

# Automated well log depth matching: Late fusion multimodal deep learning

Veronica Alejandra Torres Caceres<sup>1\*</sup>, Kenneth Duffaut<sup>1</sup>, Anis Yazidi<sup>2</sup>, Frank Westad<sup>3</sup> and Yngve Bolstad Johansen<sup>4</sup>

<sup>1</sup>Department of Geosciences and Petroleum Norwegian University of Sciences and Technology, S.P. Andersens veg 15a, Trondheim, 7031, Norway, <sup>2</sup>Department of Computer Science Norwegian University of Sciences and Technology, Sem Sælands vei 9, Trondheim, 7034, Norway, <sup>3</sup>Department of Engineering Cybernetics Norwegian University of Sciences and Technology, O. S. Bragstads Plass 2D, Trondheim, 7034, Norway, and <sup>4</sup>AkerBP ASA, Munkegata 26, Trondheim, 7011, Norway

Received November 2021, revision accepted March 2022

## ABSTRACT

Petrophysical interpretation and optimal correlation extraction of different measurements require accurate well log depth matching. We have developed a supervised multimodal machine learning alternative for the task of simultaneously matching raw logging while drilling and electrical wireline logging logs. Seven one-dimensional convolutional neural networks are trained using different log measurements: gamma-ray, resistivity, P- and S-wave sonic, density, neutron and photoelectric factors, and their depth shift estimates are aggregated using different multimodal late fusion strategies. We test the late fusion average, late fusion weighted average, late fusion with linear and nonlinear learners and model-level fusion. Depth matching results using the different fusion strategies applied to two unseen wells are compared using visual inspection and the mean Pearson correlation. All models perform well, increasing the correlation after depth matching. Late fusion weighted average achieves the highest scores for all log types. The late fusion weighted average results are compared to a cross-correlation user-assisted workflow and manual depth matching for validation. In general, the convolutional neural network fused method exhibits a lower performance than the traditional methods. For one of the wells, the cross-correlation shows higher correlation values than the other methods but for the second well the manual depth match performs best. However, the differences in Pearson correlation values are small ranging from 0.01 to 0.1. The manual depth match performs very well for the sonic logs, which tend to require slightly larger depth shifts than the other measurements, thus a common depth shift might not always be suitable. Although our convolutional neural network fused approach is limited to estimating bulk shifts and uses constant fusion weights, its performance is similar to that of more time-consuming methods. Our approach might be substantially improved by including dynamic shifts (stretch/squeeze) and depth-dependent fusion weights via long-short-term memory recurrent neural networks.

**Key words:** Logging, Petrophysics, Machine learning, Depth matching, one-dimensional Convolutional neural networks.

## INTRODUCTION

Well log measurements are continuous records of indirectly measured formation properties, from which it may be possible

\*E-mail: veronicaa.torres.c@gmail.com

to determine subsurface lithology, fluids and reservoir properties, for example, fluid saturation, porosity, permeability and stress regime. These are the main inputs for building static and dynamic reservoir models. Traditionally, mostly electrical wireline logging (EWL) logs were used for this purpose due to their superior accuracy. However, in recent years, with accelerated logging technology development, the use of logging while drilling (LWD) logs in petrophysical analysis has become more and more popular. This is due to a significant increase in their accuracy as well as the massive data acquisition and processing of LWD logs in real-time for making drilling decisions. It is well known that both LWD and EWL logs suffer from environmental effects, instrumental noise, and depth errors. The latter can be present between logs within the same run or logging pass as well as between suites of logs from different runs that measure properties over the same depth interval. These depth discrepancies are an important source of error in identifying correlations between different log measurements that are a key part of several petrophysical and rock physics interpretation workflows. Without proper correction of depth errors, any interpretation of the data will be meaningless, as shown by Zangwill (1982). He also presented the main causes of depth shifts in EWL logs and developed software to address these problems by combining computational algorithms like cross-correlation with close interaction from the analyst to provide additional information when needed.

Although all log measurements are recorded against depth, we can find relatively large depth differences between logging passes from LWD and EWL, which are recorded under two different depth measurement systems as presented by Pedersen *et al.* (2006). These systems are called drillers' depth and loggers' depth, respectively. Both suffer from inaccuracy. Drillers' depth is considered less accurate and is measured using tapes along the length or stands of the drill pipes as they run into the hole and simply add up the lengths (Rider, 1986). Corrections to the drillers' depth were seldom applied in the past, and the length measurements are highly prone to human error. Loggers' depth is more accurate and is, therefore, more widely used as a depth reference. Loggers' depth is measured by performing direct measurement using an odometer, also known as the measure wheel depth (Rider, 1986). Various depth error correction procedures have been run to improve accuracy. At earlier times, there was no correction for wheel wear. In the past decade, all depth wheel systems have been typically calibrated. Before 1980, a magnetic marker system was used, which introduced much larger errors. Varying borehole conditions and environments play a role in magnifying the depth misalignment during the logging process. For

example, some factors affect the drill string length in different drilling rig states including changes in fluid composition, temperature, mud pressure, flow rate, cutting volume, buoyancy, along-hole friction factor, tortuosity induced friction points, etc. (Bolt, 2019). In addition to the different sized pipes used, another factor that introduces uncertainty in depth measurements is the differences in formation hardness. This leads to changes in the rate of penetration and weight on the bit, which are both defined by the movement of the travelling block (Bolt, 2019). The combination of all of these factors can result in variable depth errors of up to about 10 m in the LWD measurements (Theys, 1999; Wilson *et al.*, 2004; Chia *et al.*, 2006; Bolt, 2016, 2019). The corresponding EWL measurements are typically corrected only for cable stretch. However, other causes of error that could be accounted for include temperature, permanent deformation (plastic stretch), buoyancy and radial pressure (Bolt, 2016). Note that the temperature correction can have both positive or negative signs and can have a significant impact on the depth measurements, but this is not well understood, and little information about it is publicly available (Sollie and Rodgers, 1994; Theys, 1999; Bolt, 2016). EWL depth is also very prone to tool sticking and slipping caused by variations in borehole rugosity (e.g., due to mud cake build-up or large changes in the borehole dimensions). This may lead to depth errors of up to 12 m (Theys, 1999) and significantly expanded or compressed data sections (stretch/squeeze). Buoyancy, pressure and twisting effects are considered minor effects and are usually neglected (Sollie and Rodgers, 1994; Theys, 1999). Comprehensive summaries of all these depth corrections for both EWL and LWD depth measurements, as well as a new implementation of depth corrections for each depth measurement system, are presented by Bolt (2016, 2019). He proposed a waypoint correction to improve the currently used depth determination. This included an elastic-stretch correction for the logger's depth and the equivalent driller's way-point depth correction. He shows that after applying his methods, the depth differences between LWD and EWL are considerably reduced.

Despite logging companies' multiple attempts to correct for depth mismatching before delivery of log data, depth matching remains an important step within any petrophysical pre-processing flow. Therefore, since Zangwill (1982) many authors have proposed various solutions. For example, Kerzner (1984) developed an automatic depth matching technique based on correlation coefficients to find all possible depth shifts. This technique also uses mathematical optimization to define a consistent set of shifts. Spalburg (1989) presented a method that performs a deconvolution of logging

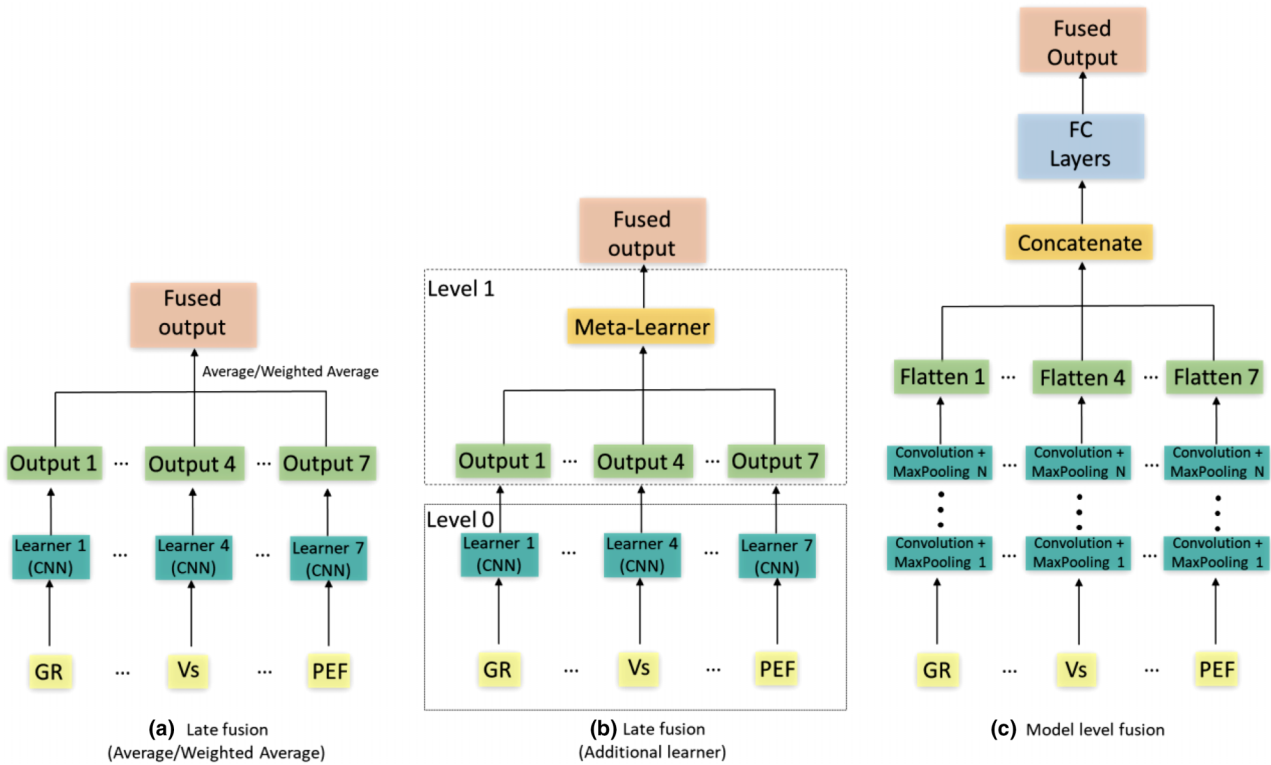


Figure 1 Sketch of multimodal machine learning fusion strategies: (a) late fusion with a simple average and weighted average aggregation for prediction from different modalities, (b) late fusion with an additional learner for aggregation of predictions from different modalities and (c) model-level fusion as a concatenation of high-level features from different modalities.

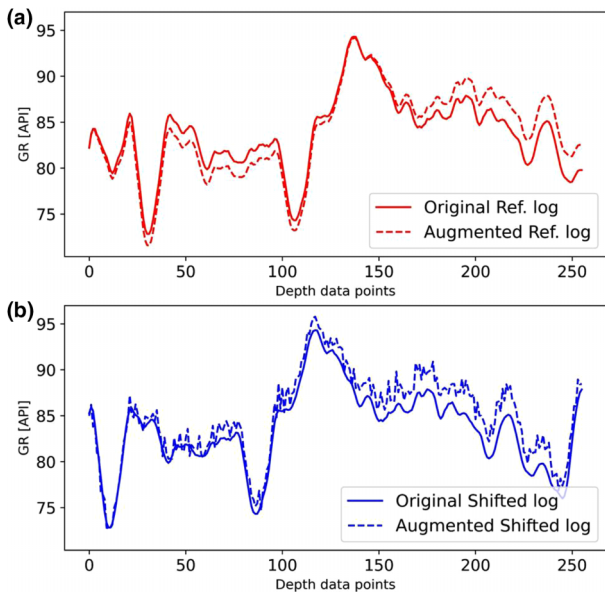
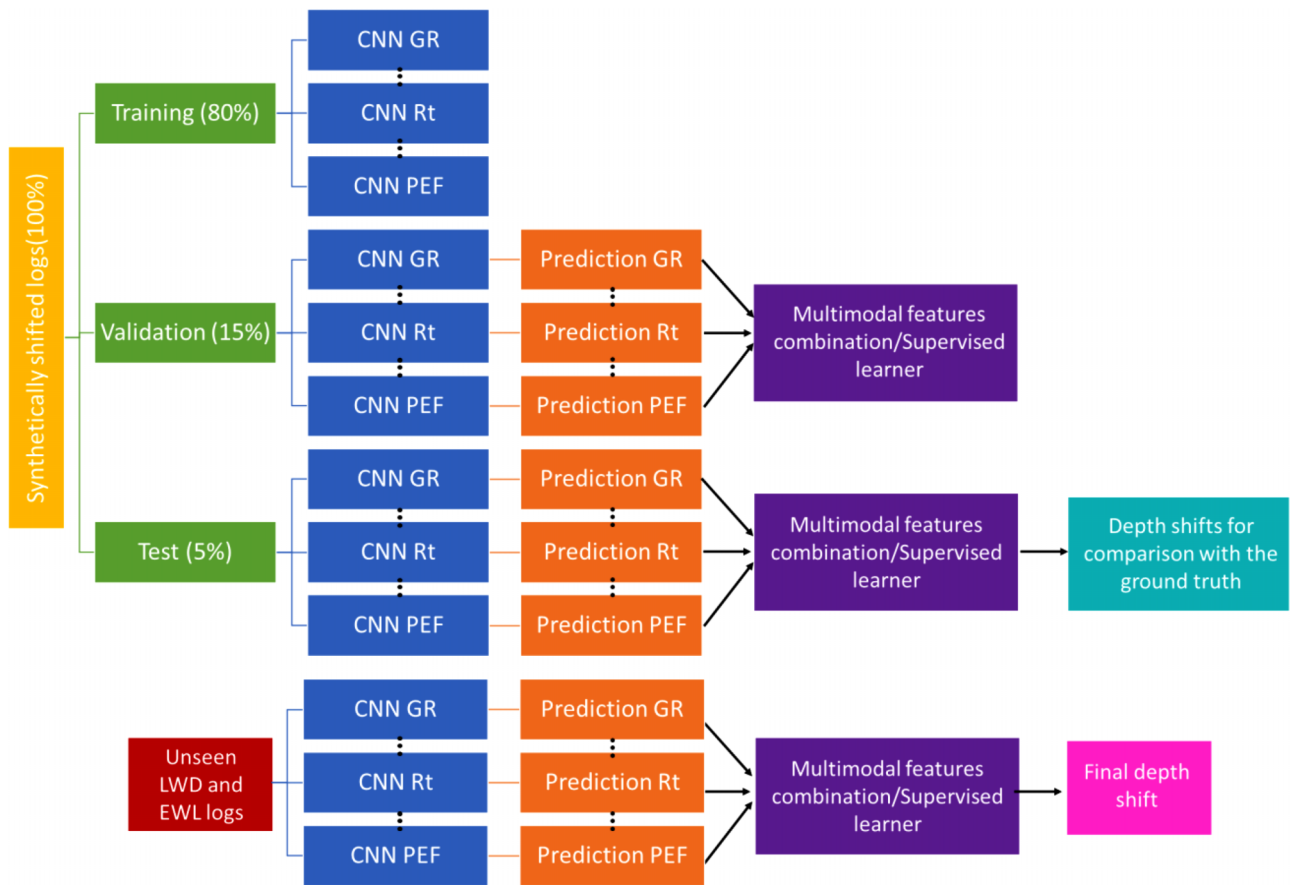


Figure 2 Example of the additional data augmentation implemented in the training and validation sets: (a) example of drift augmentation on the reference gamma-ray log and (b) example of noise and gentle drift augmentation on the shifted gamma-ray log.

data combined with blocking and depth matching. Unfortunately, both of these methods require strong smoothing and filtering of the data to yield good results. Additionally, many techniques that were originally developed for well log depth matching have been implemented for well-to-well log correlation and vice versa. For example, Startzman and Kuo (1986) were pioneers in developing an approach for well-to-well correlation using artificial intelligence based on a set of expert rules. Their approach showed good results in field data, agreeing with the interpretations of several experts. Lineman *et al.* (1987) presented a system for well-to-well correlation based on dynamic depth warping techniques and knowledge-based systems. This overcame some problems, for example, by allowing correlations across missing sections or discontinuous units.

