Predicting sonic and density logs from drilling parameters using temporal convolutional networks

Robert Smith[1], Andrey Bakulin[1], Pavel Golikov[1], and Nasher AlBinHassan[1]

https://doi.org/10.1190/tle41090617.1

Abstract

Sonic and bulk density logs are crucial inputs for many subsurface tasks including formation identification, completion design, and porosity estimation. Economic and operational concerns restrict the acquisition of these logs, meaning the overburden and sometimes entire wells are completely unlogged. In contrast, parameters that monitor drilling operations, such as weight on bit and torque, are recorded for every borehole. Previous studies have applied supervised machine learning approaches to predict these missing logs from the drilling parameters. While the results are promising, they often do not investigate the importance of different features and the corresponding practical implications. Here, we explored the feasibility of predicting compressional slowness and bulk density logs using various combinations of formation markers, gamma-ray logs, and drilling data recorded at the rig. Our tests utilized a temporal convolutional network to allow the model to learn from sequences of input features. Bayesian-based hyperparameter tuning found the optimum set of parameters for each experiment before producing the final log predictions. Finally, a permutation feature importance analysis revealed which input variables contributed most to the outputs. Although drilling parameters contain some insight into the mechanical rock properties, we found that they cannot produce the high-quality log predictions required for many tasks. Supplementing the drilling parameters with a gamma-ray log and formation data produces good-quality log predictions, with the additional inputs helping to constrain the model outputs.

Introduction

Geoscientists and drilling engineers rely on sonic (compressional and shear slowness) and bulk density logs for numerous subsurface tasks. In addition to forming the basis of reservoir characterization studies, these geophysical logs enable the derivation of geomechanical parameters such as compressive strength. However, wireline log acquisition involves placing sensors in the borehole to measure the physical properties of the rock, making them expensive and logistically challenging to acquire. As a result, logging is usually restricted to the reservoir interval, with many wells having no measurements at all. Conversely, drilling parameters (such as the rate of penetration [ROP], weight on bit [WOB], and torque [TOR]) are routinely recorded for the entire length of every well.

Intuitively, we expect some relationship between the drilling parameters and the mechanical rock properties. For example, to achieve a constant ROP, more downward force or TOR is required to drill through rocks with higher compressive strength. Although models for ROP have been developed based on controlled laboratory experiments (e.g., Teale, 1965; Hareland and Nygaard, 2007), the conditions encountered in the field are more complex. Drilling parameters often vary significantly between adjacent wells due to factors such as inefficient drilling, use of different rig equipment, and actions of rig operators.

Machine learning methods may help uncover these complex relationships to enable log prediction from data acquired while drilling. Such a data-driven solution would provide engineers with additional information where log data would usually be unavailable. Several studies have reported on this topic, including Gan et al. (2019), who applied a neural network to generate synthetic logs in the Gulf of Mexico. Meanwhile, Kanfar et al. (2020) used a technique based on a temporal convolutional network (TCN) (Bai et al., 2018) to predict sonic and porosity logs from drilling parameters. The TCN is a deep neural network used for sequence modeling tasks such as translating text or audio from one language to another. The use of input sequences allows the TCN to learn from trends in the input time (or depth) series, rather than relying on point measurements. The authors showed promising results on 25 ft sections. However, they did not produce predictions for the entire well or consider practical scenarios where different input data may be available.

We build upon the existing work by investigating the feasibility of a TCN to predict compressional slowness and density logs using different sets of input features. Table 1 shows three scenarios that span the best to worst case in data availability, allowing us to assess the potential range of the predicted log quality. Ideally,

Table 1. Different scenarios investigated in this study.

Scenario	Real time?	Input data	Potential use cases
Log prediction in the reservoir zone	No	Drilling parameters, gamma-ray log, and formation top picks	Geophysical characterization study
Real-time log prediction	Yes	Drilling parameters and gamma-ray log	Drilling optimization
Log prediction in the overburden	Either	Drilling parameters only	Seismic imaging, seismic to well tie, and formation identification

[1]Saudi Aramco, EXPEC Advanced Research Center, Dhahran, Saudi Arabia. E-mail: robsmith155@gmail.com; a_bakulin@yahoo.com; pavel.golikov@aramco.com; nasher.benhasan@aramco.com.

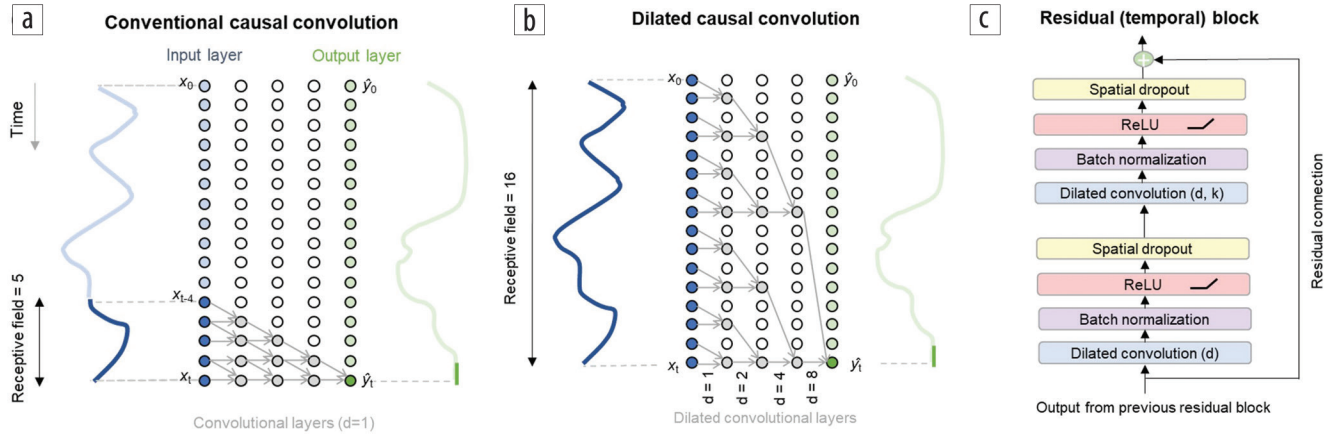

Figure 1. Illustration of the receptive field for a stack of (a) conventional convolutional layers and (b) dilated convolutional layers used in a TCN. (c) The TCN replaces standard convolutional layers with temporal blocks, which may include multiple dilated convolutional layers. Modified from Smith et al. (2022).

the drilling parameters are supplemented with a gamma-ray log and formation top picks (scenario 1). However, obtaining detailed formation picks may require cuttings analysis or log data. Hence, they may only be available for non-real-time use cases such as geophysical characterization. Conversely, drilling engineers need information in real time to optimize the drilling program. If acquired, real-time measurement-while-drilling (MWD) or logging-while-drilling (LWD) gamma-ray logs can be combined with the drilling dynamics to make real-time slowness and density estimates (scenario 2). The last case uses only drilling data as input (scenario 3), which allows us to determine what information the data contain about the rock properties. In addition to producing full log predictions for each scenario, we conducted a feature importance analysis to identify the variables with the most significant contribution to the results.

Background

This section highlights several related studies where machine learning was used to predict subsurface properties from drilling parameters. We then provide an overview of the TCN architecture, which forms the basis of the method used in this research.

