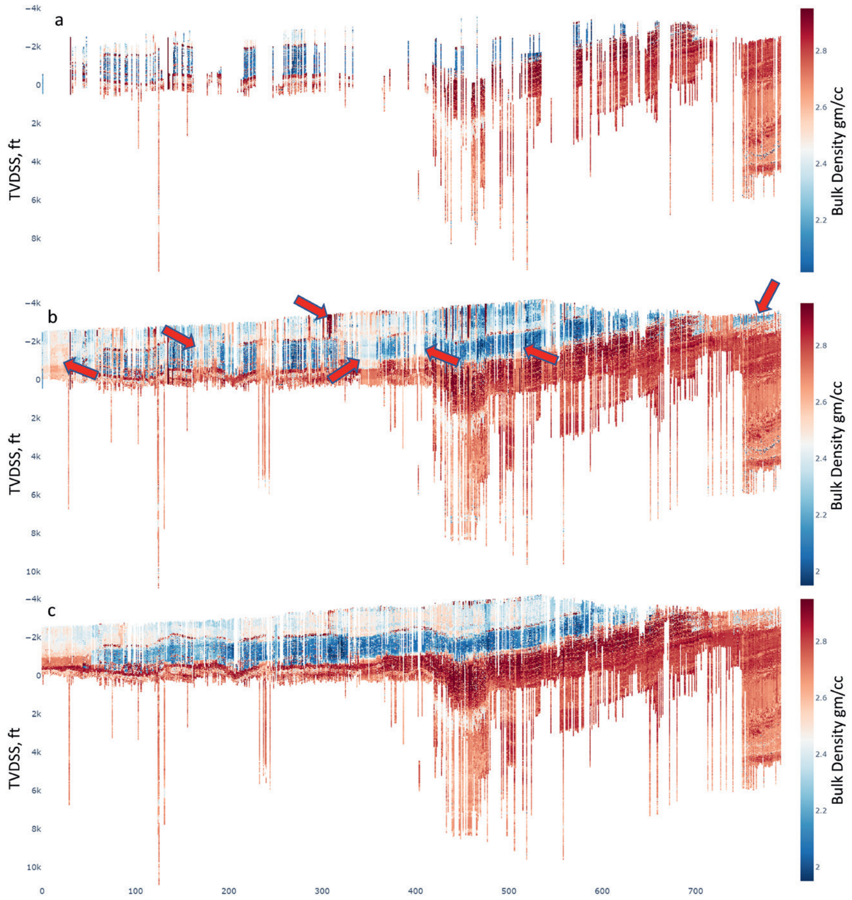

Figure 11 ARLAS modelling for a bulk density cross-section of 800 wells in the Permian Basin. (a) Shows the original measured data available in input LAS files without processing. (b) Shows the same cross-section with ARLAS predictions added to the original input data from (a). (c) Shows the improvements in ARLAS predictions using the cleaned input data. Red arrows indicate areas of most improvement.

Model training

We follow a standard ML approach to model training. Our dataset is split into training and testing sets with a ratio of 75% to 25%. We train all models using wells from the training dataset and assess the model performance on the wells from the testing dataset. We perform five-fold cross-validation during the model training stage to ensure our models are not overfitting to the training data. Cross-validation Berrar (2019) is a standard statistical approach used to estimate the effectiveness of ML models on unseen data. Hyperparameters of LGB models are tuned to achieve the best performance for each basin.

Results

This section summarises the results of ARLAS predictions using wells from the Permian Basin. As explained earlier, we choose which model to apply depending on the available data. These predictions are later combined with recorded logs to fill the existing data gaps and improve data coverage. Figure 9 shows an example of using ARLAS models to fill data gaps in the bulk density log for one of the wells in the Permian Basin. The first track on the left-hand side shows the recorded (measured) bulk density data available in a well (black). The second track shows the prediction of bulk density using ARLAS models (red). The third track plots both curves to show that prediction matches the recorded data and extends beyond the original log's scope. The last track on the right-hand side shows the final curve that combines the original data and predictions in the intervals where the original data is absent. This is the final product of the ARLAS workflow.

We calculated the prediction error in a blind test comprising 679 wells and evaluated the model performance in intervals where both data and prediction exist. Figure 10 summarises the average normalised RMSE (Root Mean Squared Error) for different models and feature combinations using wells in the Permian Basin. We normalise RMSE error by the average difference between log min and max values across all wells in the testing set. Figure 10a shows errors for different models trained and applied to wells without data clean-up. Figure 10b shows errors calculated using the same dataset, but the wells were processed using our clean-up pipeline before model training and inference steps.

The clean-up process reduces the error across all models, the bulk density log and the compressional sonic logs being the ones that benefit the most from it. Additionally, we can improve model performance by including more features in the prediction. Models with more features outperform the ones with fewer features. This is a consideration when models are selected for filling the gaps in the existing dataset. The model with the most features available will be used for each depth sample.