Later advances in computer processor speed allowed (Le Nir *et al.* 1998) to develop an automated well log correlation algorithm based on dynamic programming using multiple logs. This removed the need for pre-processing of the input data as well as increasing the number of wells that could be processed simultaneously. Following the same trend, Luthi and Bryant (1997) applied artificial neural networks (ANNs)



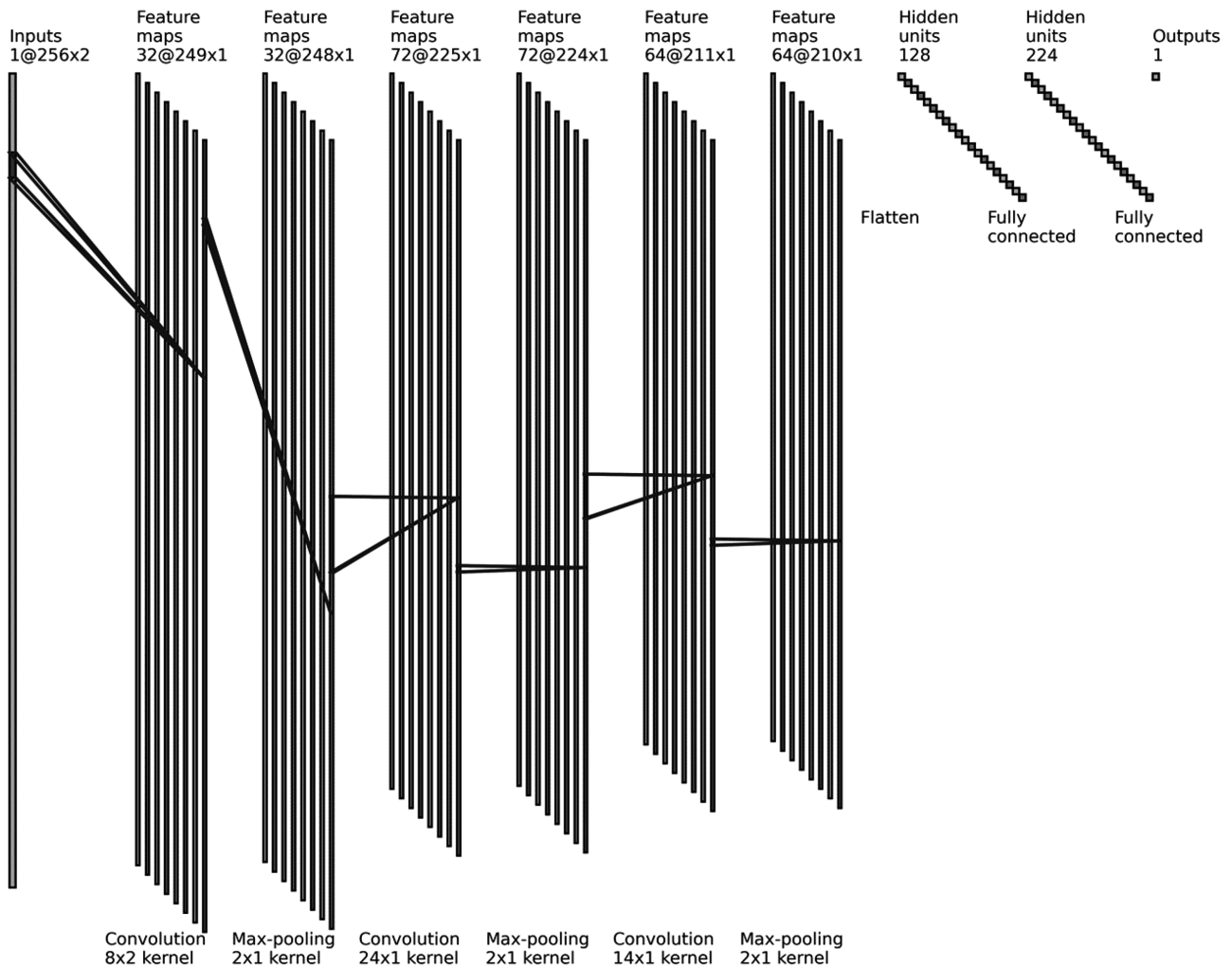
**Figure 3** Sketch of the data-splitting set-up for the CNN models training and fusion workflow and final depth shift inference on the completely unseen data.

for log pattern recognition to perform automatic well-to-well correlation within the same field based on geological datums and relevant markers. This provides the geologist with a tool that allows them to speed up the correlation process interactively. Zimmermann *et al.* (2018) and Le *et al.* (2019) implemented an ANN that mimics manual procedures carried out by expert petrophysicists to depth match well logs. Here two depth series (reference and shifted) are synchronized based on anchor points that are present in both signals and that should match. Le *et al.* (2019) extended the work originally presented by Zimmermann *et al.* (2018) by improving the robustness of the algorithm and incorporating a continuously self-evolving depth-matching framework. This is implemented as a cloud-based depth matching service in which users review the matches output from the algorithm and perform any necessary adjustments. Users' feedback is retrieved by the algorithm and used to retrain and improve the matching algorithm over time. An automatic quality control system evaluates the shifts suggested by the algorithm before they are sent to the user,

using a combination of different metrics. They also modified the anchor point selection by incorporating it into a filtering pipeline. This makes this step generalize better for different log types. Wang *et al.* (2020) deployed a deep neural network as a one-dimensional convolutional neural network (1D CNN) with a multitask set-up for fully automated well-to-well correlation. They used the pattern recognition potential of the CNNs to identify specific geological patterns from a reference gamma-ray log and to find the corresponding patterns in a sequence of logs from different wells in the same field. They proved the power of CNN to recognize geological patterns across different wells within a field and to overcome any problems associated with large depth differences and missing or discontinuous units due to lateral geological variations.

This work is an extension of previous research by Torres Caceres *et al.* (2022a) on the automation of well log depth matching algorithms and procedures. They presented a depth matching workflow that can synchronize several well log measurements to a common reference simultaneously in a





**Figure 4** Sketch of the best CNN structure after the hyperparameter tuning for gamma-ray, resistivity, P- and S-wave sonic and PEF log depth matching. The batch normalization layers after each convolution layer, fully connected layers and the drop-out layers applied after each fully connected layer are not shown. This figure was generated by adapting the code from [https://github.com/gwding/draw\\_convnet](https://github.com/gwding/draw_convnet).

couple of minutes. This workflow speeds up depth matching compared to manual depth matching by a petrophysicist. However, their workflow is limited to bulk depth shifts, and its performance is reduced whenever strongly depth-dependent shifts are present in the data. It is thought to be a good first step towards automation. Unfortunately, their proposed workflow is not fully automated as it still includes some steps in which user intervention is required. For instance, a manual statistical analysis is used to determine the weights for each log measurement type when computing a weighted average depth shift to be applied to all the logs. The selection of the weights is based on the user's selection criteria and knowledge. Motivated by the successful results achieved by 1D CNNs in similar tasks (Brazell *et al.*, 2019;

Wang *et al.*, 2020), we propose replacing Torres Caceres *et al.* (2022a)'s analytical depth matching workflow with a fully data-driven procedure based on deep learning, also extending the work by Torres Caceres *et al.* (2022b).

Torres Caceres *et al.* (2022b) showed that the 1D CNN machine learning algorithm could be a good fully automated alternative to tackle the well-log depth matching challenge. They generated seven independent CNN models trained and applied to each pair of logs, for example, gamma-ray and resistivity pairs. In general, their results indicated that the CNN solution has a similar performance as a traditional cross-correlation method. Therefore, there is potential for improving this methodology by considering more complex scenarios. These improvements will allow us to use this method, which

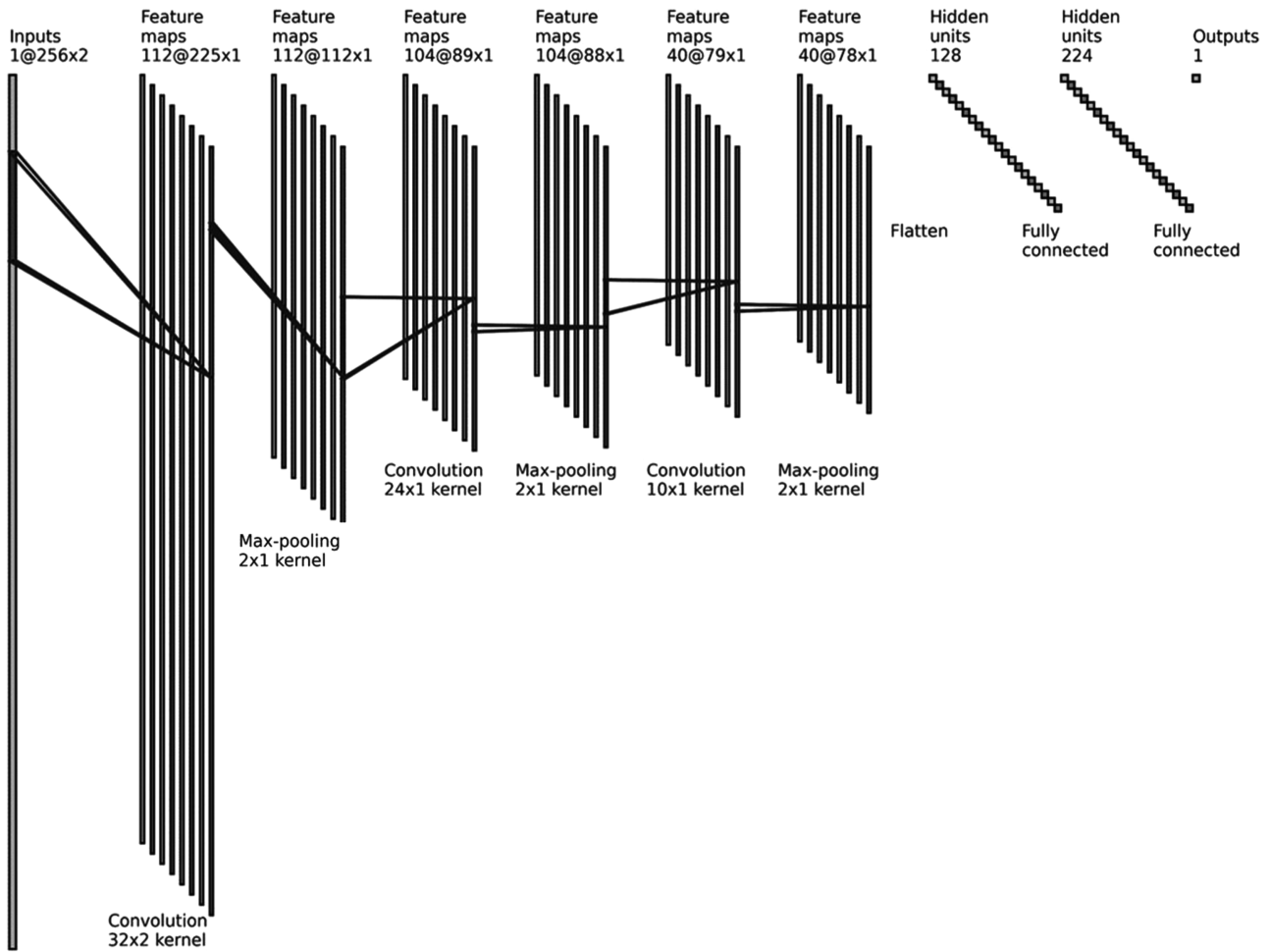


Figure 5 Sketch of the best CNN structure after the hyperparameter tuning for density and neutron log depth matching. The batch normalization layers after each convolution layer, fully connected layers and the drop-out layers applied after each fully connected layer are not shown. This figure was generated by adapting the code from [https://github.com/gwding/draw\\_convnet](https://github.com/gwding/draw_convnet).

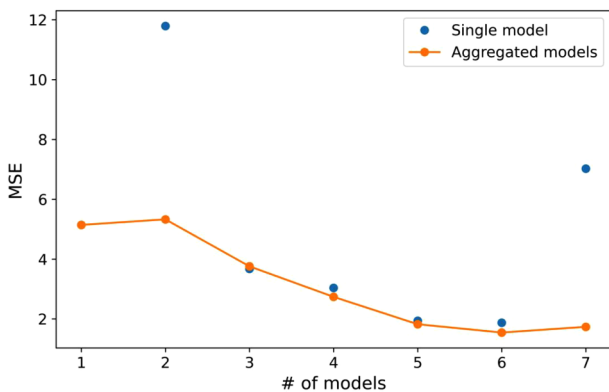


Figure 6 Comparison of model performance in terms of the MSE reduction for the individual models represented with the blue dots and the simple averaging aggregated model represented by the orange line.

reduces user intervention and suppresses the necessity of complex and demanding feature extraction methods for machine learning applications. Based on these results, we extend the work of Torres Caceres *et al.* (2022b) by proving an alternative to improve depth matching between the LWD and EWL suite of logs automatically and in a large-scale settings.

In this paper, we focus on evaluating several ways to combine 1D CNN models, aggregating depth shift predictions from different well log sensors or measurements using multimodal deep learning techniques and replacing the manual weighted average. The prediction task aims to determine an adequate common depth shift for depth matching/synchronizing of LWD and EWL suites of logs. Our implementation is a proposal for a fully automated depth matching workflow. We explore and adopt late fusion and

intermediate fusion model concepts and strategies like those widely used in multimodal machine learning. We train and validate our models using semi-synthetic data from three wells in the Norwegian North Sea. Finally, we test them on two other wells using real logs that were not used in the training and validation process. We assess the depth matching results on the real data qualitatively via visual inspection of matched log profiles and quantitatively using Pearson correlation. This implementation is limited to evaluating bulk shifts, and it assumes that all EWL logs and LWD logs from the same run are in-depth among themselves. This assumption is valid within some well-established ranges during depth control procedures at the well site (Bateman, 1986; Theys, 1999). Our results are also compared to results from other workflows such as the semi-automatic cross-correlation well log depth matching of Torres Caceres *et al.* (2022a) and manual procedures performed by an experienced petrophysicist.

First, we give a brief introduction to the classical method of cross-correlation as well as its implementation within the semi-automatic workflow of Torres Caceres *et al.* (2022a). Second, a more extensive introduction to multimodal machine learning is given focusing on fusion techniques. Third, we present theoretical concepts of CNNs and how we deploy them in our workflow as well as descriptions of datasets, data preparation, data splitting, the training process and our method of depth shift inference. Fourth, we present some quantitative and qualitative results using our depth matching workflow and comparisons to a cross-correlation-based workflow and a traditional manual depth match. Fifth, the Discussion section presents some analysis of the results highlighting the advantages and disadvantages of our proposed fully automatic 1D CNN multimodal workflow before some suggestions for improvements. Additionally, in this section, we present and discuss the comparison between results obtained using independent CNN methods applied for each pair of logs by Torres Caceres *et al.* (2022b) and results obtained using our 1D CNN multimodal workflow. Finally, we state our conclusions and give suggestions for future research.