Log prediction using drilling parameters. Numerous examples of machine learning applied to predicting logs from other wireline logs exist in the literature. However, only a few cases use drilling parameters as inputs to the model. For example, Moazzeni and Haffar (2015) trained a fully connected neural network for real-time lithology identification from drilling parameters. In the Gulf of Mexico, Gan et al. (2019) applied a neural network to generate synthetic sonic and density logs from drilling parameters and mud-logging data. Meanwhile, Glubokovskikh et al. (2020) incorporated near-bit vibrations recorded by downhole accelerometers to distinguish between the effect of rock-cutting and drill-string noise on the dynamic drilling parameters.

Sequence modeling describes a group of machine learning algorithms that convert an input sequence into a target sequence in another domain. Several authors have used sequence modeling approaches to predict logs from drilling parameters, including Osarogiagbon et al. (2020), who tested numerous machine learning methodologies for predicting gamma-ray logs. They reported that TCNs and simple recurrent neural networks (RNNs) produced the best results. Similarly, Kanfar et al. (2020) combined an inception-based convolutional network with a TCN for real-time sonic log prediction.

Temporal convolutional networks. The TCN is an extension of 1D convolutional neural networks (CNNs), incorporating a stack of modified 1D convolutional layers. In conventional 1D convolution, the network's receptive field (i.e., the input region that affects the model's output) grows linearly with the number of convolutional layers (Figure 1a). As a result, regular convolution is unsuitable for many sequential modeling tasks that require access to a long history of the input features. Dilated convolutions, first introduced in WaveNet (Oord et al., 2016), are simple modifications that skip input values to apply the filter over a larger area (Figure 2b). Here, the dilation factor (d) controls the number of omitted values. For example, a dilation factor of $d = 1$ is equivalent to regular convolution, while $d = 4$ means that the filter uses every fourth value of a series. Figure 1b shows the dilation factor increasing exponentially (d = 1, 2, 4, 8…). This results in the receptive field also growing exponentially for a fixed filter width. Dilated convolutions allow a TCN to use a much larger receptive field at the same computational cost as a regular 1D CNN.

Every convolutional layer in the TCN architecture is implemented as a temporal (or residual) block (Figure 1c). These blocks can include multiple dilated convolutional layers (using the same dilation factor and filter width), each followed by a nonlinear activation function such as a rectified linear unit (Nair and Hinton, 2010). The block may also include regularization in the form of spatial dropout (Srivastava et al., 2014) and weight normalization (Salimans and Kingma, 2016) layers. The final component is the residual connection (He et al., 2016), which helps overcome the issue of vanishing gradients often encountered in deep neural networks.

Bai et al. (2018) showed that the TCN architecture outperforms RNNs in a wide range of sequence modeling tasks. The TCN also has several practical advantages, including parallelizable computations for faster model training and reduced susceptibility to exploding and vanishing gradients. Recently, TCNs have found geophysical applications in areas such as seismic inversion (Mustafa

et al., 2019; Smith et al., 2022), automated well tie (Nivlet et al., 2020), time-lapse cross-equalization (Alali et al., 2020), and log prediction (Kanfar et al., 2020).

Data overview and preparation

This study used 13 vertical wells where both drilling parameters and wireline logs were recorded. Each well contains approximately 1000–2000 ft of data from a series of carbonates and sandy shales, providing 40,000 individual data points. Table 2 summarizes the input features available for model training.

Preprocessing. An essential step of any machine learning project is preprocessing the data to a form that is suitable for training a model. First, we replaced small gaps in the data using interpolation and clipped values outside of the allowable range (Table 2). Next, one-hot encoding was applied to the categorical features (e.g., bit type), where a binary column was added for each unique category.

Although deep neural networks can learn complex relationships between input features, the inclusion of engineered features can help a model converge to a useful solution. Therefore, we incorporated two additional features related to drilling efficiency that have been found to be predictive of the sonic and density logs (Glubokovskikh et al., 2020). The first feature is mechanical specific energy (MSE), which is the mechanical energy required to drill a unit volume of rock (Teale et al., 1965). Ideally, the MSE is close to the unconfined compressive strength, meaning that crushing the rock consumes almost all of the input energy. The second feature is a parameter designated as SQ. Lamik-Thonhauser et al. (2018) showed that SQ helps separate different lithologies. The SQ parameter is computed from the following equation:

$$Sq = 4\sqrt{\pi}\,\frac{\mathrm{WOB}\times\mathrm{RPM}}{\sqrt{A_B}\times\mathrm{ROP}}, \qquad (1)$$

where A_B corresponds to the cross-sectional area of the drill bit. Finally, we rescaled the data using normalization so that each input feature covered the range [0,1].

Figure 2a displays an example of processed data from one of the wells, showing the wireline logs, drilling parameters, and formation markers. We used nine of the 13 wells for model training, with the remaining four split between the validation and blind-test data sets. These were carefully selected to ensure similar parameter distributions between the three data sets (Figure 2b).

Method

Table 3 outlines the experiments investigated in this paper, including three data settings used to test compressional slowness and bulk-density predictions. We trained a separate model for each output for the purpose of this study to facilitate the feature importance analysis. However, a more practical solution would be to train a model that can predict multiple logs. The first scenario uses all available input features including formation data, gamma-ray log, and drilling parameters. Scenario 2 tests the impact of dropping formation data from the inputs, relying only on gamma-ray and drilling parameter data. In the final setting (scenario 3), we include only the drilling parameters to investigate what information they contain about the mechanical rock properties.

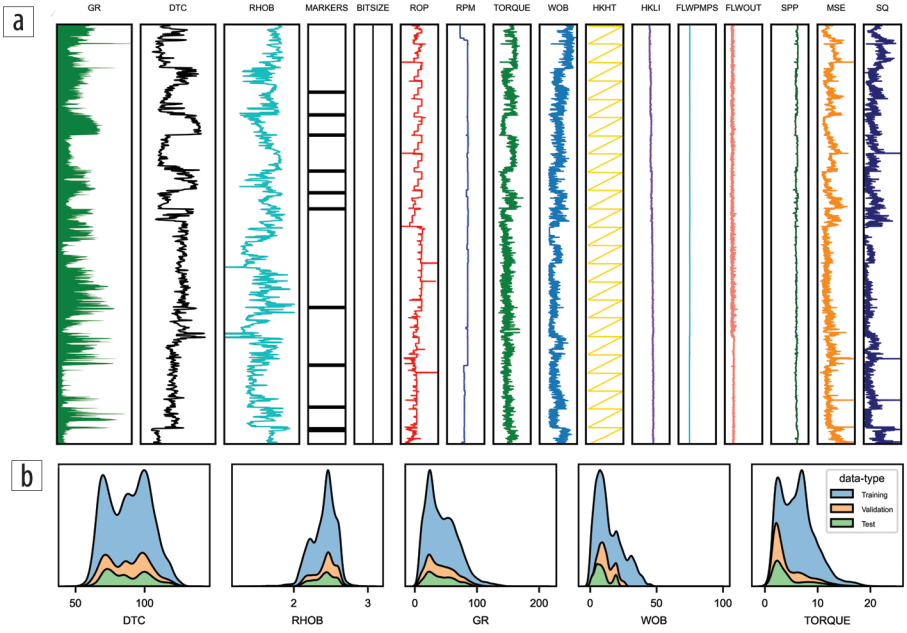

Figure 2. (a) Input features and target logs for an example well and (b) feature distributions for the training, validation, and blind wells.

Table 2. Summary of input features and target variables used in this study.