Figures 11 and 12 show the result of ARLAS modelling for a cross-section of 800 wells in the Permian Basin for bulk density (Figure 11) and compressional sonic (Figure 12) logs. Both figures show (in panel a) the original measured data available in input LAS files without any clean-up-up processing. Panel b shows the identical cross-sections with ARLAS predictions with the input data without clean-up. Panel c shows the improvements in ARLAS predictions using the cleaned input data. The difference in prediction quality between Panels b and c stresses the

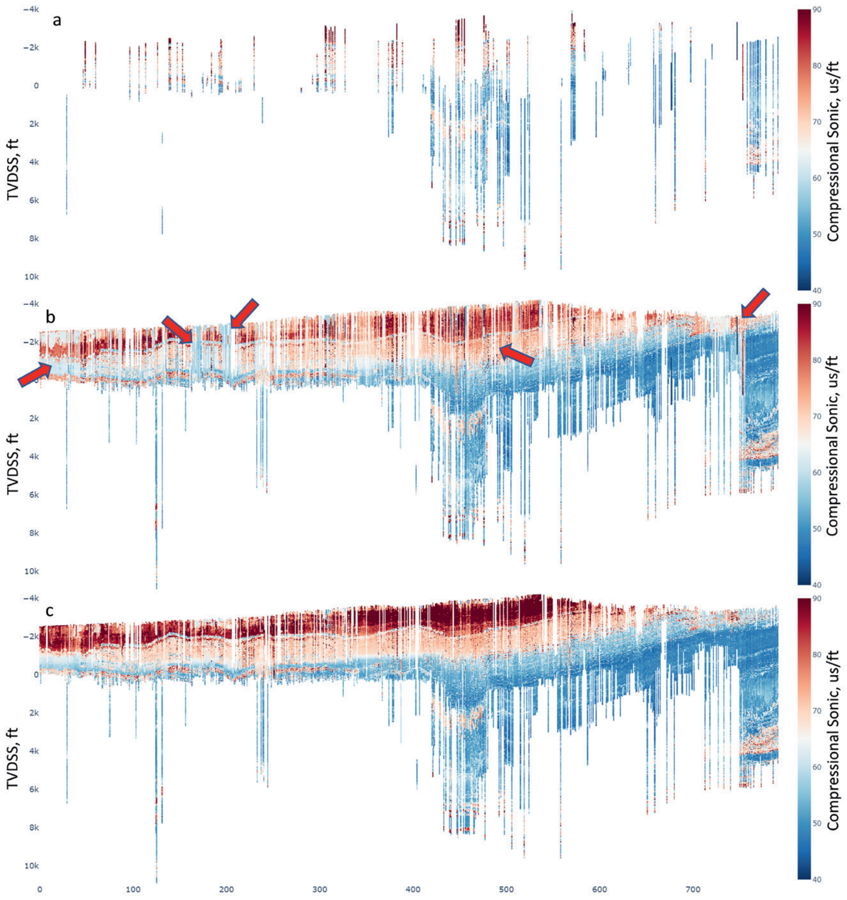

Figure 12 ARLAS modelling for a compressional sonic cross-section of 800 wells in the Permian Basin. (a) shows the original measured data available in input LAS files without processing. (b) Shows the same cross-section with ARLAS predictions added to the original input data from (a). (c) Shows the improvements in ARLAS predictions using the cleaned input data. Red arrows indicate areas of most improvement.

importance of an automated clean-up process before applying ML to log data on a basin scale.

Conclusion

We have shown the importance of data cleaning in well logs prediction using ML algorithms to obtain superior quality results at a basin scale. Our data clean-up pipeline is simple, with minimal expert user interaction. It addresses errors in LAS and logs headers, removing or editing inferior quality data, normalising logs for tool/environmental effects, splicing, and merging logs from different runs, and depth shifting for depth alignment.

Using examples from the Permian Basin in the US, we have proved that the ML models produce outstanding results if superior quality data is supplied for the model training and inference. The ML models can achieve a prediction quality of 90% to 95% if trained on clean data. Model prediction accuracy is higher for neutron porosity, bulk density, and compressional sonic logs.

Acknowledgements

The authors would like to thank Michael Hollerbach, Matt Mayer, Sribharath Kainkaryam, Ben Lasscock, Cable Warren, and Altay Sansal for their suggestions and contributions.

References

Berrar, D. [2019]. Cross-validation. *Encyclopedia of Bioinformatics and Computational Biology*, Academic Press, Volume 1, p. 242-245, https://doi.org/10.1016/B978-0-12-809633-8.20349-X.

Breiman, L. [2001]. Random forests, *Machine learning*, **45**(1), 5-32.

Chen, T. and Guestrin, C. [2016]. Xgboost: A scalable tree boosting system, In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 785-794.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y. [2017]. Lightgbm: A highly efficient gradient boosting decision tree, In *Advances in neural information processing systems*, p. 3146- 3154.

Li, P., Qiang, W. and Burges, C.J. [2007]. Mcrank: Learning to rank using multiple classification and gradient boosting, *Advances in Neural Information Processing Systems 20.*

Machado, F.P. [2003]. Communication and memory efficient parallel decision tree construction, *Proceedings of the 2003 SIAM international conference on data mining*, p. 119-129.

Ranka, S. and Singh, V. [1998]. CLOUDS: A decision tree classifier for large datasets, *Proceedings of the 4th Knowledge Discovery and Data Mining Conference*.

Shier, D.E. [2004]. Well log normalization: Methods and guidelines, *Petrophysics*, **45**(3), 268-280.