## METHODS

### Cross-correlation

Previously we developed a semi-automatic depth matching workflow, reported in Torres Caceres *et al.* (2022a), based on cross-correlation; so, we have already formulated the depth matching problem as a signal-processing problem. We focus our workflow on synchronizing log signals between logging

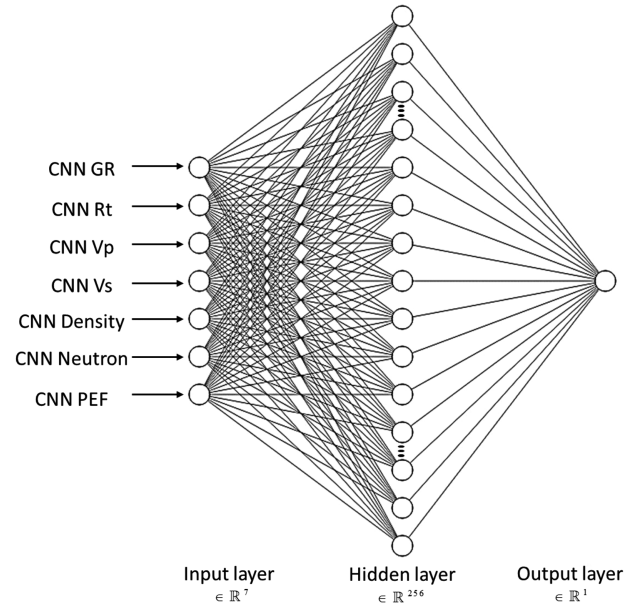


Figure 7 Sketch of the late fusion model with a non-linear meta-learner (ANN). The inputs of the ANN are the outputs of each CNN model generated for each log type, which correspond to the structures shown in Figures 4 and 5.

while drilling (LWD) and electrical wireline logging (EWL) based on the correlated depth concept. The EWL log signal is assumed to be more correctly positioned in depth; hence, it is assigned as the reference log, and the other signal, the LWD log, is shifted appropriately to match that reference (Theys, 1999). For the set-up of depth matching as a signal-processing problem, we assign the signals to the one-dimensional arrays  $\mathbf{x}$  and  $\mathbf{y}$ . Here  $\mathbf{x} = (x_1, x_2, \dots, x_N)$  is a depth series, representing the EWL reference log, and  $\mathbf{y} = (y_1, y_2, \dots, y_M)$  is a depth series representing the depth-shifted LWD log. Both have the same number of data points, for example,  $N = M = 256$  data points. The cross-correlation between the two depth series at a depth lag  $k = 0, 1, \dots, \|\mathbf{x}\| + \|\mathbf{y}\| - 2$  is denoted as  $c$ . The maximum value of  $c$  is an indication of the maximum similarity between the two signals, and we use the  $k$  value (depth lag) at which it occurs to determine the depth shift necessary to align and synchronize the pair of logs. This is expressed mathematically by equation (1):

$$c(k) = (\mathbf{x} * \mathbf{y})(k - N - 1) = \sum_{i=0}^{\|\mathbf{x}\|-1} \mathbf{x}_i \mathbf{y}_{i-k+N-1}^*, \quad (1)$$

where  $\|\mathbf{x}\|$  is the length of  $\mathbf{x}$ ,  $N = \max(\|\mathbf{x}\|, \|\mathbf{y}\|)$  and  $\mathbf{y}^*$  is the complex conjugate of  $\mathbf{y}$  (Oliphant and contributors, 2021).

We implemented a procedure that takes the raw log pairs of each log type, applies some preprocessing (local

**Table 1** Weights and regression coefficients for the late fusion models with weighted average and linear regression as fusion methods

Fused Models	Normalized Weights or Regression Coefficients						
	Gamma-Ray	Resistivity	P-Wave Velocity	S-Wave Velocity	Density	Neutron	Photoelectric Factor
Late fusion weighted average (DE)	0.09	0.00	0.09	0.05	0.42	0.31	0.03
Late fusion Linear Regression (Lasso)	1.22	-0.88	1.11	0.91	3.23	5.37	0.94
Late Fusion Linear Regression (Ridge)	1.31	-0.89	0.55	0.97	2.95	5.91	1.04

**Table 2** Tuning parameter range for the meta-learner (ANN)

Layers	Tuning Parameters	Maximum / Option 1	Minimum / Option 2	Default	Step
Dense layer	Number of units	512	32	128	32
	Activation function	ReLU	Tanh	ReLU	NA
	Number of layers	1	2	NA	NA
Dropout layer	Dropout rate	0.5	0	0.25	0.05
Optimizer	Learning rate	0.01	0.0001	NA	0.001

normalization-standardization, smoothing/filtering, and filling gaps that are no larger than 50 data points), estimates depth shifts for each pair of logs of a given measurement type. For example, gamma-ray – gamma-ray and aggregates the results for each log type via a weighted average to shift each of the LWD logs to a common depth reference established by the EWL logs. The weighted average is parameterized by the user who evaluates the depth shift variability across log types, identifies outliers and assigns plausible weights to each measurement type for each individual logged section. This is done based on the user’s knowledge and experience and data quality requirements to obtain a single depth shift and its standard deviation or uncertainty (Torres Caceres *et al.*, 2022a). Once a single depth shift has been estimated for each

window across all log types, it is applied to the data. Further steps would then have been performed in the originally published workflow. However, for this paper, we stop at this point and evaluate the single depth shift calculated for the data compared to the depth shifts from alternative depth matching procedures using multimodal machine learning and a manual workflow.

### Multimodal machine learning

Multimodal machine learning is a popular branch of research for human activity recognition (HAR), natural language processing, media description, affective computing, audio-visual speech recognition and pedestrian recognition for the development of self-driving cars, to mention some examples. This

**Table 3** Tuning parameter ranges for the model-level fusion’s fully connected layers (dense layers)

Layers	Tuning Parameters	Maximum / Option 1	Minimum /Option 2	Default	Step
Dense layer 1	Number of units	256	32	128	32
	Activation function	ReLU	Tanh	ReLU	NA
Dropout layer 1	Dropout rate	0.5	0	0.25	0.05
Dense layer 2	Number of units	256	32	64	32
	Activation function	ReLU	Tanh	ReLU	NA
Dropout layer 2	Dropout rate	0.5	0	0.25	0.05
Optimizer	Learning rate	0.01	0.0001	NA	0.001

**Table 4** Means and standard deviations of the Pearson correlations for each fusion strategy and log type in well 16/1-9

Fusion Strategies	Gamma-Ray		Resistivity		P-Wave Sonic		S-Wave Sonic		Density		Neutron		Photoelectric Factor	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Late fusion Average	0.83	0.21	0.85	0.14	0.48	0.43*	0.30	0.42*	0.59	0.21	0.71	0.19	0.37	0.31*
Late fusion Weighted average	<b>0.84</b>	0.22	<b>0.86</b>	0.14	0.48	0.42*	0.28	0.44*	<b>0.61</b>	0.23	<b>0.73</b>	0.19	<b>0.38</b>	0.37*
Late Fusion Linear Ridge Regression	<b>0.84</b>	0.25	0.85	0.14	0.47	0.42*	0.29	0.44*	0.60	0.22	<b>0.73</b>	0.19	<b>0.38</b>	0.31*
Late fusion ANNs	0.83	0.24	<b>0.86</b>	0.14	0.48	0.42*	<b>0.31</b>	0.42*	0.58	0.22	0.71	0.20	0.37	0.32*
Model-level fusion	0.83	0.21	0.84	0.14	<b>0.49</b>	0.43*	<b>0.31</b>	0.41*	0.60	0.21	0.71	0.19	<b>0.38</b>	0.31*

We use bold type and \* to highlight the highest mean values for the PC for each log type and to draw attention to relatively high standard deviations, respectively.

**Table 5** Means and standard deviations of the Pearson correlations for each fusion strategy and log type in well 16/1-21 s

Fusion Strategies	Gamma-Ray		Resistivity		P-Wave Sonic		Density		Neutron		Photoelectric Factor	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Late fusion Average	0.88	0.15	0.90	0.13	0.56	0.32*	0.83	0.15	0.73	0.24	0.34	0.37*
Late fusion Weighted average	0.89	0.15	<b>0.93</b>	0.05	<b>0.57</b>	0.32*	<b>0.85</b>	0.14	<b>0.75</b>	0.21	0.35	0.37*
Late Fusion Linear Ridge Regression	0.88	0.15	0.91	0.07	0.56	0.33*	0.83	0.15	0.74	0.22	0.34	0.37*
Late fusion ANNs	0.89	0.15	0.92	0.08	<b>0.57</b>	0.33*	0.84	0.15	<b>0.75</b>	0.21	0.34	0.38*
Model-level fusion	<b>0.90</b>	0.13	0.92	0.06	0.55	0.32*	<b>0.85</b>	0.12	0.74	0.22	<b>0.37</b>	0.36*

We use bold type and \* to highlight the highest mean values for the PC for each log type and to draw attention to relatively high standard deviations, respectively.

emerging field is necessary for artificial intelligence to make good progress in understanding the world. The concept of modality can be defined as the way things happen and the way things are experienced. For example, natural language is manifested either as text or speech (Baltrušaitis *et al.*, 2018). In other words, we could associate modality with several types of information and forms of how this information is transmitted and stored. Modality can also refer to the sensor modality, which is the form of a sensation such as vision, touch, taste, smell or hearing. Multimodal machine learning has as its main goal the generation of models that can process and relate information from multiple modalities/sensors (Baltrušaitis *et al.*, 2018).

Many research studies have revealed the advantages of using multimodal machine learning instead of traditional unimodal approaches. The popularity of multimodal approaches has increased because of increasing access to new large-scale multimodal datasets, faster computers and graphics processing units and the need to solve problems including high-level visual features (e.g., faces, objects and bodies) and dimensional linguistic features such as syntaxis, phonemes, lexemes, etc. (Baltrušaitis *et al.*, 2018). Multimodal machine learning exploits possible complementary features and redundancy between modalities. This means that depending on how the multimodal machine learning is implemented, it may be possible to use valuable information from the



**Table 6** Mean and standard deviation of the Pearson correlation for each depth matching method and log types in well 16/1-9

Depth Matching Methods	Gamma-Ray		Resistivity		P-Wave Sonic		S-wave Sonic		Density		Neutron		Photoelectric Factor	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Original data	0.76	0.28	0.76	0.21	0.42	0.39*	0.26	0.42*	0.48	0.26	0.61	0.25	0.30	0.37*
Late fusion	0.84	0.22	0.86	0.14	0.48	0.42*	0.28	0.44*	0.61	0.23	<b>0.73</b>	0.19	0.38	0.37*
Weighted average														
User assisted cross-correlation (weighted average)	<b>0.86</b>	0.22	<b>0.87</b>	0.14	0.47	0.42*	0.28	0.44*	<b>0.63</b>	0.23	0.71	0.20	<b>0.39</b>	0.32*
Manual depth match	0.85	0.22	0.85	0.12	<b>0.58</b>	0.42*	<b>0.34</b>	0.47*	0.62	0.21	0.69	0.21	0.37	0.32*

We use bold type and \* to highlight the highest mean values for the PC for each log type and to draw attention to relatively high standard deviations, respectively.

**Table 7** Mean and standard deviation of the Pearson correlation for each depth matching method and log types in well 16/1-21 s

Depth Matching Methods	Gamma-Ray		Resistivity		P-wave sonic		Density		Neutron		Photoelectric Factor	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Original data	0.61	0.22	0.63	0.32	0.35	0.29*	0.50	0.30*	0.41	0.30*	0.20	0.32*
Late fusion	0.89	0.15	0.93	0.05	0.57	0.32*	0.85	0.14	0.75	0.21	0.35	0.37*
Weighted average												
User assisted cross-correlation (weighted average)	0.92	0.13	<b>0.94</b>	0.04	0.55	0.31*	0.87	0.11	0.76	0.22	<b>0.39</b>	0.37*
Manual depth match	<b>0.95</b>	0.06	0.93	0.05	<b>0.79</b>	0.17	<b>0.91</b>	0.09	<b>0.83</b>	0.17	0.37	0.38*

We use bold type and \* to highlight the highest mean values for the PC for each log type and to draw attention to relatively high standard deviations, respectively.

internal correlation and interaction between different modalities to further improve the performance of several machine learning algorithms in terms of their robustness and recognition power, especially algorithms from the deep learning category such as convolutional neural networks (CNNs) (Chen and Jin, 2016).

One example of multimodality could be the identification of a user's gender and age in a social network, where their profile's pictures and posts can be used as visual and textual modalities, respectively. Therefore, one way to implement multimodal algorithms is to aggregate signals from different available modalities and to build learning models using aggregated information. In this way, the deep learning algorithm is responsible for determining the relative weights (importance) to be assigned to different modalities for a specific task (Liu

*et al.*, 2018). There is a range of multimodal techniques, including early and late fusion, hybrid fusion, and joint training methods using neural networks. The main concept behind these techniques is that features or intermediate features are merged to make decisions during a specific task. Liu *et al.* (2018) called this an additive approach. Liu *et al.* (2018) focused their multimodal approach on tackling the challenges of weak modalities using a novel deep learning combination that automatically discriminates between strong and weak modalities per observed sample. They also developed a method to automatically select a mixture of modalities that exploits the possible correlations between modalities and identifies whether or not they are complementary.

Similar to ensemble models, multimodal machine learning aims to combine different predictions since it has been

proven that an ensemble classifier or ensemble estimator is generally more accurate than any of the individual classifiers or estimators that build-up the ensemble (Opitz and Maclin, 1999). Gadzicki *et al.* (2020) investigated the potential for multimodal fusion strategies using CNNs and compared them to a traditional unimodal model in the context of HAR. They showed that regardless of the type of fusion it always brings an improvement in the performance. However, for their specific case, the early fusion gives better results than the late fusion. In contrast, Münzner *et al.* (2017) also implemented CNN multimodal fusion for HAR, testing several fusion strategies (early, late and hybrid fusions) combined with specific normalization methods such as standard normalization, batch normalization and pressure mean subtraction (PMS). They found that PMS normalization increases the prediction accuracy of the CNNs and that both late and hybrid fusion outperform early fusion. However, further investigation is needed to determine the optimal fusion strategy for HAR (Münzner *et al.*, 2017).

Multimodal machine learning has several challenges associated with its implementation and problem set-up. The five fundamental challenges are representation, translation, alignment, fusion and co-learning (Baltrušaitis *et al.*, 2018).

### Fusion model strategies

We will focus on different fusion strategies that aim to increase the overall performance of a system. Fusion strategies are motivated by the idea that different data sources can contribute with different kinds of information to alleviate the effects of low data quality and noise while exploiting correlations between modalities.

Early fusion, also called input-level fusion, creates a joint representation of the input features consisting of different modalities. This type of fusion has a relatively low computational cost because it only requires training on a single model to learn the correlations and interactions between low-level features (Liu *et al.*, 2018). The single model can be represented as follows:

$$P = h([\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]), \quad (2)$$

where  $h$  is the single model,  $\mathbf{v}_m$  is the input modality/signal vector, where  $m = 1, 2, \dots, M$ .  $M$  is the total number of modalities available, in our case  $M = 7$ ,  $\mathbf{v}_1$  is the gamma-ray pair of logs,  $\mathbf{v}_2$  is the pair of resistivity logs,  $\mathbf{v}_3$  and  $\mathbf{v}_4$  are the P- and S-wave sonic logs, respectively,  $\mathbf{v}_5$  is density,  $\mathbf{v}_6$  is neutron, and  $\mathbf{v}_7$  is the pair of photoelectric factor (PEF) logs.  $P$  denotes the final depth shift prediction.

Early fusion is simple to implement since only one model needs to be trained. However, it requires highly engineered and preprocessed features to ensure that the different modalities are aligned or similar in semantics. This can be a disadvantage (Liu *et al.*, 2018). Also, the fusion is performed using low-level features, which might be irrelevant to the task; hence, the fusion capacity may decrease (Gadzicki *et al.*, 2020).