Name	Mnemonic	Units	Acceptable range	Type
Rate of penetration	ROP	ft/hour	0–300	Continuous
Weight on bit	WOB	klbf	0–100	Continuous
Torque	TOR	kft.lbf	0–30	Continuous
Revolutions per minute	RPM	1/minute	0–200	Continuous
Standpipe pressure	SPP	psi	0–5000	Continuous
Hook height	HKHT	ft	0–100	Continuous
Hook load	HKLI	klbf	0–500	Continuous
Flow in	FLWPMPS	galUS/minute	0–2000	Continuous
Flow out	FLWOUT	%	0–100	Continuous
Bit size	BITSIZE	inch	8–16	Categorical
Bit type	-	-	-	Categorical
Gamma ray	GR	API	0–200	Continuous
Formation	-	-	-	Categorical
Compressional sonic	DTC	µs/ft	110–200	Continuous
Bulk density	RHOB	g/cc	2.2–2.8	Continuous

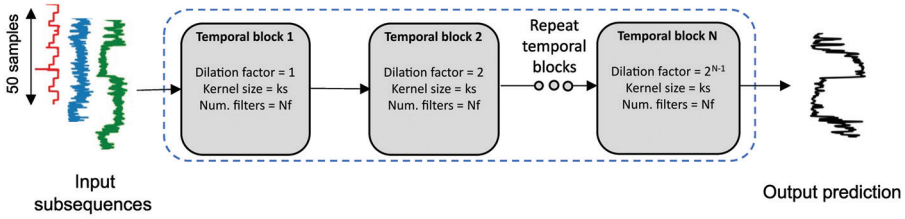

Figure 3. Input subsequences passed into the TCN model to predict a section of the log.

Table 3. Summary of experiments.

Experiment	Target	Input data
1a	DTC	Formation markers + drilling parameters + gamma ray
1b	Density	
2a	DTC	Drilling parameters + gamma ray
2b	Density	
3a	DTC	Drilling parameters only
3b	Density	

Table 4. Hyperparameter search space.

Hyperparameter	Search space	Comments
Learning rate	$1e^{-6} - 1e^{-1}$	Log sampling
Number of temporal blocks	[2,3,4,5,6]	
Number of convolutional filters per layer	10–100	Uniform sampling
Dropout rate	0–0.8	Uniform sampling
Filter size (k)	[2,3,4,5]	
Activation function	[ReLU, Leaky ReLU]	

We trained a separate TCN model for each experiment in Table 3. The exact architecture of the TCN (e.g., the number of temporal blocks) depends on the choice of model hyperparameters. This is covered in the following section. As in Nivlet et al. (2020), the data from each well were split into smaller subsequences to provide enough samples for training (Figure 3). We used a 50-sample (25 ft) sliding window. We felt that this was geologically meaningful and would enable the network to find patterns in the noisy drilling data. A total of 5000 sequences from the training wells were used to train each model, with 1000 sequences used to score the model performance based on the validation data.

During training, the input subsequences are propagated forward through the network to output the corresponding subsequence of predicted compressional slowness or density ($\hat{y}$). The mean square error, a differentiable loss function, was then used to measure the error between the predicted and measured logs. We used the AdamW optimizer (Loshchilov and Hutter, 2017), a variant of the famous Adam optimizer (Kingma and Ba, 2014) with improved weight decay, to update model weights during backpropagation.

Hyperparameter tuning. Model hyperparameters are variables that define network architecture (e.g., number of convolutional layers) and how the model learns (e.g., learning rate). Finding an optimal set of hyperparameters is often crucial to obtaining good network performance. We implemented a Bayesian-based hyperparameter search using the Optuna open-source package (Akiba et al., 2019) to determine the optimum set of values. Other alternatives include grid and random searches. However, these methods treat hyperparameter sets independently, and this typically results in having to run more trials to find the best solution.

Table 4 shows the hyperparameter search space explored for the TCN model, including the filter (kernel) size, number of dilation layers, and model learning rate. We ran 1000 trials for each experiment, where each trial trained a model using a unique set of hyperparameters. The first 100 trials utilized random sampling to initialize the search. Model performance was measured using the root mean square error (RMSE) of the validation well predictions. The tree Parzen estimator (TPE) algorithm (Bergstra et al., 2011) took over for the remaining 900 models. It used Bayesian optimization to select a set of promising hyperparameters based on the performance of the previous runs. The goal of the objective function was to minimize the validation RMSE. Each trial was run up to 100 epochs, where one epoch means the algorithm passed through the entire training data set. We implemented several strategies to reduce the training time for the hyperparameter search. First, early stopping was applied if the validation performance did not improve for 15 consecutive epochs. Second, median pruning stopped unpromising models early if the intermediate performance of the trial was worse than the median of previous runs at the same epoch.

Log prediction. Postprocessing was applied to construct the full log predictions. The best model from each hyperparameter search (as determined by the lowest validation RMSE) was used to predict the logs for training, validation, and blind wells. When comparing different model architectures, one should avoid comparing the test data to prevent biasing results. However, the objective of this study was to compare different data scenarios to understand the effect on log prediction quality. Therefore, we feel it is appropriate to include the test results for each scenario in this case.

Model interpretation. The permutation feature importance is a relatively simple model interpretation technique that was first introduced for random forests (Breiman, 2001) and later extended to other algorithms by Fisher et al. (2019). The permutation feature importance is defined as the change in model score after randomly shuffling a particular input feature. Mixing a feature that the model has learned to be important breaks the relationship with the target output, resulting in a significant increase in prediction error. Although it has some issues, such as not separating the effects of correlated features, it is a simple and intuitive measure of feature importance. A viable alternative is SHapley Additive exPlanations (SHAP) by Lundberg and Lee (2017), which aims to compute the contribution of each feature on a prediction.

To use this approach for a TCN, we randomly shuffled a given feature (e.g., gamma ray) for each well before extracting the subsequences and recomputing the log prediction. We repeated this process 10 times for each feature and used the mean change in

DOI:10.1190/tle41090617.1

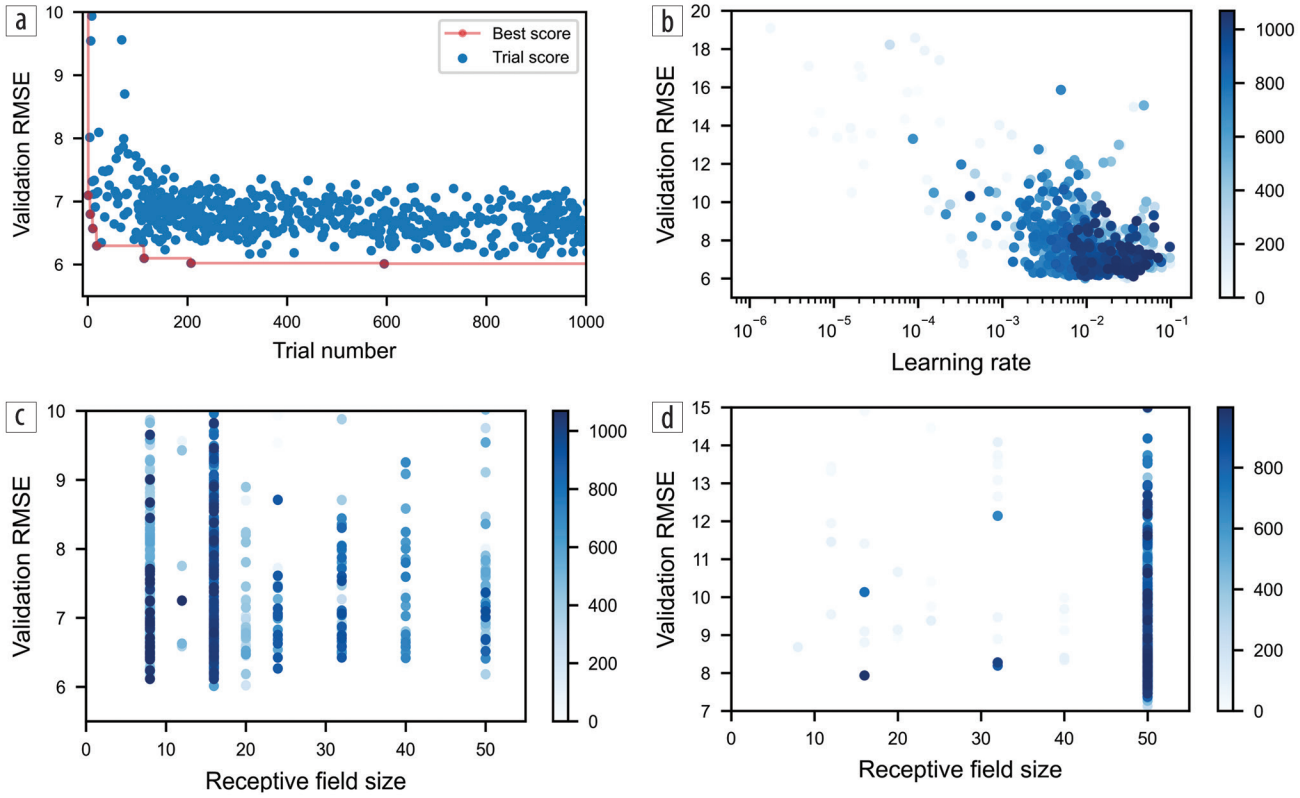

Figure 4. Observations from the hyperparameter searches. (a) TPE search history for experiment 1a shows the improvement in validation score (red line) with the number of trials, while (b) shows the learning rate convergence during the search. Here, color represents the trial number, with darker blue points toward the end of the search. The effect of the receptive field on the results of experiments 1a and 2a are shown in (c) and (d), respectively.

RMSE as the importance measure. This produced a global average of the feature contribution. To better understand the impact of input features on local predictions, we also computed a sample-based permutation feature importance using the absolute change in the log prediction as a local measure of importance.

Table 5. Summary of the best model hyperparameters from the TPE search of each experiment.

Experiment	Best RMSE	Learning rate	No. filters per layer	Dropout rate	No. dilation layers (*d*)	Kernel size (*k*)	Receptive field size
1a	0.068	0.0096	15	0.065	3	2	16
1b	0.057	0.0047	35	0.358	5	5	50
2a	0.081	0.0076	50	0.218	6	4	50
2b	0.063	0.0026	20	0.061	3	5	40
3a	0.104	0.0351	35	0.021	5	5	50
3b	0.076	0.0081	55	0.017	6	4	50

Results

This section presents observations from the hyperparameter searches and log predictions produced by the corresponding best models. We also include feature importance analysis to aid the interpretation of the results.

Hyperparameter tuning. Figure 4 shows observations made from the TPE searches of the hyperparameter space. In Figure 4a, we see the search history for experiment 1a. This shows the best validation score for each trial (a trained model with a unique set of hyperparameters). We observed higher RMSE variance for the first 100 trials, where the process randomly selected hyperparameters. After this point, the TPE algorithm took over and focused only on the most promising parameter combinations. Figure 4b illustrates the convergence to the optimum learning rate for the same experiment, where color represents the trial number.

Figures 4c and 4d demonstrate the effect of the receptive field (a function of kernel size and the number of temporal blocks) on the results. In the case of experiment 1a (prediction of compressional slowness using all input features), performance is less affected by the size of the input available to the model. Here, the best models use a receptive field of 10 to 20 samples. On the other hand, the omission of formation data from the input features (experiment 2a) means that the model must rely more on patterns in the remaining data (Figure 4d). The hyperparameter search converges on models with the largest receptive field size (50 samples) in this scenario.

Table 5 summarizes the results of the hyperparameter search for each experiment including the best set of hyperparameters and the corresponding validation RMSE score. In the case of density and compressional sonic, models trained using all of the input data (scenario 1) produced the best results, while using drilling parameters alone generated the worst outcomes.

Log predictions. Figure 5 shows the predicted compressional sonic logs produced by the best model for each experiment. Here,

DOI:10.1190/tle41090617.1

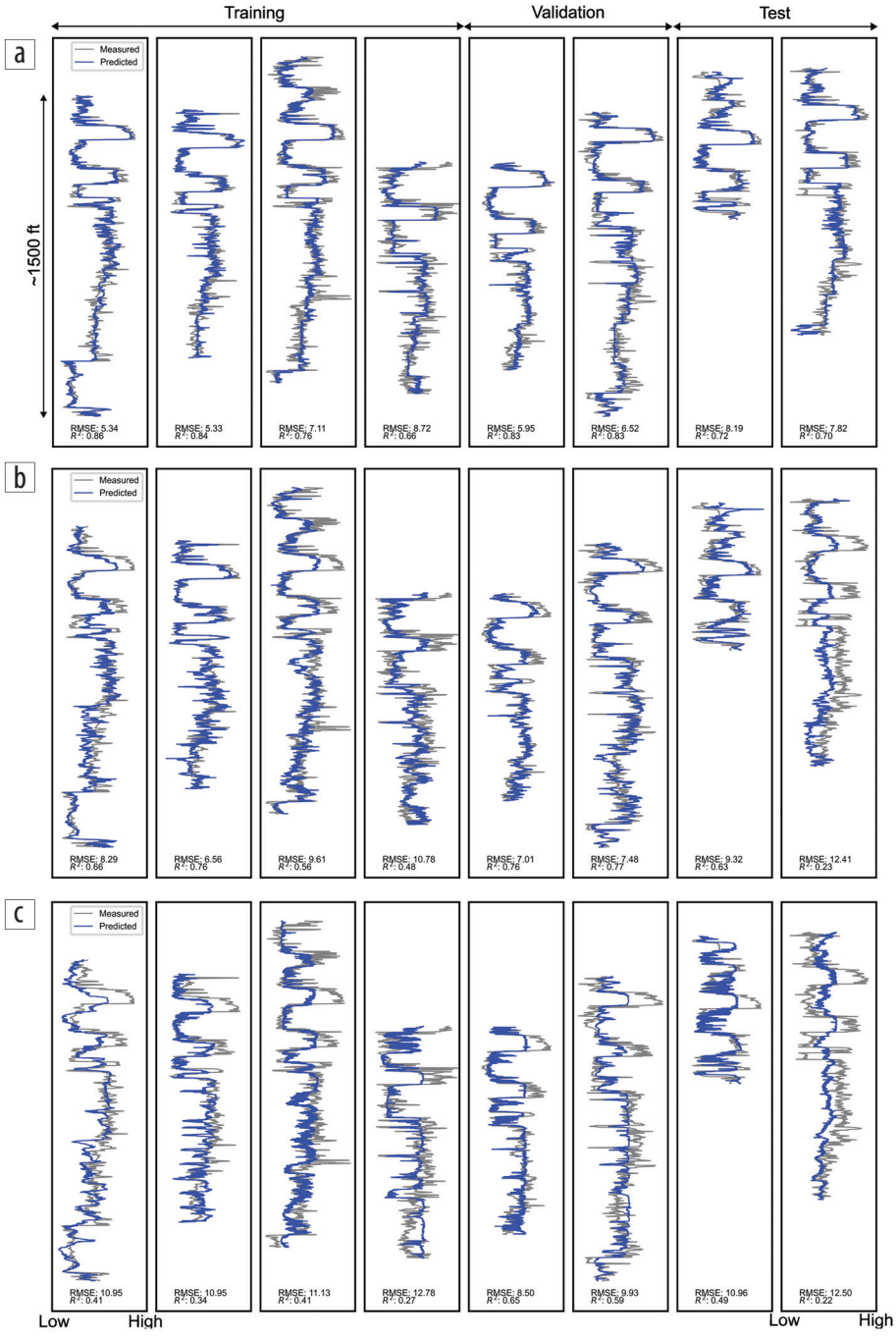

Figure 5. Measured versus predicted compressional sonic logs for (a) a model using all input features, (b) a model using gamma-ray and drilling parameters, and (c) a model using only drilling parameters.