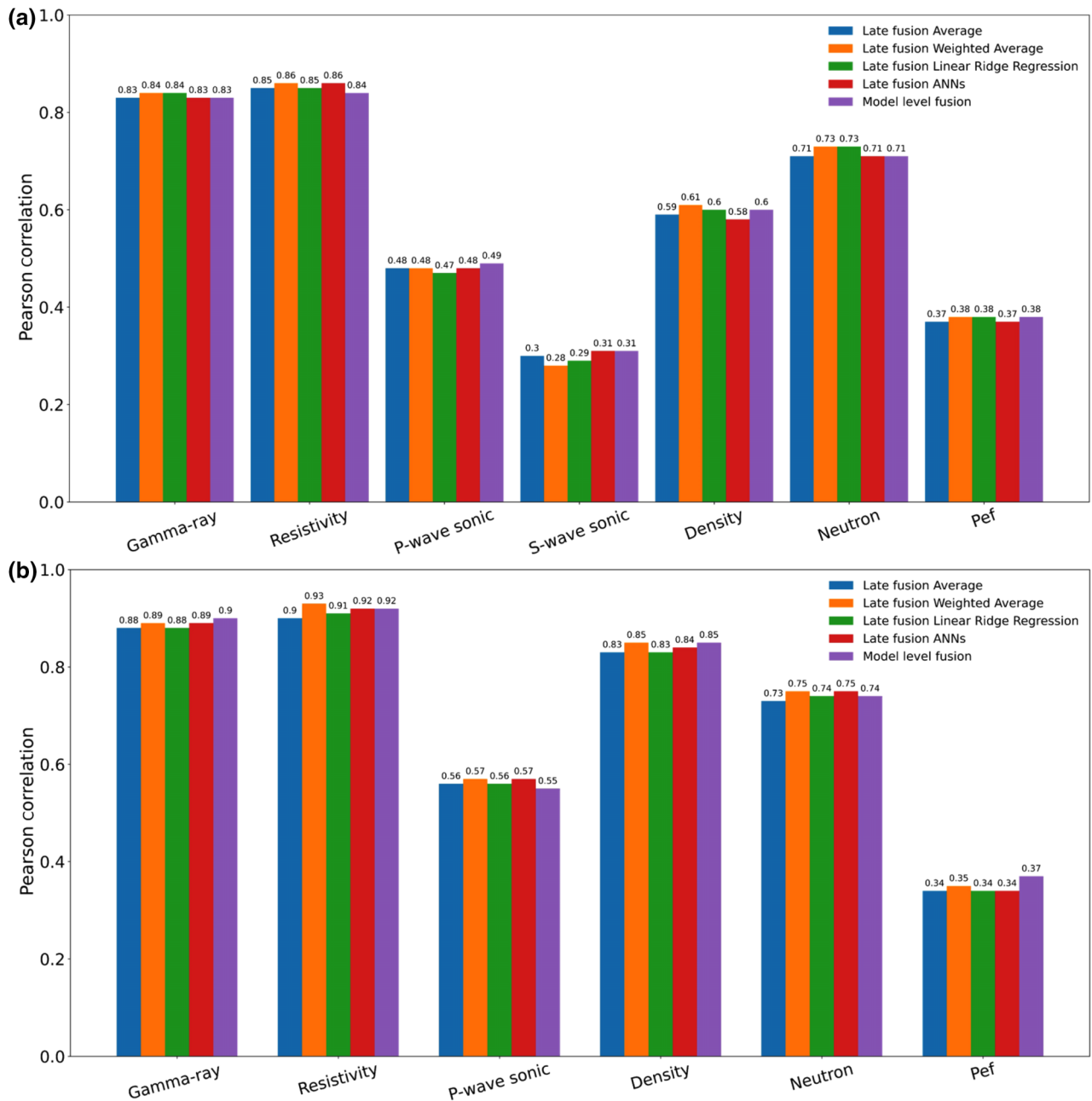
Late fusion, also known as decision-level fusion, uses unimodal decision values or predictions and merges them using fusion mechanisms ( $F$ ) such as averaging, voting or a learned model. When an additional learner model is used to aggregate the predictions this is equivalent to implementing a stacking generalization, which is a machine learning technique developed and explained in detail by Wolpert (1992). A simpler mathematical representation of the final prediction for this type of model is given in equation (3):

$$P = F(h_1(\mathbf{v}_1), h_2(\mathbf{v}_2), \dots, h_m(\mathbf{v}_m)), \quad (3)$$

where  $h_i$  are the individual models trained independently on each log type (modality)  $i$  ( $i = 1, 2, \dots, m$ ). These are also known as level 0 learners or generalizers. The level 0 space consists of a learning set for each log type (modality). These learning sets are divided into subsets of data where the first subset is used to train the learners. A second subset is used to make predictions at level 0. The level 0 predictions, for example,  $h_1(\mathbf{v}_1), h_2(\mathbf{v}_2), \dots, h_m(\mathbf{v}_m)$  are the level 1 learning set. If the fusion mechanism  $F$  is an additional machine learning algorithm this algorithm will be the level 1 learner of the stacking generalization (late fusion model), which needs to be trained on the level 1 learning set. The prediction from this final learner,  $P$ , is the result of testing the whole system on an unseen third subset of data (Wolpert, 1992).

Late fusion provides flexibility as it allows the use of different models on different modalities. This means that we can use one algorithm to train on gamma-ray log pairs and another one to train on resistivity log pairs, for example. This is also the simplest and the most common fusion method (Gadzicki *et al.*, 2020). However, late fusion fails to model signal-level interactions across modalities, thus it has a limited potential for exploiting cross-correlations between different unimodal data types because it works at the inference or decision level instead of working with raw data or features (Liu *et al.*, 2018; Gadzicki *et al.*, 2020). Representations of this type of fusion are shown in Figure 1(a,b).

Model-level fusion is also known as multimodal deep learning when it only uses neural networks as classifiers or estimators. This means that in the model no other algorithm is used. In such cases, it deploys domain-specific neural networks

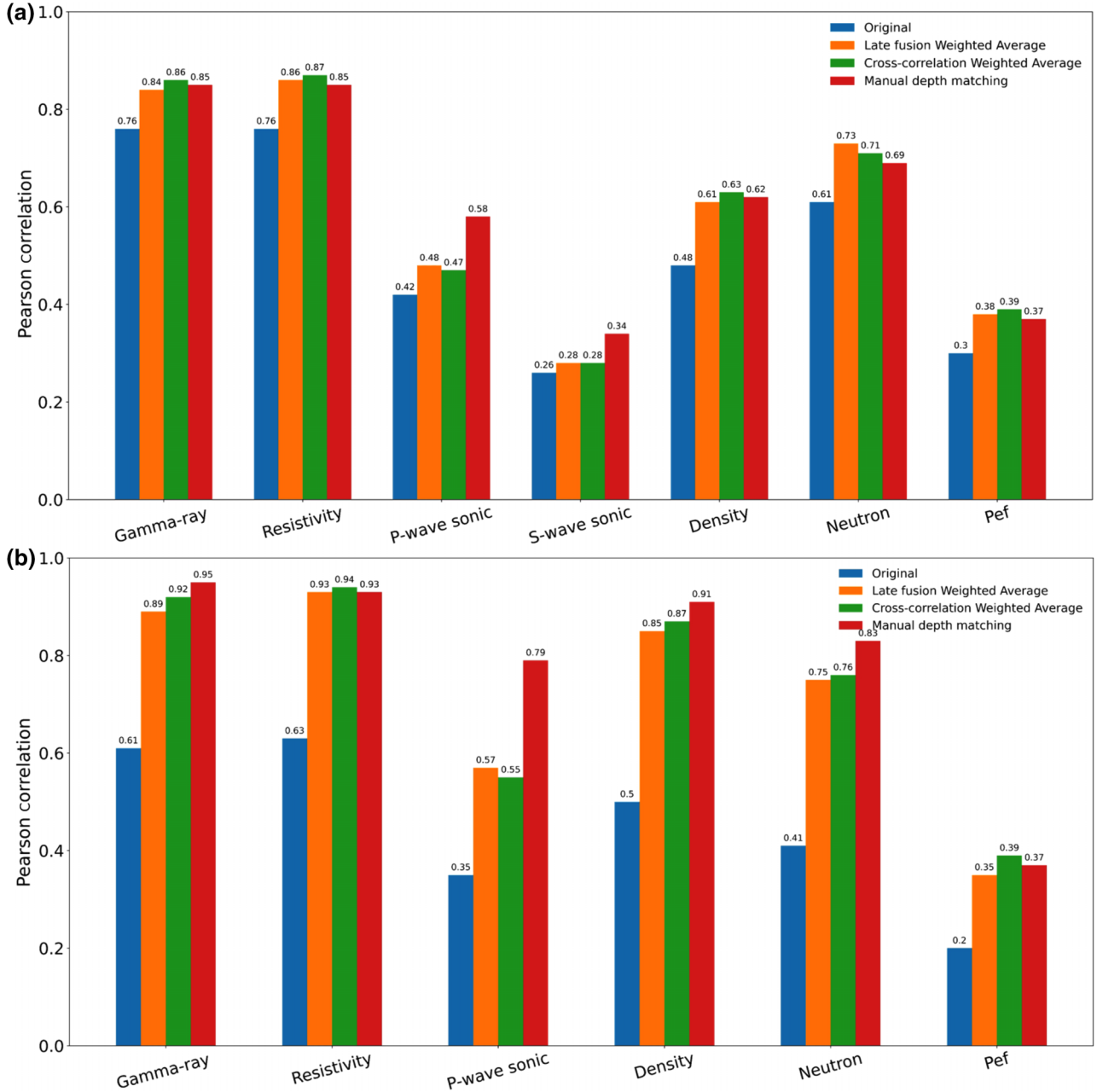


**Figure 8** Overall mean Pearson correlation as a measure of the performance of the different CNN fusion strategies for automatic depth matching. The vertical axis shows the mean correlation values, the horizontal axis shows the labels for each log type, and the colour code represents the CNN fusion strategies used: (a) overall results for well 16/1-9 and (b) overall results for well 16/1-21 S. Note that well 16/1-21 S does not have results for S-wave sonic due to poor data quality.

on the different modalities capturing their main high-level features (more abstract representations) that are then merged or aggregated. The final prediction is made on the aggregated representation usually by another neural network, which captures the interaction between modalities while also learning complex function mapping from the input to the output. Two

widely used aggregation methods are addition (average) and concatenation. Their mathematical expressions are shown in equations (4) and (5), respectively:

$$u = \sum_m^M f_m(\mathbf{v}_m), \quad (4)$$



**Figure 9** Overall mean Pearson correlation as a measure of performance between different depth matching methods before and after depth matching for each log type. The vertical axis shows the mean correlation values, the horizontal axis shows the labels for each log type and the colour code indicates values before depth matching (original data) and after depth, matching using the best performing CNN fusion strategy, cross-correlation, and manual depth matching by a petrophysicist: (a) overall results for well 16/1- 9 and (b) overall results for well 16/1-21 S. Note that well 16/1-21 S does not have results for S-wave sonic due to poor data quality.

$$\mathbf{u} = [f_1(\mathbf{v}_1), f_2(\mathbf{v}_2), \dots, f_m(\mathbf{v}_m)], \quad (5)$$

$f_m: \mathbb{R}^{dm} \rightarrow \mathbb{R}^d$  ( $m = 1, 2, \dots, M$ ), another network  $g$  computes the final output prediction  $P$  as described by equation (6):

where  $\mathbf{u}$  is an aggregated representation that fulfils that  $\mathbf{u} \in \mathbb{R}^d$  (for equation (4)) or  $\mathbf{u} \in \mathbb{R}^{\Sigma^{dm}}$  (for equation (5)). Given that the domain-specific neural networks are denoted by  $f$  and

$$P = g(\mathbf{u}), \quad (6)$$

Table 8 Training relative execution times in minutes for the different CNN fusion strategies

CNN Fusion Strategies	Execution Times (minute)		
	Individual CNN Models' Training	Additional Learner/ Optimization	Total
Late fusion average	153.7	NA	153.7
Late fusion weighted average	153.7	598.2	752.0
Late fusion linear Ridge Regression	153.7	0.1	153.8
Late fusion ANNs	153.7	1.2	155.0
Model-level fusion	NA	NA	120.5

where  $g: \mathbb{R}^d \rightarrow \mathbb{R}^k$  and  $k = (1, 2, \dots, n)$  for multimodal classification or  $k = 1$  for single prediction (Liu *et al.*, 2018). An example of this fusion strategy implementation is illustrated in (Figure 1c).

### Deep learning implementation

The models deployed in this paper are implemented in *TensorFlow Keras* (Chollet, 2015). We mainly focus on one-dimensional convolutional neural network (1D CNN) architecture due to their high success in tackling similar tasks (Brazell *et al.*, 2019; Imamverdiyev and Sukhostat, 2019, and Wang *et al.*, 2020; Deng *et al.*, 2021), as well as in several multi-sensor problems that use time-series data (e.g., Abdoli *et al.* (2019), Gadzicki *et al.* (2020), and Münzner *et al.* (2017)). Additionally, CNN's versatility and avoidance of complex feature engineering and extensive preprocessing of the input data is an important advantage that contributed to this choice.

A CNN is a multilayer architecture that consists of several sequences of convolutional layers and pooling layers. These layers' sequences are also known as the feature extractor term within the CNN (convolutional block). The convolutional block is followed by a stack of fully connected layers (FC) with an output layer that can be chosen depending on whether the network's task is to be a classifier or a regression estimator, similar to artificial neural networks (ANNs). During the training process, the mapping of the inputs (raw well log data) to the outputs (depth shift values for example,  $\pm$  number of regularly sampled data points) is learned through an optimization process involving an update of the model parameters of each layer by choosing parameters that minimize a loss function. The convolutional layers have units that are represented as feature maps  $\mathbf{a}_j^{(l+1)}$  for layer  $l+1$ . These feature maps are connected through local patches with weights called kernels or filters  $\mathbf{W}_{jk}^l$  to the feature maps of the previous layer  $l$ . The

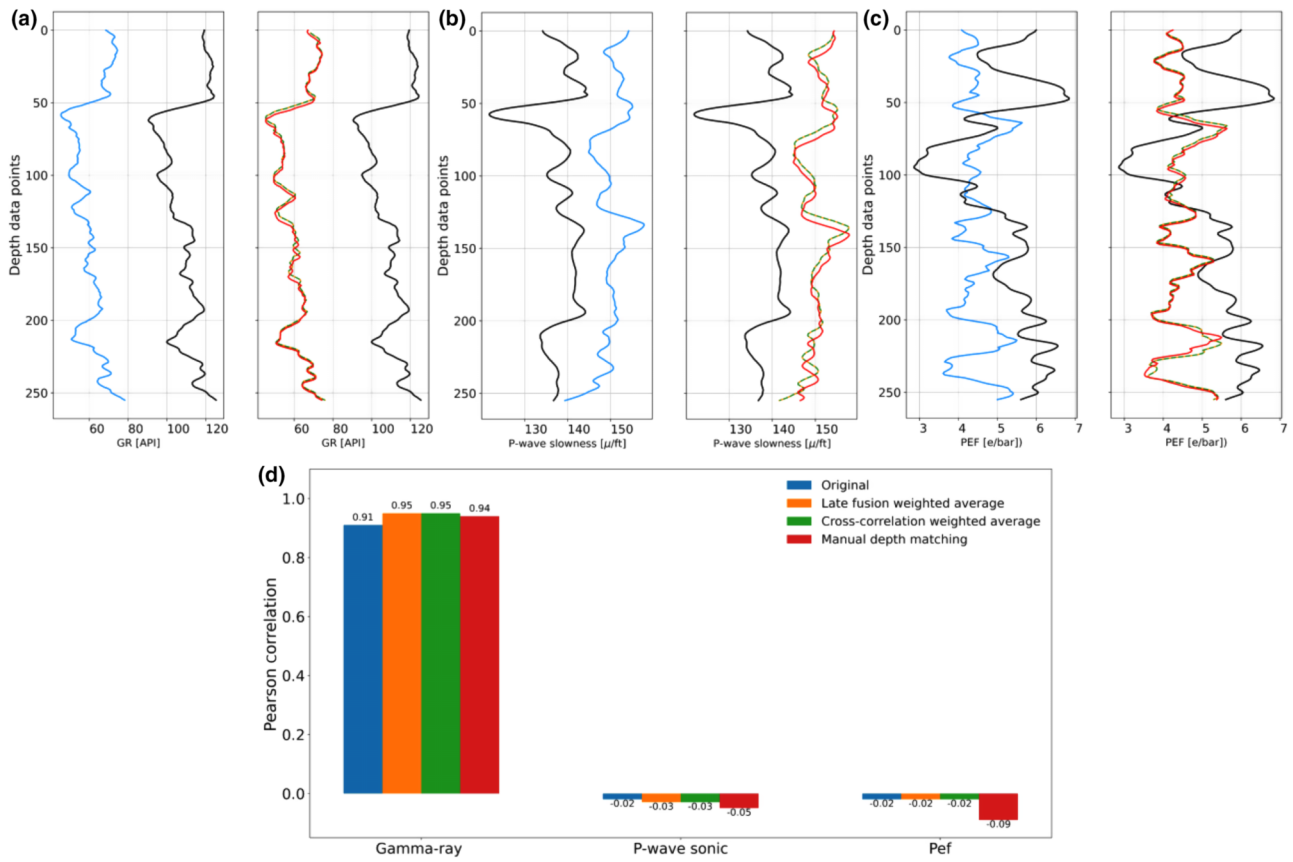
output of this locally weighted sum is passed through a non-linear function  $\sigma$  such as the rectified linear unit (ReLU). Note that all units in a single feature map share the same weights. In other words, each feature map has a unique kernel to compute discrete convolutional filtering (LeCun *et al.*, 2015). The representation of a convolutional layer is given in equation (7):

$$\mathbf{a}_j^{(l+1)} = \sigma \left( b_j^l + \sum_{k=1}^{n_f^l} \mathbf{W}_{jk}^l * \mathbf{a}_k^l \right), \quad (7)$$

where  $\mathbf{a}_j^{(l+1)}$  is the feature map or unit  $j$  in layer  $l+1$ ,  $\sigma = \max(x, 0)$ ,  $\mathbf{x}$  is the output of the convolution operation and  $n_f^l$  is the number of convolutional filters in layer  $l$ .  $\mathbf{W}_{jk}^l$  and  $b_j^l$  are the weights of the convolutional filter and the bias term. The weights are defined as a two-dimensional matrix  $\mathbb{R}^{d \times f}$ , where  $d$  is the number of channels, equal to 2 in our case (EWL reference and LWD shifted logs), and  $f$  represents the filter length. The convolution operator is denoted as  $*$  (Münzner *et al.*, 2017). The movement of the convolutional filters is defined by the stride, which can be equal to 1 or 2. Our input layer can be defined as  $\mathbf{a}_1^l$  of size  $\mathbb{R}^{n_c \times n_{dp}}$ , where  $n_c$  is the number of channels = 2 and  $n_{dp}$  is the number of data points in each channel = 256. We divide the logs into segments of 256 data points each. Each segment represents a single sample in the machine learning algorithm.