we include a sample of the training data (ranging from worst to best quality) along with the validation and blind wells. In addition to the RMSE, we show the coefficient of determination (R^2) to indicate whether the prediction is useful. An R^2 of 1 implies a perfect prediction, while an R^2 of 0 is equivalent to naively predicting the mean value (i.e., it accounts for none of the data variance and is not a useful solution).

The drilling parameters supplemented with the gamma-ray log and formation data produced the best results (Figure 5a). Here, the model does an excellent job of predicting the general trends of the sonic log, although it is unable to capture all of the fine details. We observe worse performance when formation data are omitted from the inputs (Figure 5b), although the level of decay varies from well to well. It is unclear why the performance of some wells deteriorates more than others, even when comparing validation and blind wells. It could be related to how well the drilling parameters relate to the rock properties. Despite the issues associated with the drilling parameters, Figure 5c shows that they contain sufficient information to capture the major changes in the compressional sonic log.

We observe similar trends in the bulk density predictions shown in Figure 6. In general, we see an overall reduction in performance when we go from using all input features (Figure 6a) to dropping the formation picks (Figure 6b). In some local regions, we sometimes observe improvements without the formation data, such as in the first blind well. The results using only the drilling parameters (Figure 6c) show mixed outcomes. While the model is still able to predict general trends for some wells, in many cases the results are effectively unusable. The density logs tend to not have the sharp contrasts that we observe in the sonic logs. Perhaps this relationship is higher order and more susceptible to issues with the drilling parameters.

Feature importance. Figure 7 displays the global permutation feature importance for the compressional sonic predictions using the validation and blind wells. The value here is the change in metric score (RMSE in this case) after randomizing the values of that feature and recomputing the log prediction. The formation data are most important to the models in scenario 1 (Figures 7a and 7b), closely followed by the gamma-ray log. Omission of the formation data results in gamma ray becoming the predominant predictive feature, supplemented by the TOR and SQ inputs (Figures 7c and 7d). The SQ parameter appears to be a valuable engineered feature for predicting compressional slowness, particularly for a model trained using only drilling parameters (Figures 7e and 7f).

When available, the formation data are also the most critical feature for the density log predictions (Figures 8a and 8b). Gamma ray is important when formation data are omitted (Figures 8c and 8d); however, the impact of the drilling parameters is less clear. We see very little change in the metric scores for the model

trained using only drilling parameters (Figures 8e and 8f). These results suggest that the relationship between density and the drilling parameters is more complex and likely more susceptible to drilling data quality.

The results in Figures 7 and 8 are informative, but they only show the average behavior for a set of wells. Figure 9 displays the sample-based permutation feature importance for one validation and one blind well, predicting compressional slowness. This allows us to evaluate the contribution of several features at each depth sample. Several input features are shown, with the background color representing the change in absolute error when the column is randomized (average of 10 iterations). Darker colors indicate higher importance. Figures 9a and 9b are the results from scenario 1 using all input features. We do not show formation data here because categorical data are more challenging to represent. Still, we observe clear trends between the gamma-ray log and compressional slowness that the model has learned to use. The increased importance of TOR is apparent in Figures 9c and 9d when we omit formation data. Here, we see that an increase in TOR generally corresponds to a decrease in compressional slowness (i.e., stiffer rock material), particularly in the case of the validation well (Figure 9c). The SQ_parameter appears to be very important for the model trained using only the drilling parameters (Figures 9e and 10f). One should be careful when interpreting the impact of features showing little variation. For example, a fixed bit size is used in the validation well, so the change in the performance metric will always be zero after permuting this column. Locally, it can have an effect, as demonstrated by the cored sections of the blind well where a smaller bit was used.

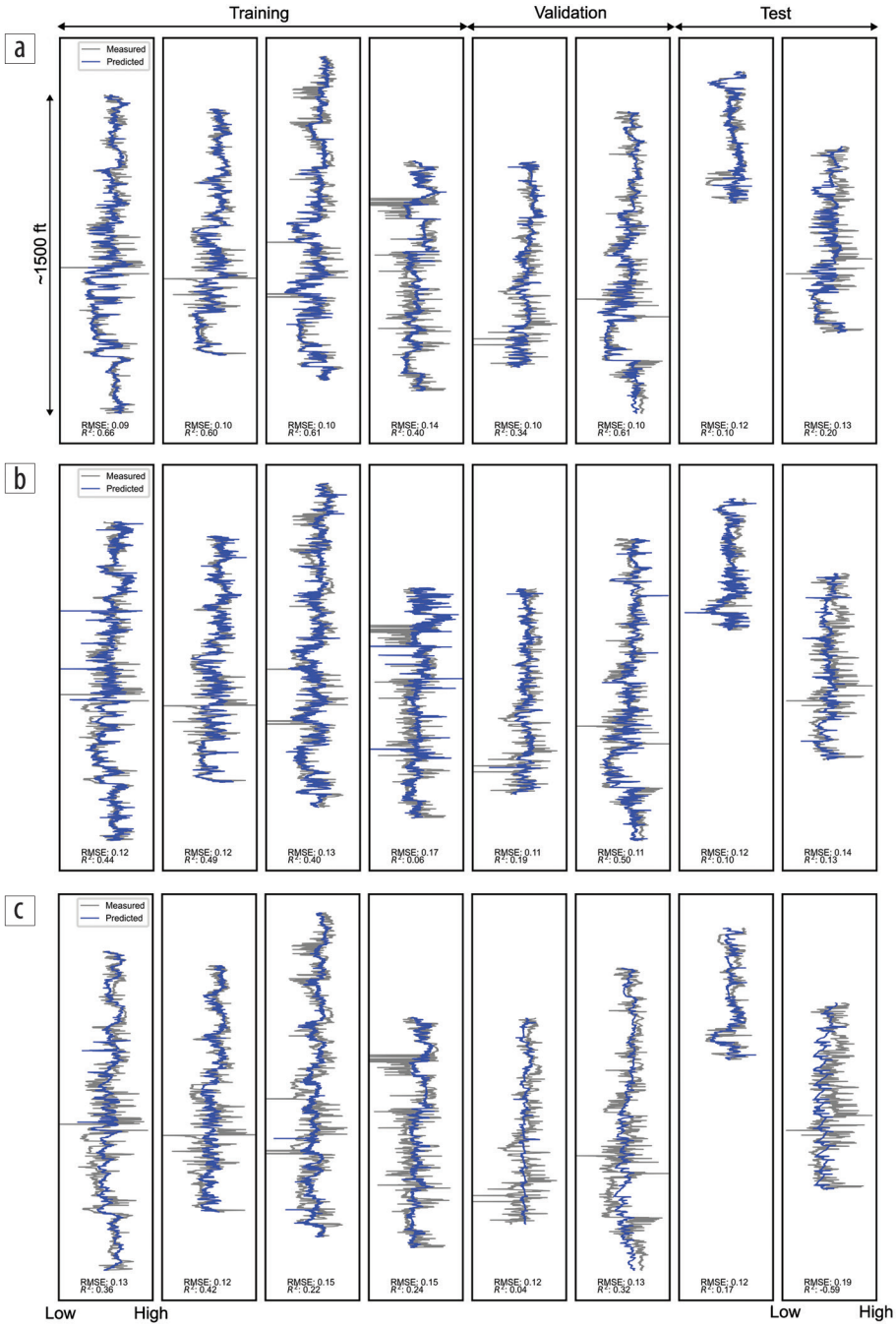

Figure 6. Measured versus predicted density logs for (a) a model using all input features, (b) a model using gamma-ray and drilling parameters, and (c) a model using only drilling parameters.