We now set up the depth matching problem as a pattern recognition task using the 1D CNN algorithm, where we define an EWL log as a reference channel and its equivalent LWD log as a shifted channel. First, we aim to extract distinctive patterns in the reference log. Second, we seek to identify the same patterns in the shifted log. Finally, we match the patterns to synchronize the signals automatically. We define the same search space for all CNN log-type models. This hyperparameter space is based on a 1D CNN model used for environmental sound identification with limited data presented by Abdoli *et al.* (2019). The hyper-parameterization is performed using





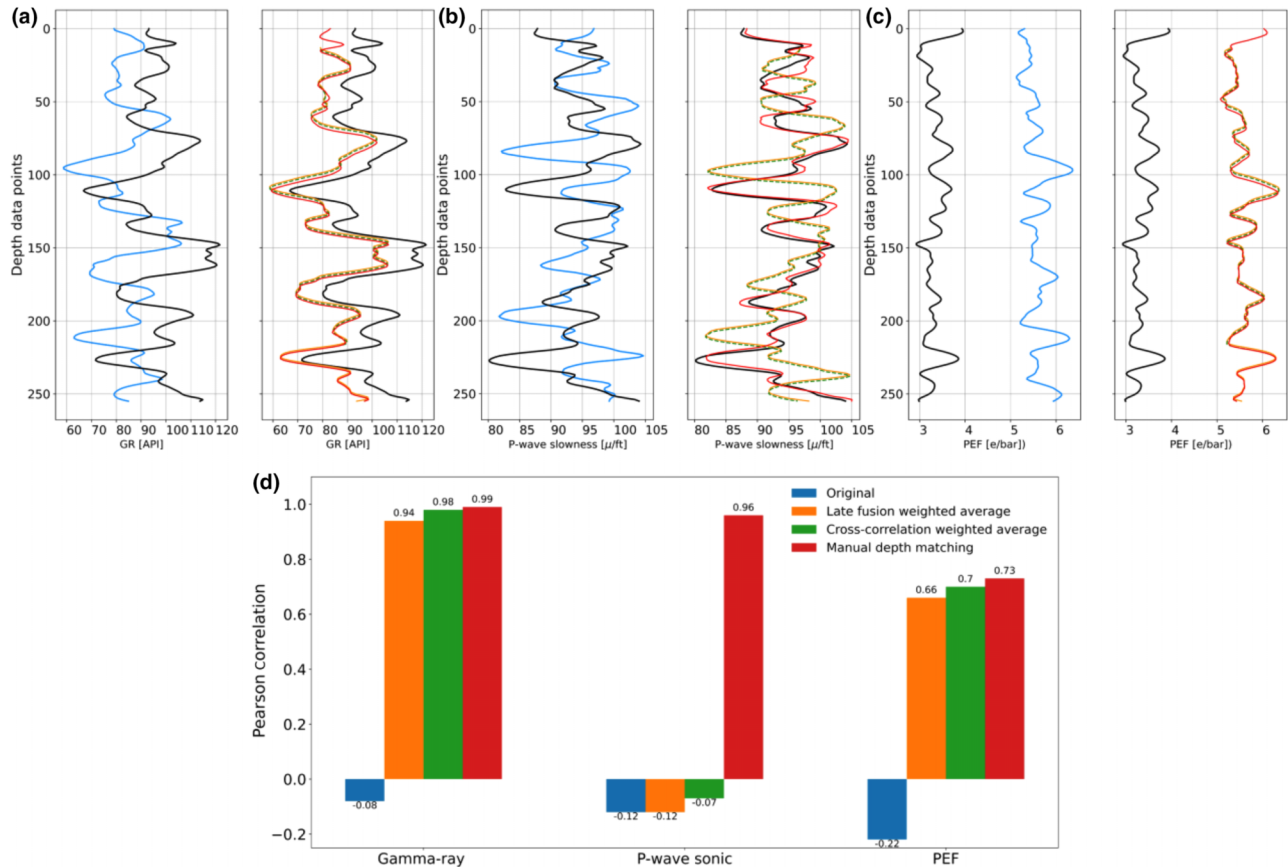
**Figure 10** Log profiles of window/sample number 17 for well 16/1-9: (a) Gamma-ray logs before (left) and after depth matching (right), (b) P-wave sonic logs before (left) and after depth matching (right), (c) PEF logs before (left) and after depth matching (right), and (d) Pearson correlation of the same window/sample number 17 before (original) and after depth matching using different methods. In panels (a) to (c), solid blue lines represent the unmatched LWD curves, the black solid lines are the corresponding reference EWL logs, the orange solid line is the depth-matched LWD using the CNN late fusion approach, the green dashed line is the depth-matched LWD using the cross-correlation workflow, and the red solid line represents the depth-matched LWD using manual adjustments, which correspond to the colour code of the Pearson correlation bar plots in (d).

hyperband as a search algorithm, which has shown better and faster results than other methods (Li *et al.*, 2017). The number of convolutional layers and FC layers remains unchanged at 3 and 2, respectively. The learning rate varies from 0.0001 to 0.1. The number of filters ranges from 8 to 128, and the filter length  $f$  varies depending on the convolutional layer. For example,  $f$  in the first convolutional layer can be of length 8–64, in the second convolutional layer its length ranges from 8 to 32, and in the third its length is between 2 and 16. This gradual reduction in the filter length allows the extraction of global features (long-wavelength content) by the first convolutional layer and more local features (short-wavelength content) by the deeper layers within the network. The stride can be one or two, and the non-linear function  $\sigma$  for all the convolutional layers is ReLU. The pooling layers are specifically max pool-

ing, which extracts patches from the output feature maps and takes the maximum value ignoring the rest (Yamashita *et al.*, 2018). For the two FC layers, the number of neurons varies from 32 to 256, the dropout ranges from 0 to 0.5, and the non-linear functions can be either ReLU or hyperbolic tangent (tanh). Between each convolutional layer and FC layer, we introduce a batch normalization layer. This layer reduces overfitting and speeds up the training process, allowing for higher learning rates (Ioffe and Szegedy, 2015). We use the identity function and one single unit for the output layer.

### Semi-synthetic dataset

Here we use semi-synthetic data to facilitate the evaluation of the machine learning models during the training and

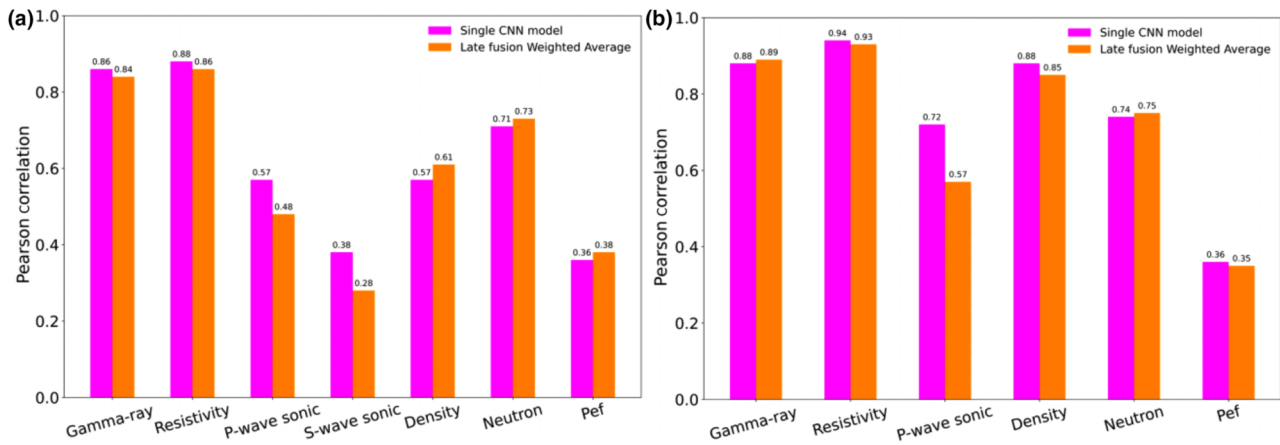


**Figure 11** Log profiles of window/sample number 28 for well 16/1-21 S: (a) Gamma-ray logs before (left) and after depth matching (right), (b) P-wave sonic logs before (left) and after depth matching (right), (c) PEF logs before (left) and after depth matching (right), and (d) Pearson correlation of the same window/sample number 28 before (original) and after depth matching using different methods. In panels (a) to (c), solid blue lines represent the unmatched LWD curves, the black solid lines are the corresponding reference EWL logs, the orange solid line is the depth-matched LWD using the CNN late fusion approach, the green dashed line is the depth-matched LWD using the cross-correlation workflow, and the red solid line represents the depth-matched LWD using manual adjustments, which correspond to the colour code of the Pearson correlation bar plots in (d).

validation process. However, unseen logs are used to assess the performance of the different aggregation methods. We also compare the results from the best-aggregated model to the cross-correlation and manual depth matching results.

We use three wells from the Ivar Aasen field in the Norwegian North Sea that have full suites of logs (gamma-ray, resistivity, P- and S-wave sonics, density, neutron and PEF). These wells are limited to final petrophysical composite versions formatted as LAS (Log ASCII Standard) files. The available log curves are, therefore, already depth matched, spliced and edited according to the Norwegian Petroleum Directorate's (NPD) regulatory requirements for log data delivery. We therefore only have a single log curve of each log type for example, gamma-ray, deep resistivity, P- and S-wave sonic, density, neutron and PEF. These log curves are the result of merging the

best data from the LWD and EWL runs. The merging of EWL and LWD logs aims to provide log measurements that cover the largest possible depth interval within a borehole (NPD, 2019). However, our goal is to use machine learning models to automatically perform log alignment/synchronization between LWD and EWL suites of logs within the same depth interval. To do this, we synthetically shift the log measurements within an acceptable depth shift range based on depth controls. These depth controls are performed by the logging companies at the wellsite (Bateman, 1986). We use a depth shift range that goes from  $-20$  to  $20$  data points, including zero shift. A shift of  $20$  data points is approximately equivalent to  $\pm 3$  m ( $\approx 10$  ft) as the sampling interval is  $0.15$  m ( $\approx 0.5$  ft). We chose this depth shift range because the shifts we have observed in the data do not exceed  $4.5$  m ( $\approx 30$  data points). This



**Figure 12** Comparison of the overall mean Pearson correlation to evaluate the performance of the CNN fusion model (late fusion weighted average) and the individual CNN models trained separately for each log measurement (Torres Caceres *et al.*, 2022b). The vertical axis shows the mean values of the Pearson correlation, the horizontal axis shows the labels for each log type and the colour code indicates the CNN model used: (a) overall results for well 16/1-9 and (b) overall results for well 16/1-21 S.

limitation of the depth shift range works as a soft constraint, allowing us to test our approach faster without compromising the results.

Before proceeding with the semi-synthetic data generation and preparation for training on a well, we impose the following requirements:

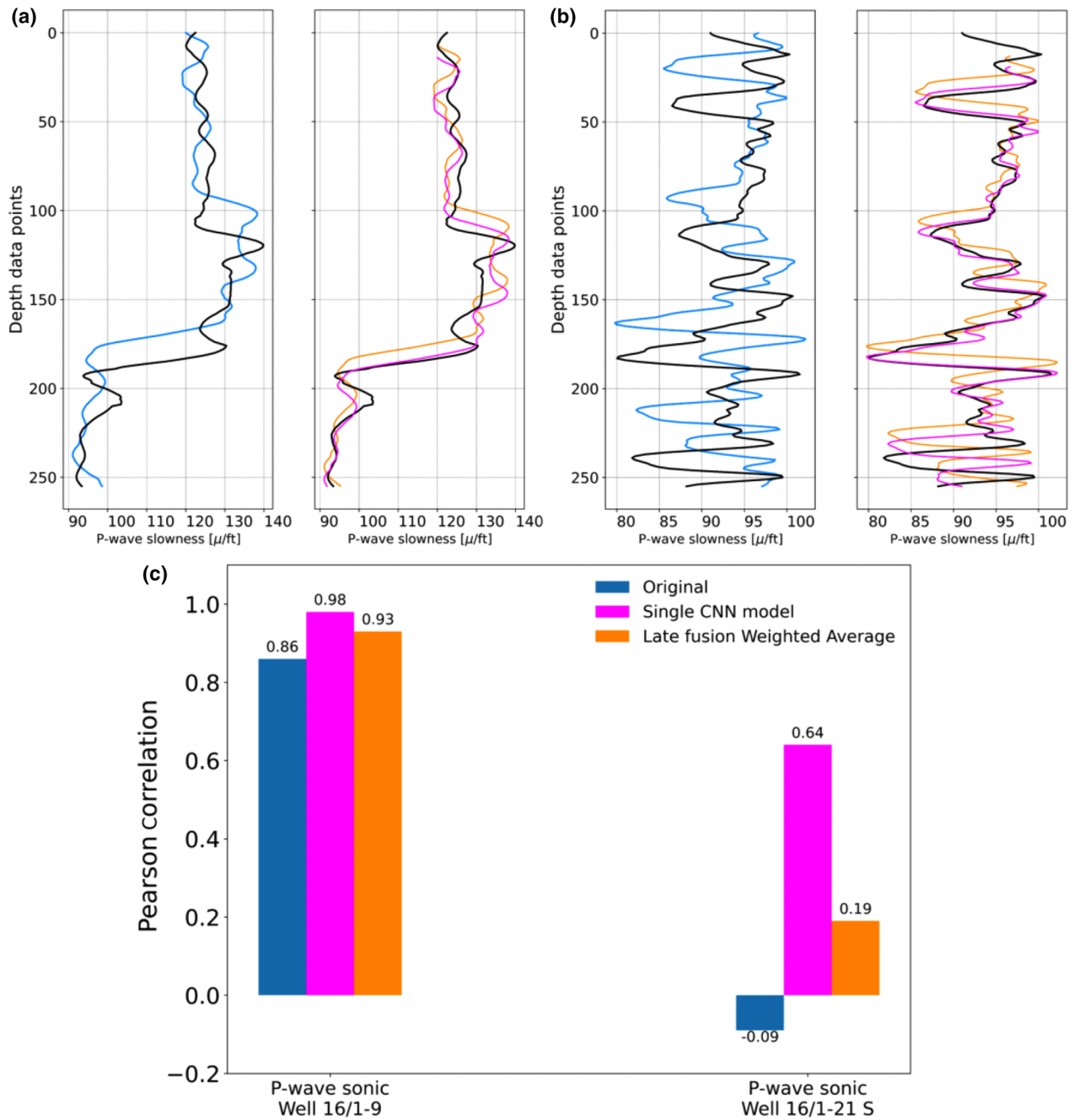
- All log types are available.
- The data selection depends on the depth range in which all the log measurements are acquired. In other words, if we have all the log measurements acquired only around the reservoir zone, even though there are other logs like gamma-ray, resistivity or sonic logs acquired in a larger depth range, we limit the data to the reservoir zone. We do this so that the number of samples for all the measurement types is the same.
- The selected data must honour the assumption made in Torres Caceres *et al.* (2022a) that no depth shifts exist among logs acquired within the same logging run, and if they exist they are limited to a range of 0.15 m ( $\approx 0.5$  ft) to 0.60 m ( $\approx 2$  ft) for vertical wells and of 0.6 m ( $\approx 2$  ft) to 1.22 m ( $\approx 4$  ft) for deviated wells (Bateman, 1986).