Discussion

We have demonstrated the potential for predicting compressional sonic and bulk density logs using different combinations of input features. Although the three cases studied in this paper are not an exhaustive set of scenarios, they cover the range of possible outcomes. Models trained using formation data, gamma-ray log, and drilling parameters produced the best results, doing an excellent job of predicting the general trends. While the model may not capture high-frequency details, it can still provide valuable information for unlogged wells. For example, the outputs may assist non-real-time use cases such as geophysical characterization studies.

Feature importance analysis revealed formation data as the most significant input feature for the trained models. Knowledge of the formation constrains the possible output range, making it easier for the model to learn. However, picking formation tops is typically a laborious process, meaning they are generally unavailable for real-time applications such as drilling optimization. Predicting the logs using only the gamma-ray and drilling

parameters produced reasonable log quality in this study, although accuracy is reduced compared to when formation data are available. Note that the LWD or MWD gamma-ray log required for drilling applications uses a sensor placed up to 15–30 m behind the drill bit, so true real-time predictions may not be possible. In addition, the recorded data require correction for the borehole conditions. This also requires further consideration.

Models trained using drilling parameters alone may have some predictive power. While unsuitable for detailed characterization work or drilling program design, we found the model predicted the major changes in compressional slowness that could aid tasks such as formation top identification. The mixed results using only drilling parameters as input are not surprising. Many factors can affect these measurements including drilling inefficiency, poorly calibrated sensors, and operator decision making. For example, much of the input energy is wasted when drilling becomes inefficient. So, drilling parameters are no longer directly related to the rock properties. Several possible improvements could help in this case. First, using a much larger and more diverse data set could assist the model in revealing relationships within the data. Second, acquiring new data MWD may help remedy some issues. For example, Glubokovskikh et al. (2020) supplemented the drilling parameters with near-bit vibrations to predict acoustic logs. We could also include the relative depth from a known casing point when detailed formation data are absent to provide some positional encoding to the model.

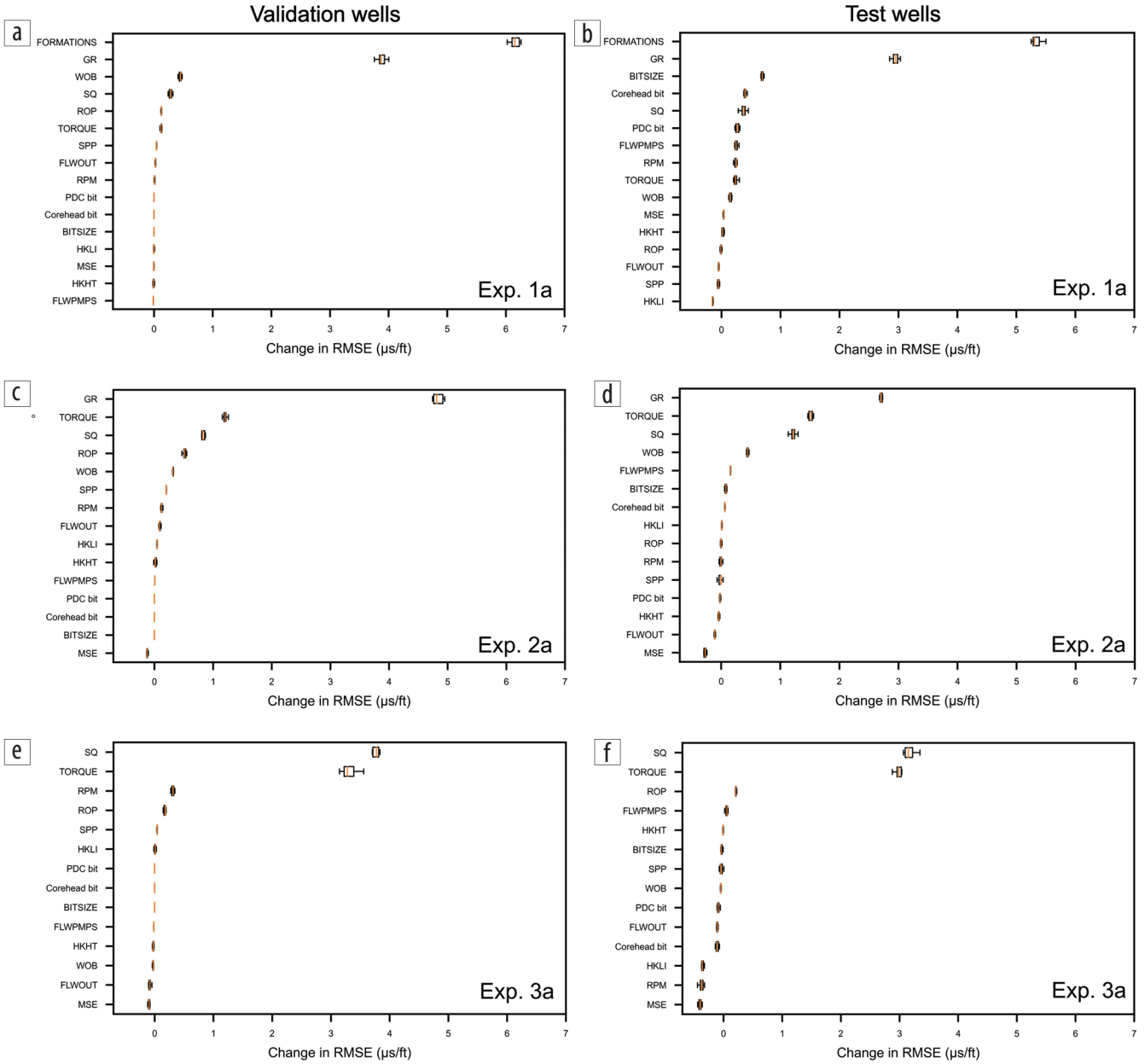

Figure 7. Global permutation feature importances for the compressional slowness predictions. The feature importances using the model trained on all input features (experiment 1a) for the (a) validation well and (b) blind well. Similarly, (c) and (d) show the corresponding results for a model trained without formations (experiment 2a), while (e) and (f) are for the model trained using only drilling parameters (experiment 3a).

Predicting acoustic and density logs in the overburden, where geophysical log coverage is often minimal, would benefit drilling engineers and geoscientists. The additional data could enable improved geomechanical models to aid in drill planning while also helping to identify and understand seismic imaging issues. However, the limited log availability is also a data-science challenge because we may have insufficient data to train and validate a model. One possibility for further investigation is whether training a model on the deeper subsurface can be transferred to the overburden (possibly with some fine tuning).