#### Data preparation for training

First, we apply a gentle smoothing filter to the logs. Second, we generate multiple copies of the logs and shift them either up or down (positive or negative depth shift value) by a specific integer number of data points within our chosen depth shift range ( $-20$  to  $20$  data points). The total number of copies is equivalent to the number of simulated depth shifts, which is

41 per log measurement. Having all possible depth shifts along the entire depth range allows us to have a balance among the different labels (signed integer number of shifts) in the dataset, ensuring that all the shifts are possible along the borehole. The original log represents the reference EWL log, and the shifted copy corresponds to the shifted LWD log in our problem setup. Third, we divide the logs into segments or windows of 256 data points ( $\approx 39$  m) with an overlap of 50%. This division acts as an augmentation of the data, doubling the dataset size. Fourth, some preprocessing such as local standardization and normalization is performed over each window. Finally, we generate a depth shift vector containing the signed values of the simulated depth shifts and store them as ground truth labels to feed the supervised learning algorithm.

In addition, other augmentation techniques are implemented via the Python library *tsaug* (time series augmentation) that offers several time-series augmentation methods that can be combined and implemented within a simple pipeline (Arundo Analytics, Inc., 2020). We use *tsaug* and design a small pipeline function that randomly applies two augmentation techniques that do not alter or influence the depth shifts, avoiding inconsistencies within the generated labels. We, therefore, only use augmenters that inject variability in the log's measurements. For example, we add random noise to log measurement using the *AddNoise* option with a scale from 2% to 4% and with a probability of occurrence of 50%. This augmentation can be applied to both the reference (EWL) and the shifted log (LWD/Mesure While Drilling) to either one of them or neither of them. Whether the augmentation is applied or not is controlled by the probability of occurrence. Another



**Figure 13** Log profiles of window/sample number 21 for well 16/1-9 and window/sample number 27 for well 16/1-21 S: (a) P-wave sonic logs before (left) and after depth matching (right) well 16/1-9, (b) P-wave sonic logs before (left) and after depth matching (right) well 16/1-21 S, (c) Pearson correlation of the same window/sample number 21 and 27 before (original) and after depth matching using a CNN model trained on P-wave sonic logs and the CNN fusion model (late fusion weighted average). In panels (a) and (b), solid blue lines represent the unmatched LWD curves, the black solid lines are the corresponding reference EWL logs, the magenta solid line is the depth-matched LWD using a single CNN model trained on P-wave sonic logs, and the orange solid line is the depth-matched LWD using the CNN late fusion approach, which corresponds to the colour code of the Pearson correlation bar plot in (c)

example of an augmentation method is *drift*, which adds noise to the values of the original signal randomly and smoothly. The effect of the drift is controlled by the maximum drift allowed and the number of drift nodes. The maximum drift determines how much the signal values can deviate from their original values, and the number of drift nodes is the maximum number of data points in the signal that can be drifted. We introduce drift into our augmentation pipeline with a maximum drift value ranging between 10% and 25%, three drift nodes and a probability of occurrence of 50%. Similar to the additive noise augmentation, drift can be applied to both signals either to one of them or to neither of them. Figure 2 shows an example of the data augmentation process. This additional augmentation introduces more variation and realism into the training and validation datasets. It also increases their size.

### Real dataset

We use misaligned suites of logs to evaluate the depth matching results for the different fusion models. We also assess the best fusion model compared to common current approaches (cross-correlation and manual depth matching by a petrophysicist). This dataset (unseen real data) is kept out from training, validation and preliminary testing. It is used only for depth shift inference. The unseen real dataset consists of two wells from the Ivar Aasen field. They have separate suites of LWD and EWL logs contained in DLIS (Digital Log Interchangeable Standard) files. The LWD and EWL logs are compared by estimating the necessary shift needed to obtain the best possible alignment between them. For these two wells, we also have a depth shift solution provided by an experienced petrophysicist, that is, LWD logs shifted to match the EWL reference log following the petrophysical best practices, standards and regulations set by the NPD).

### Training models

We train different models for each fusion strategy tested. For late fusion, we train seven CNN models corresponding to each log measurement type. When we use another learner as a fusion mechanism an additional algorithm is trained. In contrast, for the model-level fusion, we train a single large multi-input CNN model. The different fusion strategies are trained, validated and tested using the same semi-synthetic data. In both cases, final depth shift inference is carried out using the unseen real data.

We split the whole semi-synthetic dataset into a training set (80%), a validation set (15%) and a test set (5%). The

validation set is used to monitor the model selection process during the training of the individual CNN models corresponding to each log type. The validation set is also used to make predictions using level 0 models and then to train the level 1 model for final predictions (late fusion with an additional learner as a fusion mechanism). We also keep 5% of the data as the test set to evaluate the final model against ground truth labels (signed depth shifts) before using them for depth shift predictions on the real data, which lack ground truth labels. The size of the dataset is the same across log types and consists of 8405 samples. Figure 3 illustrates the general data-splitting set-up.

The training and tuning of the CNNs and ANNs are carried out using the *Keras tuner* library (O'Malley *et al.*, 2019). We use the hyperband algorithm to search for the best combinations of hyperparameters and model architectures. The optimizer used is the Adam optimizer with the default parameters from *Tensorflow Keras backed*, which are a constant learning rate of 0.001, batch size of 128 samples and the number of epochs of 100. The Adam optimizer is a stochastic gradient descent method based on adaptive estimates of first- and second-order moments (Kingma and Ba, 2015). The best model is selected based on its performance measured by the mean squared error on the validation set (MSE) after each trial. The best models for each log type are then used to test the different fusion strategies. Similarly, the model fusion level and the ANNs for one of the late fusion models are tuned using hyperband with *Keras tuner*. Note that the CNN models for each log type are generated only once and stored for later use in each fusion model. The two best architectures are shown in Figures 4 and 5. Five out of seven log measurement models (gamma-ray, resistivity, P- and S-wave sonic and PEF) share the structure shown in Figure 4. The density and neutron models also share a structure, which is illustrated in Figure 5.

### Late fusion models

#### *Late fusion average or voting ensemble*

Once we obtain the individual models/learners per log measurement type, we proceed with the preliminary assessment of the fusion models using either the validation set or the test set. Either the validation set or the test set of the individual log measurements are the inputs for each model depending on which stage we are in the fusion workflow (see Fig. 3). Their outputs are aggregated to obtain a reasonable depth shift for all log types simultaneously. This is equivalent to assigning equal weights to all of the learners/models. They, therefore, all



have the same contribution to the final estimation as shown in equation (8), presented as an aggregation function for the basic ensemble model by Perrone and Cooper (1992):

$$P = \frac{1}{M} \sum_{i=1}^M h_i(\mathbf{v}_i), \quad (8)$$

where  $h_i$  denotes each model/learner per log type,  $\mathbf{v}_i$  is the input data corresponding to each log measurement type,  $M$  is the total number of models/learners, and  $P$  is the final prediction of the fusion model.

Figure 6 shows how the mean squared error (MSE) is reduced when more than one model/learner is aggregated. This is a preliminary assessment using the synthetically shifted test set that demonstrates the potential of this technique to combine outputs from different models/learners to estimate a single output. However, we can see that the individual models 3, 4, 5 and 6 corresponding to P-wave and S-wave velocities, density and neutron, respectively, have a similar performance to the ensemble models that include them. On the other hand, models 2 and 7 (resistivity and PEF, respectively) show a much higher prediction error than the final ensemble model. This behaviour might be an indication that both velocity models and the density seem to be good enough to make predictions with a low error rate, and the opposite is seen for the resistivity, neutron and PEF.

#### Late fusion weighted average

A slightly different procedure is performed when we merge the learners using a weighted average approach. We use validation data to run an optimization process to find the best set of weights within a pre-defined search space. Afterwards, the test set is used to assess the ensemble model by comparing its results with the ground truth depth shifts. We use differential evolution (DE) as an optimization algorithm. This algorithm is characterized as being a stochastic population-based approach that is useful for global optimization problems. DE fulfils four different requirements: (i) DE can handle several cost functions regardless of their complexity including non-differentiable, non-linear and multimodal cost functions, which is an advantage over gradient-based methods. DE is a stochastic direct search. (ii) DE can solve complex computationally intensive cost functions in a reasonable time because it uses a vector population in which stochastic perturbations of each vector in the population are done independently. In other words, it is possible to parallelize the optimization process. (iii) DE has a self-organizing scheme that makes it easy to use as few parameters as possible to steer the minimization

process. For example, DE uses the difference between two randomly chosen population vectors to perturb a third existing vector. This perturbation is done for every vector within the population by comparing the perturbed result to the best. (iv) DE has a consistent convergence property obtained in different trials (Storn and Price, 1997). Details of the mathematical derivations of DE are presented in the Appendix.

We use the DE algorithm implemented in the function *differential\_evolution* from the SciPy Python library (The SciPy community, 2021). This function requires both a cost function and a set of weights to be evaluated as inputs, and it returns a score to be minimized. We specify the cost function as the MSE of the fused models, and we also specify the bounds of the optimization process. The bounds are a seven-dimensional hypercube specified using seven weights for the seven CNN models or learners with values ranging from 0.0 to 1.0. We set a maximum of 1000 iterations and a tolerance of  $10^{-7}$  to ensure convergence of the algorithm. The final weights are normalized and used for the depth shift inference.

#### Late fusion with a linear learner (stacking model)

We deploy a stacking model by testing different meta-learner algorithms. We use the term meta-learner to refer to the machine learning algorithm used to merge several outputs from the initial models. In our case, the initial models are the best 1D CNN models/learners corresponding to each log measurement type. We first test linear algorithms such as linear regression with Lasso and Ridge regularization. The regularization term is added to the cost function, working as a stabilizer of the least-squares algorithm, by penalizing the estimated parameter coefficients that best fit the data to a straight line. This penalization term could be either the L1 or L2-norm (Gareth *et al.*, 2013). In the context of stacking models, we set the linear model as follows. First, our level 0 consists of the seven models/learners (1D CNN) already defined. Second, our level 1 consists of the predictions made by our level 0 learners on a validation set. These predictions are the training set for our level 1 learner or meta-learner. Third, we assess the trained meta-learner model by inputting our semi-synthetic test set into the entire system. We do the same with real unseen data to predict the single depth shift for all the log measurements.

The training of the linear meta-learner is set as given pairs of  $n$  samples  $\{x_{ij} \in \mathbb{R}^M \text{ and } y_i \in \mathbb{R}\} 1 \leq j \leq M$  (prediction on the validation set)  $M = 7$  and  $1 \leq i \leq n$ . The linear regression fits a linear model for an  $i$ th sample given as

$$y_i = W_0 + W_1x_{i1} + W_2x_{i2} + \dots + W_7x_{i7}, \quad (9)$$

where  $W_0$  is the bias term and  $(W_1, W_2, \dots, W_m)$  are the coefficients learned during the minimization of the residual sum of squares (RSS) or cost function. We use the outputs of the level 0 learners as the variables or features  $x_{ij}$ , while the coefficients are equivalent to the weights used in the late fusion with a weighted average. Equations (10) and (11) show the two different regularization terms for Lasso and Ridge regressions (second term in the equations), respectively, added to the RSS (first term in the equations).

$$\sum_{i=1}^n \left( y_i - W_0 - \sum_{j=1}^M W_j x_{ij} \right) + \lambda \sum_{j=1}^M |W_j|, \quad (10)$$

$$\sum_{i=1}^n \left( y_i - W_0 - \sum_{j=1}^M W_j x_{ij} \right) + \lambda \sum_{j=1}^M W_j^2, \quad (11)$$

where  $\lambda \geq 0$  is a tuning parameter that controls the constraint or shrinkage of the coefficients towards zero. The main advantage of shrinking the coefficients is the variance reduction of the model, which could lead to making it simpler, more interpretable in the case of a large number of variables, and reducing overfitting when a feature selection is performed (Gareth *et al.*, 2013). When  $\lambda = 0$ , the shrinkage term has no effect, and the result will be equivalent to a standard linear regression. In contrast, when  $\lambda \rightarrow \infty$  the value of the penalty terms grows, the coefficients of the linear regression will approach zero, and for the Lasso regression some of the estimated coefficients will be forced to be equal to zero. This is the main difference between the Ridge and Lasso regressions; the former does not perform any variable or feature selection since it uses all the estimated coefficients in the final model despite large values of  $\lambda$ . On the other hand, when Lasso regression sets some coefficients to zero, we automatically remove the variables associated with those coefficients from the model, performing an automatic feature selection (Gareth *et al.*, 2013).

We tune the  $\lambda$  parameters via 10-fold cross-validation. The values of  $\lambda$  we test range from 0.0001 to 10 and increase with a step of 0.01. After tuning, the optimal  $\lambda$  values for Lasso and Ridge regression are both equal to 0.0001. Table 1 shows the regression coefficients obtained for Lasso and Ridge regression for well 16/1-9 and their equivalent values for the DE optimization. Similar values of the regression coefficients are obtained for well 16/1-21 S. Note that the output weights from DE are normalized, that is, their values range between 0 and 1 and their sum is equal to 1. The regression coefficient values and the optimized weights show roughly similar trends. No weights or negative coefficients are assigned to the resistivity model, little contribution is made to the final pre-

dictions by the PEF, gamma-ray and P- and S-wave sonic logs, and a large contribution is made by the density and neutron log models.

#### Late fusion with a non-linear learner (stacking model)

We test a multi-layer perceptron algorithm as a meta-learner. This algorithm is also known as a type of ANN. We use the outputs of the level 0-learners as training datasets for the level 1 learner (ANN). The ANN needs tuning of many parameters, as well as the proper selection of the network topology. We, therefore, perform an extra step to produce a small validation set that can be evaluated during the tuning process of the extra ANN. The original 5% of the data providing the test set is halved, and now 2.5% of the halved dataset is used for validation of the ANNs' tuning parameters. The tuning of the ANNs is performed using *Keras banked* in *TensorFlow* and a hyperband search strategy. Table 2 shows the search range of the tuned parameters. The number of hidden layers was constrained to be either 1 or 2 since more complex networks are not needed for this problem. Note that in Tables 2 and 3, we provide the maximum and minimum values that could be selected from the parameter search space, and for no numerical values like the type of activation function, we call them option 1 and option 2. The final ANN topology is presented along with the whole stacked model structure in Figure 7. Note that the seven learners of level 0 are merged with the ANN (learner of level 1); however, their weights and topologies are frozen, meaning that these CNNs are not updated or affected during the meta-learner (ANN)'s training.

#### Model-level fusion

This fusion method aims to combine the different log types into a single model to reproduce a unique depth shift per sample that can correct misalignments for all the log types simultaneously without computing additional averages manually. To perform this test, we use the seven different types of logs as simultaneous head inputs into the network, each of which passes through convolutional blocks, whose architectures are the same as the individual models already defined in the previous test. In other words, the best model architecture for each log type is implemented without further changes to build this new integrated model. The fusion of the models is performed after flattening (transformation into a one-dimensional array) of each convolutional block output followed by their concatenation as described by equation (5). The hyperparameter tuning process affects only the topology

of the fully connected layers and the optimizer (e.g., the learning rate) since we keep the convolutional block structures fixed. However, the weights for the entire structure will be updated during the training process. The search space for the dense layer tuning is summarized in Table 3.

### Depth shift estimations

At this stage, all the fusion models are ready to make estimations of the depth shift in a real dataset. We use our unseen samples from well 16/1-9 (total number of samples = 28) and 16/1-21 S (total number of samples = 32) as inputs for all the fusion models and compare their performance. The lack of ground truth labels on the real dataset made necessary the use of some qualitative and quantitative measurements for evaluation. First, we use the Pearson correlation, denoted as PC, between the reference EWL log, and the shifted LWD log before and after depth, matching using the results from the different fusion CNN strategies. Second, we compute their mean values over the total number of samples per log measurement type. For example, we obtain average values of the Pearson correlation for gamma-ray, resistivity, P- and S-wave sonic, density, neutron and PEF log types. Third, we compare the performance of each fusion strategy for the individual log types based on the mean values of the Pearson correlation, and we select the best strategy model for that log type. Fourth, the Pearson correlation achieved by the best fusion method is compared to that from the cross-correlation workflow with a user-assisted weighted average and from the manual depth matching performed by an experienced petrophysicist. We also carry out a qualitative assessment of the different depth matching approaches, using well log profiles to evaluate the results of depth matching by visual comparison of the logs before and after depth matching. We refer the interested reader to the Appendix for further details about the Pearson correlation, which we use to assess and rank the machine learning and other matching methods and results.

## RESULTS

We present the results of the different model fusion strategies for well log depth matching for wells 16/1-9 and 16/1-21 S. We compare the mean of Pearson correlations (PCs) computed for each of all the samples per log type per log to obtain an overview of each fusion method performance. We compare five different fusion strategies. They are late fusion average, late fusion weighted average, late fusion linear Ridge regression, late fusion artificial neural networks (ANNs) and model-

level fusion. We do not show the late fusion linear Lasso regression results since its performance is the same as that of the Ridge regression. Therefore, it does not add any value to the final comparisons. Tables 4 and 5 summarize the PC values and their standard deviations. We use bold type and \* to highlight the highest mean values for the PC for each log type and to draw attention to relatively high standard deviations, respectively. These notations also apply to Tables 6 and 7.

The tabulated results are displayed in Figure 8. This helps us to see that there is not a significant difference in performance between the fusion strategies. The same trend is visible for the fusion strategies' performance at the log type level. For example, we see that all the fusion strategies achieve a high overall PC for gamma-ray, resistivity, and neutron logs, whose values range are 0.83–0.84, 0.84–0.86 and 0.71–0.73, respectively, for well 16/1-9 (Fig. 8a). The density log shows moderate PC values of 0.59–0.61, and poorer results are seen for the P- and S-wave sonic logs, and the photoelectric factor (PEF) where the PC values range from 0.28 to 0.49, indicating a low correlation between the logging while drilling (LWD) and electrical wireline logging (EWL) logs after depth matching. Table 4 also shows the standard deviation of the PC values. The highest standard deviations are associated with the log types that yield lower correlation values. In most cases, the standard deviation has the same order of magnitude as the mean or is even higher than the mean as it is for the S-wave sonic with values ranging from 0.41 to 0.44. These high standard deviations indicate poor matching in some samples of these log types and large uncertainty in the results.

Well 16/1-21 S shows similar results to those from well 16/1-9. However, the PC values for all the log types except PEF are slightly higher. For instance, gamma-ray, resistivity, density and neutron mean PC values show the following ranges 0.88–0.9, 0.9–0.93, 0.83–0.85, and 0.73–0.75, respectively. The PC value for the P-wave sonic shows an increase between 0.08 and 0.1 compared to well 16/1-9. On the other hand, the PEF PC values remain the lowest and have actually decreased slightly compared to well 16/1-9, ranging from 0.34 to 0.37 (Fig. 8b).

Table 5 presents the mean PC values and their standard deviations from well 16/1-21 S. We see that the average standard deviations for the P-wave sonic and the PEF are 0.33 and 0.37, respectively. In contrast, the standard deviations for the resistivity, gamma-ray and density are slightly lower than for well 16/1-9 with average values of 0.15, 0.08, and 0.14, respectively. The corresponding standard deviations of the same log types for well 16/1-9 are 0.23, 0.14, and 0.22, respectively. The values for the neutron log are similar in both wells.

Based on these results, we select the fusion strategy that has the highest PC values for most log types in both wells. The late fusion weighted average fulfils this requirement. It presents the highest PC value in five out of seven log types for well 16/1-9 and four out of six for well 16/1-21 S. We also make a direct comparison of the mean PC between LWD and EWL logs before depth matching (original data) and after depth matching using this fusion strategy, cross-correlation weighted average workflow and manual depth matching. The results are presented in Tables 6 and 7 for wells 16/1-9 and 16/1-21 S, respectively. Figure 9 shows the results for both wells. We can see a clear improvement in the correlation between the logs after any depth matching methodology is applied to the data.

For well 16/1-9, the highest PC values are achieved by the cross-correlation weighted average method for gamma-ray, resistivity, density and PEF, which are equal to 0.86, 0.87, 0.63, and 0.39, respectively (see Fig. 9a). The late fusion weighted average (convolutional neural network [CNN] fusion strategy) has a slightly higher PC value than other methods for the neutron logs. This value is 0.73 compared to 0.71 and 0.69 from the cross-correlation and manual depth matching, respectively. Manual depth matching shows the highest values for the P- and S-wave sonic logs of 0.58 and 0.34, respectively. However, in general, the sonic logs and the PEF logs are poorly correlated after depth matching. They also show relatively high standard deviations for all the matching methods used (see Table 6).

(Figure 9b) shows the results for well 16/1-21 S from which we see improvements after depth matching for most of the log types. Additionally, the manual depth matching method seems to outperform the cross-correlation and the CNN fusion strategy for gamma-ray, P-wave sonic, density and neutron logs with PC values of 0.95, 0.79, 0.91, and 0.83, respectively. Cross-correlation shows higher PC values for resistivity and PEF logs of 0.94 and 0.39, respectively. However, the differences between these correlation values and their corresponding values for the manual depth matching are only 0.01 and 0.02, respectively. Small differences are also seen between the cross-correlation and the CNN fusion strategy PC values, suggesting a possible similar performance. Table 7 shows the standard deviations of the PC for well 16/1-21 S, high standard deviation values of 0.32 and 0.31 persist for the P-wave sonic match using cross-correlation and the CNN fusion strategy, respectively while the standard deviation for the manual method is only 0.17. PEF logs always show the same large standard deviation values of between 0.37 and 0.38 in this well, regardless of the method used. The reason for this

could be associated with the physics of the borehole. The mud systems used to condition boreholes have a high barite content in the North Sea. Barite has a high capture cross section for gamma rays and has a significant impact on the PEF measurements. The PEF might display considerable differences between LWD and EWL logs because the barite concentration could slightly change with time, affecting the rock properties (regardless of whether the rock is permeable or not).

## DISCUSSION

As we have seen, the different model fusion strategies applied to the individual log type convolutional neural network (CNN) models achieve apparently similar results under the assumption that a single common depth shift is sufficient to adequately synchronize all the log types simultaneously. However, there exist some differences between strategies in regard to model complexity and, therefore, execution times. Examples of relative execution times are summarized in Table 8, where we can see late fusion average, linear Ridge regression, and artificial neural networks (ANNs) that take the same time to train via the hyperband tuning search strategy using the structures that we proposed in this work. These three models need at least 2 hours and a half to train. For the model-level fusion, which is a single multi-input head CNN model the training time is shorter and equal to around 2 hours. Note that the seven convolutional blocks do not change their architecture only experience updates in their weights. In contrast, we have the late fusion weighted average that takes almost six times more time than the others, requiring 12 and a half hours to find the best weights to aggregate the individual models. We also see that the additional time to perform Ridge regression or define an optimal ANN shallow structure to carry out the aggregation of the models is negligible. Considering the execution times and that all models perform equally well, it might be better to select the model that takes the shortest time to be trained as the best model instead. In addition, it would be possible to reduce the execution times for the late fusion weighted average model by using parallelization techniques during the optimization process, making it more competitive compared to the other models in terms of execution time.

From Figures 9(a,b), as well as Tables 6 and 7, we see that for well 16/1-9 the highest Pearson correlation (PC) values are given by the cross-correlation user-assisted workflow and for well 16/1-21 S this is accomplished by the manual depth matching. However, the difference across the methods does not exceed 0.1 in PC for most of the log types, which is not significant. We only see a slight difference in PC of 0.24

between the manual depth matching and the cross-correlation, and a difference of 0.22 between the manual depth matching and the CNN fusion strategy for the P-wave sonic log in well 16/1-21 S. Small differences and higher correlation between logging while drilling (LWD) and electrical wireline logging (EWL) log pairs after depth matching regardless of the matching method used, are indications of CNNs' potential to provide acceptable depth matching results for a massive amounts of data simultaneously without user intervention.

The disadvantages of using a CNN fusion approach instead of a cross-correlation user-assisted workflow or a manual depth matching can be summarized as follows. First, the selection of the weights for the cross-correlation workflow is based on the user's knowledge, and hence the quality of the weights' values relies on the quality of the user's criteria. This selection of the weights is carried out for each drilled well section individually, which means that for each drilled section we have a different and independent set of weights assigned by the user. These weights are tailored to the borehole conditions at each specific section when the logs were acquired. In contrast, our CNN fusion strategy implementation estimates and uses a single set of weights/regression coefficients for the whole depth range. It does not try to account individually for local changes due to variations in the drilling and logging process between sections. In other words, our CNN fusion approach could be improved by using depth-dependent weights/regression coefficients to correct for differences between well sections and significant changes in borehole conditions. This could be accomplished by using another type of neural network that can capture the long depth dependencies. For example, Chen and Jin (2016) developed a multi-modal fusion strategy called conditional attention that can handle different modalities at each time step using a long-short term memory recurrent neural network. In this way, they could extract the long time dependencies and obtain different weights for each modality at each time step based on current input features and history information (Chen and Jin, 2016). Furthermore, we have shown in previous work (Torres Caceres *et al.*, 2022b) that changes in the borehole conditions have an impact on the distortion of the log patterns, thus discrepancies between LWD and EWL logs arise affecting the depth matching results. For example, in the case of the photoelectric factor (PEF) log which has a shallow depth of investigation, measurements are highly sensitive to the mud type, especially when the mud contains barite. Figure 10 shows an example of this type of problem where the P-wave sonic and the PEF logs (panels b and c) experience substantial changes in the log patterns while the gamma-ray does

not (panel a). We also see that the three depth matching methods converge to a similar solution, giving a good matching for the gamma-ray but poor matching for the P-wave sonic and PEF.

Second, the manual depth matching has an advantage over the cross-correlation user-assisted workflow and the CNN fusion approach since it can handle any depth-dependent shifts that could be present in the data. The manual depth matching is also performed individually for each log. This means that in the presence of substantial stretch/squeeze effects due to the stick-slip of the logging cable, for example, it is still possible to achieve a good depth match using manual adjustments. In such cases, the cross-correlation fails to provide a high correlation result because it is limited to a single constant bulk shift. Similarly, our CNN fusion model will face the same problem since depth-dependent effects are not included in the training. This problem can be alleviated by simulating and introducing stretch/squeeze effects into the training set and adding additional input information like cable head tension and surface tension logs, which are used to identify stick-slip zones. These modifications should help the algorithm recognize these events and find better solutions that align/synchronize the corresponding log patterns while remaining a fully automatic depth matching workflow. We also compare the mean PC after depth matching using the individual CNN models that were described for use on each log type (Torres Caceres *et al.*, 2022b) to the results using the CNN fusion approach and evaluate the difference between using a single depth shift across all measurements and using individual depth shifts per measurement (see Fig. 12). These comparisons help to investigate the reasons for the higher PC values of the manual depth match over the other methods, particularly for the P-wave sonic logs since the sonic logs seem to have larger relative depth shifts compared to other measurements, as shown in Fig. 13. Therefore, the single depth shift might underestimate the sonic log shifts (see Figs. 11b and 13).

For well 16/1-9, if we compare the mean PC for the P-wave sonic log using the CNN model trained only with P-wave sonic logs (0.57) and the manual shift depth matching (0.58), we see that the difference in PC is only of 0.01. However, the difference between the CNN fusion approach and the manual depth matching is 0.1. For well 16/1-21 S, we have an overall mean PC value of 0.72 using a CNN training only with P-wave logs and an overall mean PC value of 0.79 using the manual depth matching with a difference of 0.05. Therefore, the difference between the CNN fusion approach and the manual matching is 0.22. Note that for gamma-ray, resistivity and the sonic logs (P and S wave) there is a general drop in the PC



values using the CNN fusion approach compared to the CNN models trained on each log type in well 16/1-9. In contrast, the density, neutron and PEF experience a slight increase in correlation when using the CNN fusion approach instead of the individual CNN models (see Fig. 12a).

A similar pattern is seen in well 16/1-21 S (see Fig. 12b) with the difference that only the neutron has an increase in correlation with the use CNN fusion approach. However, the differences between the CNN fusion approach and the individual CNNs are small and have a minimum value of 0.01 (gamma ray) and a maximum value of 0.07 (P-wave sonic). Having a single depth shift for all log measurements implies a trade-off between poorer quality logs (density, neutron and PEF) and better quality logs (gamma ray, resistivity and sonics). It seems that in a well with data quality problems the CNN fusion approach could provide a reasonable depth matching result, whereas in a well with higher data quality logs the individual CNN models seem to provide slightly better results.

Figure 13(a,b) shows the log profiles for P-wave sonic logs before and after depth matching, using a CNN model trained on P-wave sonic logs and the CNN late fusion approach for both wells, respectively. Their corresponding PC values are depicted in a bar plot (Fig. 13c). In both cases, the independent CNN approach achieves better alignments and higher PC values than the CNN late fusion approach. These findings emphasize our observations from Figures 11 and 12, which suggest that the sonic logs in some sections of the wells seem to present larger depth shifts than the other measurements.

Pattern differences due to significant changes in the borehole conditions or any tool failures during logging could affect our results (Torres Caceres *et al.*, 2022b). For example, well 16/1-9 is reported to have had several operational problems during log acquisition. Therefore, we see that the three depth matching methods tested have similar performances while in well 16/1-21 S, where the logs are of better quality, the manual depth matching has higher PC values than both the CNN fusion approach and the cross-correlation workflow. Also, we should remember that the manual depth matching will be highly dependent on the analyst's experience and more prone to subjectivity.

We believe that our CNN fusion workflow can be considerably improved. Also, our workflow can be implemented as a fast first alignment of the log measurements, aiming to build a fully automated depth matching tool. We also envisage that our workflow could be implemented in a cloud-based system while integrated into a database. For example, thousands of LWD logs can be quickly depth matched with their

corresponding thousands of EWL logs effortlessly and without any user intervention. This auto depth matching function will open possibilities to use all data in a database instead of just using the 10 most essential logs.

## CONCLUSION

We implemented and compared several multimodal machine learning fusion strategies. These combined one-dimensional convolutional neural networks (1D CNNs) for depth matching of logging while drilling (LWD) and electrical wireline logging (EWL) well log pairs. Fusion combined models described for different log measurement types to estimate a single common depth shift to be applied to all LWD logs. Our implementation aimed to reduce user intervention and fully automate the depth matching workflow based on cross-correlation. For all the models, we used the same semi-synthetic training, validation and test datasets. All models were compared using the same unseen real data from two wells located in the Ivar Aasen field in the Norwegian North Sea. The individual 1D CNN models corresponding to each log measurement such as gamma-ray or resistivity are the same for all the fusion tests. We focused on evaluating late fusion models and model-level fusion by comparing their mean values of the Pearson correlation (PC) between samples from pairs of logs before and after depth matching. We showed that there are no substantial differences in the PC values between the fusion strategies implemented in this work. However, differences were seen when we compared their training execution times. For example, the fastest model to train was the model-level fusion, which took about 2 hours, followed by the late fusion average, late fusion linear Ridge regression, and the late fusion artificial neural networks (ANNs), which all took 2 and  $\frac{1}{2}$  hours. The slowest model was the late fusion weighted average which took about 12 and  $\frac{1}{2}$  hours.