This study used a TCN to predict the log data. In our experience, algorithms that predict based on single data samples (such as fully connected neural networks or random forests) produce comparable results when formation data are available. The TCN has a significant advantage when this is not the case. This is because using sequential inputs enables the model to learn from spatial trends in the data. The hyperparameter search results show that the models trained without formation data converged to solutions with the largest receptive field. This was less important when formations were available.

Conclusions

A TCN was used to predict compressional sonic and bulk density logs using different combinations of data acquired while drilling. We achieved the best results when the drilling parameters were supplemented with a gamma-ray log and formation

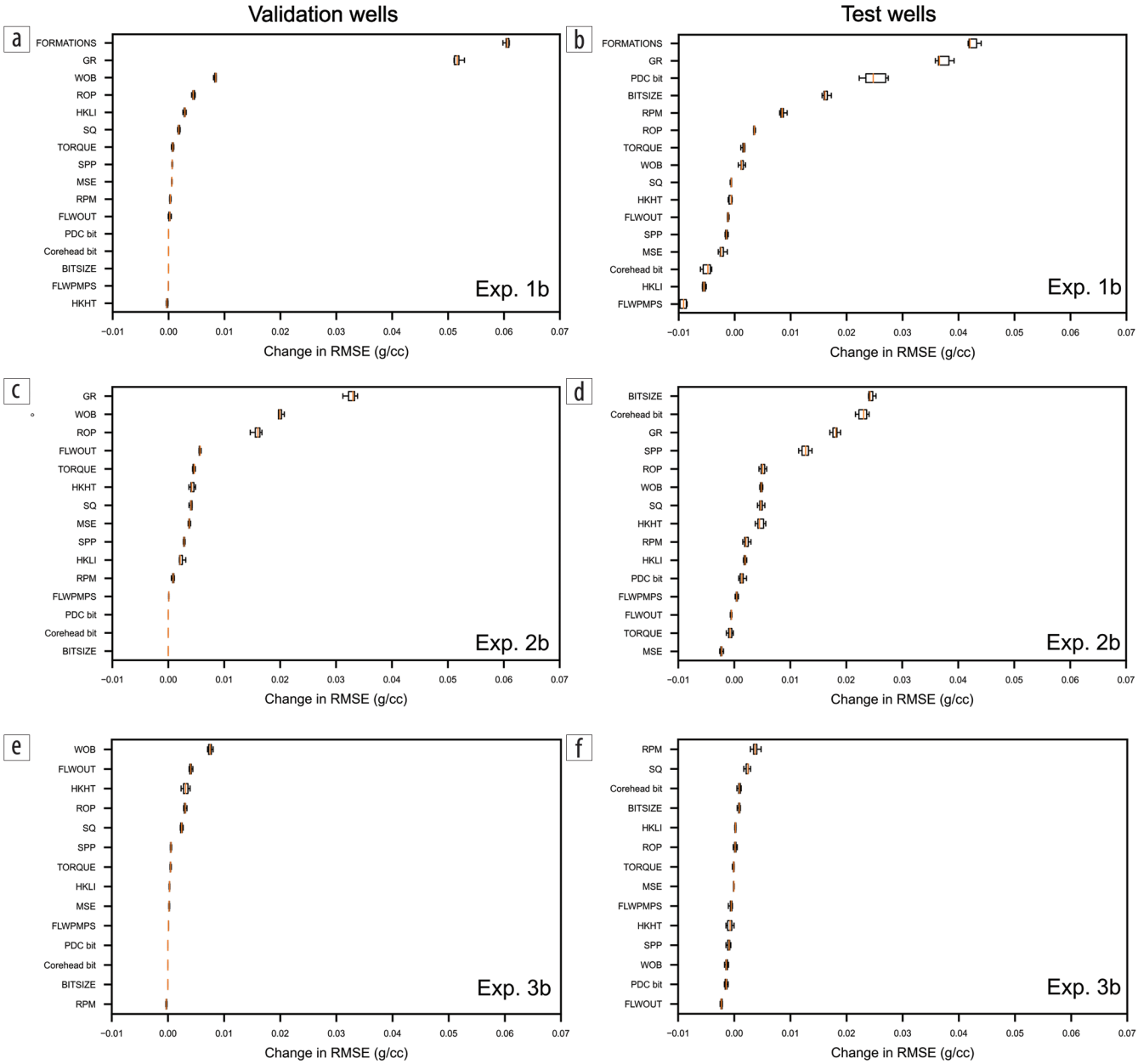

Figure 8. Global permutation feature importances for the density log predictions. The feature importances using the model trained on all input features (experiment 1b) for the (a) validation well and (b) blind well. Similarly, (c) and (d) show the corresponding results for a model trained without formations (experiment 2b), while (e) and (f) are for the model trained using only drilling parameters (experiment 3b).