Additionally, we compared the fusion strategy with the highest PC values (late fusion weighted average) with a cross-correlation user-assisted workflow and manual depth matching results. We showed that all the methods increase the mean PCs of the log samples compared to the data before depth matching. Also, all the methods converge to a similar result with some exceptions. For example, the manual matching always outperforms the other methods for the P-wave sonic logs in both wells. This is likely associated with the slightly larger relative depth shifts affecting the sonics logs compared to the other measurements. Therefore, the single solution from the models' aggregation will have large residuals for the sonic logs that are well addressed using manual adjustments or

CNN models trained only on P-wave sonic logs. The CNN fusion approach's results show a trade-off between poor quality logs and better quality logs. This trade-off suggests that in wells with data quality problems this strategy might provide reasonable depth matching. In wells with higher quality logs, the single CNN models might show slightly better depth matches. The cross-correlation user-assisted workflow shows slightly higher PC values than the other methods for well 16/1-9. The manual depth matching also shows slightly higher PC values than the other methods for well 16/1-21 S. Although our fusion approach is not superior in performance relative to other depth shift estimation methods, it is fully automatic, competitive, unbiased to human perception (relevant in unitization questions), and we also have identified potential improvements. Our future research will include tests of depth-dependent weights/regression coefficients to correct the difference between well sections and local changes in borehole conditions, by using long-short term memory recurrent neural networks capturing long depth dependencies. This approach will update the weights for the models' aggregation at each depth step whenever needed. In addition, we would like to investigate the introduction of depth-dependent shifts into the training data to overcome the limitations of the cross-correlation method, obtaining some of the benefits of manual depth matching in an automated context.

Finally, we recommend using our CNN fusion approach as a fast depth shift screening that can be combined with the second pass of depth shift adjustments using the CNN individual models to reduce the residuals. We foresee the implementation of our workflow on a cloud-based database where we can depth match a massive number of logs simultaneously, opening new opportunities to use all data instead of being limited to the essential well logs.

## ACKNOWLEDGEMENTS

This research is part of the BRU21 – NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry ([www.ntnu.edu/bru21](http://www.ntnu.edu/bru21)) and is supported by AkerBP. We also thank NTNU-NPD-Schlumberger Petrel for providing us with the borehole dataset. Special thanks to Andrew J. Carter for his dedication to give a constructive feedback and valuable suggestions for improving this manuscript.

## DATA AVAILABILITY STATEMENT

Data are subject to third-party restrictions.

## APPENDIX

### Differential evolution algorithm

The differential evolution (DE) algorithm can be described by three stages called mutation, crossover and selection (Storn and Price, 1997). First, the DE utilizes  $NP$  (where  $NP$  is the size of the population that does not change during the minimization process) and  $D$ -dimensional parameter vectors such as

$$\mathbf{x}_{i,G}, i = 1, 2, \dots, NP, \quad (\text{A.1})$$

where each  $D$ -dimensional parameter vector represents the population of each generation  $G$ . The initial vector population is chosen randomly, and it should cover the whole parameter space. A uniform random distribution is used to make all random decisions required by the algorithm. Given a preliminary solution (target vector), the initial population is generated by adding normally distributed random deviations to the nominal solution ( $\mathbf{x}_{\text{nom},0}$ ). DE generates new parameter vectors by using groups of three existing population vectors. For example, the new parameter vector will be the result of adding to the first vector (target vector) the weighted difference between the second and third population vectors, all of them chosen randomly. This operation is called mutation, and its mathematical expression is given by equation (A.2). For each target vector within a generation  $\mathbf{x}_{i,G}, i = 1, 2, 3, \dots, NP$ , a mutated vector is generated by three randomly selected parameter vectors as follows:

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}), \quad (\text{A.2})$$

where  $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$  are the random integer indices, which are mutually different. They are constrained to be different from the running index  $i$ . Hence,  $NP$  must be greater than or equal to four.  $F$  is a real and constant factor  $\in [0, 2]$  and  $F > 0$ , which controls the amplification of the differential variation ( $\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}$ ).

Each resulting new parameter vector or mutated parameter vector is mixed with the parameters of the target vector. The result of this process yields a new vector called the trial vector. This parameter mixing process is known as crossover. This step is introduced to increase the diversity of the perturbed parameter vectors. The trial vector is defined by equation (A.3):

$$\mathbf{u}_{i,G+1} = (\mathbf{u}_{1i,G+1}, \mathbf{u}_{2i,G+1}, \dots, \mathbf{u}_{Di,G+1}), \quad (\text{A.3})$$

where

$$\mathbf{u}_{j,G+1} = \begin{cases} \mathbf{v}_{j,G+1} & \text{if } (\text{randb}(j) \leq CR) \text{ or } j = \text{rnbr}(i) \\ \mathbf{x}_{j,G+1} & \text{if } (\text{randb}(j) > CR) \text{ and } j \neq \text{rnbr}(i) \end{cases}, \quad (A.4)$$

$j = 1, 2, \dots, D,$

where  $\text{randb}(j)$  is the  $j$ th evaluation of a uniform random number generation with an outcome  $\in [0,1]$ .  $CR$  is the crossover constant that must be given by the user and  $\in [0,1]$ . Generally, the crossover is generated by a binomial random distribution. Additionally,  $\text{rnbr}(i)$  is a randomly chosen index  $\in 1, 2, \dots, D$  that ensures that  $\mathbf{u}_{i,G+1}$  gets at least one parameter from  $\mathbf{v}_{i,G+1}$ .

The selection is the step in which the trial vector  $\mathbf{u}_{i,G+1}$  is evaluated, and the decision of whether or not it should be part of the next generation  $G+1$  is made. To do this,  $\mathbf{u}_{i,G+1}$  is compared with the target vector  $\mathbf{x}_{i,G}$  using a greedy criterion. If  $\mathbf{u}_{i,G+1}$  achieves a lower cost function than  $\mathbf{x}_{i,G}$ , then the target vector for the next generation  $\mathbf{x}_{i,G+1}$  will be  $\mathbf{u}_{i,G+1}$ , otherwise the old value  $\mathbf{x}_{i,G}$  is retained.

The whole process is repeated until each population vector has served once as a target vector, meaning that  $NP$  competitions take place in one generation (Storn and Price, 1997).

## Metric for depth matching assessment


### Pearson correlation coefficient

The Pearson correlation coefficient (PC) is defined as the covariance of two variables or distributions divided by the product of their standard deviations. PC is used as a metric, which indicates the degree of relationship between two variables. For example, if  $x$  and  $y$  are independent the covariance and PC will be both equal to zero. In the case in which one variable accurately determines the other, such that all the points  $(x_i, y_i)$  lie perfectly on a straight line with either a positive or negative slope PC will be equal to 1 or  $-1$ , respectively. These statements are valid, assuming that there is a linear relation between them (Bulmer, 1979). The Pearson correlation coefficient is described by the following equation:


$$PC = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (A.5)$$

where  $n$  is the sample size,  $x_i$  and  $y_i$  are the individual sample points of the depth series, and  $\bar{x}$  and  $\bar{y}$  are the corresponding mean values of  $x$  and  $y$ .

## ORCID

Veronica Alejandra Torres Caceres 

<https://orcid.org/0000-0002-5011-591X>

Kenneth Duffaut  <https://orcid.org/0000-0003-3091-9598>

## REFERENCES

- Abdoli, S., Cardinal, P. and Koerich, A.L. (2019) End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications*, 136, 252–263. DOI: 10.1016/j.eswa.2019.06.040
- Arundo Analytics, Inc. Revision 8d539c82.. (2020) tsaug documentation: Examples of augmenters. Available at: <https://tsaug.readthedocs.io/en/stable/notebook/Examples%20of%20augmenters.html> [Accessed 20 April 2021].
- Baltrušaitis, T., Ahuja, C. and Morency, L.-P. (2018) Multimodal machine learning: a survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 423–443. DOI: 10.1109/TPAMI.2018.2798607
- Bateman, R.M. (1986) *Openhole Log Analysis and Formation Evaluation*. Society of Petroleum Engineers.
- Bolt, H. (2016) Wireline logging depth quality improvement: methodology review and elastic-stretch correction. *Petrophysics*, 57(3), 294–310.
- Bolt, H. (2019) Correction of driller's depth: field example using driller's way-point depth correction methodology. *Petrophysics*, 60(1), 76–91. DOI: 10.30632/PJV60N1-2019a7.
- Brazell, S., Bayeh, A., Ashby, M. and Burton, D. (2019) A machine-learning-based approach to assistive well-log correlation. *Petrophysics*, 60(4), 469–479. DOI: 10.30632/PJV60N4-2019a1
- Bulmer, M.G. (1979) *Principles of Statistics*, Courier Corporation.
- Chen, S. and Jin, Q. (2016) Multi-modal conditional attention fusion for dimensional emotion prediction. *Proceedings of the 24th ACM International Conference on Multimedia*, Amsterdam, the Netherlands, 15–19 October 2016. ACM. pp. 571–575. DOI: 10.1145/2964284.2967286
- Chia, C.R., Laastad H., Kostin A.V., Hjortland F., and Bordakov G.A. (2006) A new method for improving LWD logging depth. Paper 102175-MS, SPE Annual Technical Conference and Exhibition, San Antonio, TX, 24–27 September, 2006. DOI: 10.2118/102175-MS.
- Chollet, F. (2015) Keras. Available at <https://keras.io> [Accessed 24 September 2021].
- Deng, T., Xu, C., Lang, X. and Doveton, J. (2021) Diagenetic facies classification in the Arbuckle formation using deep neural networks. *Mathematical Geosciences*, 53 (7) 1491–1512. DOI: 10.1007/s11004-021-09918-0
- Gadzicki, K., Khameshashari, R. and Zetsche, C. (2020) Early vs late fusion in multimodal convolutional neural networks. *Proceedings of the 23rd International Conference on Information Fusion (FUSION IEEE)*, Rustenburg, South Africa, 6–9 July 2020. IEEE. pp 1–6. DOI: 10.23919/FUSION45008.2020.9190246
- Gareth, J., Daniela, W., Trevor, H. and Robert, T. (2013) *An Introduction to Statistical Learning: with Applications in R*. Springer.
- Imamverdiyev, Y. and Sukhostat, L. (2019) Lithological facies classification using deep convolutional neural network. *Journal of*

- Petroleum Science and Engineering*, 174, 216–228. DOI: 10.1016/j.petrol.2018.11.023.
- Ioffe, S. and Szegedy, C. (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning, Lille, France, 7–9 July 2015. *Proceedings of Machine Learning Research*, 37, 448–456
- Kerzner, M.G. (1984) A solution to the problem of automatic depth matching. Paper SPWLA-1984-VV, Proceedings of the SPWLA 25th Annual Logging Symposium, New Orleans, LA, 10–13 June 1984.
- Kingma, D.P. and Ba, J. (2015) Adam: A method for stochastic optimization. Proceedings of the Third International Conference for Learning Representations (ICRL), San Diego, CA, 7–9 May 2015.
- Le Nir, I., Van Gysel, N. and Rossi, D. (1998) Cross-section construction from automated well log correlation: A dynamic programming approach using multiple well logs. Paper SPWLA-1998-DDDm Proceedings of the SPWLA 39th Annual Logging Symposium, Keaton, CO, 26–28 May 1998.
- Le, T., Liang L., Zimmermann T., Zeroug S., and Heliot D. (2019) A machine-learning framework for automating well-log depth matching. *Petrophysics*, 60(5), 585–595. DOI:10.30632/PJV60N5-2019a3.
- Lecun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning. *Nature*, 521(7553), 436–444. DOI: 10.1038/nature14539.
- Li, L., Jamieson, K., Desalvo, G., Rostamizadeh, A. and Talwalkar, A. (2017) Hyperband: a novel bandit based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1), 6765–6816.
- Lineman, D., Mendelson, J. and Toksoz, M.N. (1987) Well to well log correlation using knowledge-based systems and dynamic depth warping. Paper SPWLA-1987-UU, Proceedings of the SPWLA 28th Annual Logging Symposium, London, UK, 29 June–2 July 1987.
- Liu, K., Li, Y., Xu, N. and Natarajan, P. (2018) Learn to combine modalities in multimodal deep learning. arXiv preprint arXiv:1805.11730.
- Luthi, S.M. and Bryant, I.D. (1997) Well-log correlation using a back-propagation neural network. *Mathematical Geology*, 29(3), 413–425.
- Münzner, S., Schmidt, P., Reiss, A., Hanselmann, M., Stiefelhagen, R. and Dürichen, R. (2017) CNN-based sensor fusion techniques for multimodal human activity recognition. *Proceedings of the ACM International Symposium on Wearable Computers*, Maui, Hawaii, 11–15 September 2017. ACM. pp. 158–165 DOI: 10.1145/3123021.3123046
- Norwegian Petroleum Directorate (NPD), (2019) Guidelines for reporting well data to authorities after completion "Blue Book". Available at [https://www.npd.no/globalassets/1npd/regelverk/forsk\\_rifter/en/b\\_og\\_b\\_digital\\_rapportering\\_e.pdf](https://www.npd.no/globalassets/1npd/regelverk/forsk_rifter/en/b_og_b_digital_rapportering_e.pdf) [Accessed 10 April 2021].
- Oliphant, Travis, and contributors (2021) Scipy.signal.correlate, <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.correlate.html>, [Accessed 24 September 2021].
- O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H. and Invernizzi, L. (2019) Keras Tuner, Github, [Accessed on 28 November 2020].
- Opitz, D. and Maclin, R. (1999), Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11, 169–198. DOI: 10.1613/jair.614
- Pedersen, B.K., Constable, M.V. and Risbakken, J.K. (2006) Operational procedures and methodology for improving LWD and WL depth control, Kristin field. Paper SPWLA-2006-XXX, Proceedings of the SPWLA 47th Annual Logging Symposium, Veracruz, Mexico, 4–7 June 2006.
- Perrone, M.P. and Cooper, L.N. (1992) When networks disagree: ensemble methods for hybrid neural networks, Institute for Brain and Neural Systems Brown University, Providence RI.
- Rider, M.H. (1986) *The Geological Interpretation of Well Logs*, Rider-French Consulting Ltd.
- Sollie, F.O. and Rodgers S.G. (1994), Towards better measurements of logging depth. Paper SPWLA-1994-D, Proceedings of the SPWLA 35th Annual Logging Symposium, Tulsa, OK, 19–22 June 1994.
- Spalburg, M. (1989) An algorithm for simultaneous deconvolution, squaring and depth-matching of logging data. Paper SPWLA-1989-P, Proceedings of the SPWLA 30th Annual Logging Symposium, Denver, Colorado, USA, 11–14 June.
- Startzman, R. and Kuo, T. (1986) An artificial intelligence approach to well log correlation. Paper SPWLA-1986-WW, Proceedings of the SPWLA 27th Annual Logging Symposium, Houston, TX, 9–13 June 1986.
- Storn, R. and Price, K. (1997) Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359.
- The SciPy community, (2021) Scipy.optimize.differential\_evolution. Available at [https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential\\_evolution.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html) [Accessed 26 April 2021].
- Theys, P.P. (1999) *Log Data Acquisition and Quality Control*. Editions Technip.
- Torres Caceres, V.A., Duffaut, K., Westad, F.O., Stovas, A., Johansen, Y.B. and Jenssen, A. (2022a) Automated log data analytics workflow—the value of data access and management to reduced turnaround time for log analysis. *Petrophysics*, 63(1), 35–60. DOI:10.30632/PJV63N1-2022a3
- Torres Caceres, V.A., Duffaut, K., Yazidi, A., Westad, F.O. and Johansen, Y.B. (2022b) Automated well-log depth matching—1d convolutional neural networks vs. classic cross correlation. *Petrophysics*, 63(1), 12–34. DOI:10.30632/PJV63N1-2022a2
- Wang, S., Shen, Q., Wu, X. and Chen, J. (2020) Automated gamma-ray log pattern alignment and depth matching by machine learning. *Interpretation*, 8(3), SL25–SL34. DOI: 10.1190/INT-2019-0193.1
- Wilson, H., Lofts J., Page G., Brooks A., and Walder D. (2004) Depth control: reconciliation of LWD and wireline depths, standard practice and an alternative simple but effective method. Paper SPE-89899-MS, Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, TX, 26–29 September 2004. DOI: 10.2118/89899-MS.
- Wolpert, D.H. (1992) Stacked generalization. *Neural Networks*, 5(2), 241–259.
- Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K. (2018) Convolutional neural networks: An overview and application in radiology. *Insights into Imaging*, 9(4), 611–629. DOI: 10.1007/s13244-018-0639-9.
- Zangwill, J. (1982) Depth matching - a computerized approach. Paper SPWLA-1982-EE, Proceedings of the SPWLA 23rd Annual Logging Symposium, Corpus Christi, TX, 6–9 July, 1982.
- Zimmermann, T., Liang, L. and Zeroug, S. (2018) Machine-learning-based automatic well-log depth matching. *Petrophysics*, 59(6), 863–872. DOI:10.30632/PJV59N6-2018a10.