DOI:10.1190/tle41090617.1

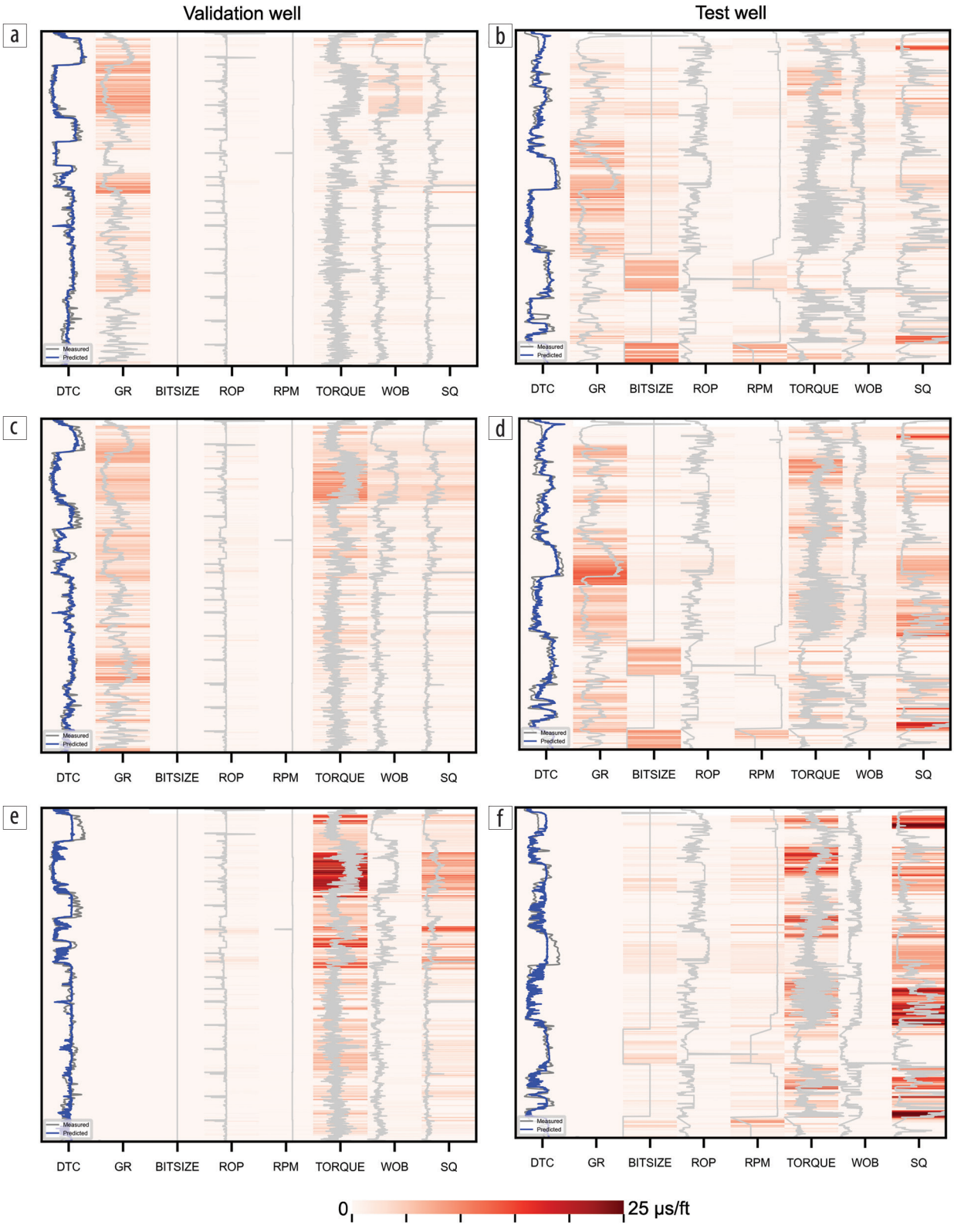

Figure 9. Local permutation feature importance for compressional slowness predictions. The feature importances using the model trained on all input features (experiment 1a) for (a) an example validation and (b) example blind well. Similarly, (c) and (d) show the corresponding results for a model trained without formations (experiment 2a), while (e) and (f) are for a model trained using only drilling parameters (experiment 3a). The background color represents the mean change in RMSE after permuting the feature value at that point.

data. Feature importance analysis showed the inclusion of stratigraphic data to be particularly important, which likely act as a constraint by limiting the possible output range for the model. Although the network cannot predict the fine log details, the results are still useful for many tasks. They should be seen as providing supplementary information rather than as a replacement for acquiring measured logs. Omission of formation data generally results in worse performance. The drilling parameters alone do not produce logs suitable for detailed design or characterization work. Further work may include incorporating additional data sources such as near-bit vibrations. This may help separate the effects of rock cutting and drilling noise on the drilling parameters to improve the model performance for real-time applications. TLE

Data and materials availability

Data associated with this research are confidential and cannot be released.

Corresponding author: robsmith155@gmail.com

References

Akiba T., S. Sano, T. Yanase, T. Ohta, and M. Koyama, 2019, Optuna: A next-generation hyperparameter optimization framework: arXiv: 1907.10902.

Alali, A., R. Smith, P. Nivlet, A. Bakulin, and T. Alkhalifah, 2021, Time-lapse seismic cross-equalization using temporal convolutional networks: First International Meeting for Applied Geoscience & Energy, SEG/AAPG, Expanded Abstracts, 1440–1444, https://doi.org/10.1190/segam2021-3583439.1.

Bai, S., J. Z. Kolter, and V. Koltun, 2018, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling: arXiv: 1803.01271.

Bergstra, J., R. Bardenet, Y. Bengio, and B. Kégl, 2011, Algorithms for hyper-parameter optimization: Presented at the 24th Annual Conference on Neural Information Processing Systems.

Breiman, L., 2001, Random forests: Machine Learning, **45**, 5–32, https://doi.org/10.1023/A:1010933404324.

Fisher, A., C. Rudin, and F. Dominici, 2019, All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously: Journal of Machine Learning Research, **20**, 1–81.

Gan, T., A. Kumar, M. Ehiwario, B. Zhang, C. Sembroski, O. de Jesus, O. Hoffmann, and Y. Metwally, 2019, Artificial intelligent logs for formation evaluation using case studies in Gulf of Mexico and Trinidad and Tobago: Annual Technical Conference and Exhibition, SPE, Extended Abstracts, https://doi.org/10.2118/196064-MS.

Glubokovskikh, S., A. Bakulin, R. Smith, and I. Silvestrov, 2020, How can machine learning synthesize acoustic logs from drilling dynamics and near-bit vibrations? 82nd Conference and Exhibition, EAGE, Extended Abstracts, https://doi.org/10.3997/2214-4609.202010392.

Hareland, G., and R. Nygaard, 2007, Calculating unconfined rock strength from drilling data: Presented at the 1st Canada-U.S. Rock Mechanics Symposium, American Rock Mechanics Association.

He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: Conference on Computer Vision and Pattern Recognition, IEEE, Extended Abstracts, https://doi.org/10.1109/CVPR.2016.90.

Kanfar, R., O. Shaikh, M. Yousefzadeh, and T. Mukerji, 2020, Real-time well log prediction from drilling data using deep learning: arXiv: 2001.10156.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: arXiv: 1412.6980.

Lamik-Thonhauser, B., J. H. Schoen, C. S. Koller, and A. M. Arnaout, 2018, Correlation between drilling parameters and lithology — The hidden geological information of drilling data: Abu Dhabi International Petroleum Exhibition and Conference, Extended Abstracts, https://doi.org/10.2118/192916-MS.

Loshchilov, I., and F. Hutter, 2017, Decoupled weight decay regularization: arXiv: 1711.05101.

Lundberg, S. M., and S.-I. Lee, 2017, A unified approach to interpreting model predictions: Presented at the 30th Annual Conference on Neural Information Processing Systems.

Moazzeni, A., and M. A. Haffar, 2015, Artificial intelligence for lithology identification through real-time drilling data: Journal of Earth Science & Climatic Change, **6**, no. 3, https://doi.org/10.4172/2157-7617.1000265.

Mustafa, A., M. Alfarraj, and G. AlRegib, 2019, Estimation of acoustic impedance from seismic data using temporal convolutional network: 89th Annual International Meeting, SEG, Expanded Abstracts, 2554–2558, https://doi.org/10.1190/segam2019-3216840.1.

Nair, V., and G. E. Hinton, 2010, Rectified linear units improve restricted Boltzmann machines: Presented at the 27th International Conference on Machine Learning.

Nivlet, P., R. Smith, and N. AlBinHassan, 2020, Automated well-to-seismic tie using deep neural networks: 90th Annual International Meeting, SEG, Expanded Abstracts, 2156–2160, https://doi.org/10.1190/segam2020-3422495.1.

Oord, A. V. D., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, 2016, WaveNet: A generative model for raw audio: arXiv: 1609.03499.

Osarogiagbon, A. U., O. Oloruntobi, F. Khan, R. Venkatesan, and S. Butt, 2020, Gamma ray log generation from drilling parameters using deep learning: Journal of Petroleum Science and Engineering, **195**, https://doi.org/10.1016/j.petrol.2020.107906.

Salimans, T., and D. P. Kingma, 2016, Weight normalization: A simple reparameterization to accelerate training of deep neural networks: Presented at the 29th Annual Conference on Neural Information Processing Systems.

Smith, R., P. Nivlet, H. Alfayez, and N. AlBinHassan, 2022, Robust deep learning-based seismic inversion workflow using temporal convolutional networks: Interpretation, **10**, no. 2, SC41–SC55, https://doi.org/10.1190/INT-2021-0142.1.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014, Dropout: A simple way to prevent neural networks from overfitting: The Journal of Machine Learning Research, **15**, 1929–1958.

Teale, R., 1965, The concept of specific energy in rock drilling: International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, **2**, no. 1, 57–73, https://doi.org/10.1016/0148-9062(65)90022-7.