Journal of Computational Science 72 (2023) 102100

Contents lists available at ScienceDirect

Journal of Computational Science

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

# Improving predictive models for rate of penetration in real drilling operations through transfer learning

Felix James Pacis [a,*], Adrian Ambrus [b], Sergey Alyaev [b], Rasool Khosravanian [c], Tron Golder Kristiansen [c], Tomasz Wiktorski [a]

[a] University of Stavanger, Stavanger, 4021, Norway[1]
[b] NORCE Norwegian Research Centre, Bergen, 5838, Norway
[c] AkerBP, Stavanger, 4020, Norway

ARTICLE INFO

*Keywords:*
Rate of penetration model
Transfer learning
Deep learning

ABSTRACT

The rate of penetration (ROP) is a key performance indicator in the oil and gas drilling industry as it directly translates to cost savings and emission reductions. A prerequisite for a drilling optimization algorithm is a predictive model that provides expected ROP values in response to surface drilling parameters and formation properties. The high predictive capability of current machine-learning models comes at the cost of excessive data requirements, poor generalization, and extensive computation requirements. These practical issues hinder ROP models for field deployment. Here we address these issues through transfer learning. Simulated and real data from the Volve field were used to pre-train models. Subsequently, these models were fine-tuned with varying retraining data percentages from other Volve wells and Marcellus Shale wells.

Four out of the five test cases indicate that retraining the base model would always produce a model with a lower mean absolute error than training an entirely new model or using the base model without retraining. One was on par with the traditional approach. Transfer learning STL allowed for reducing the training data requirement from a typical 70 percent down to just 10 percent. In addition, transfer learning reduced computational costs and training time. Finally, results showed that simulated data could be used without real data or in combination with real data to train a model without trading off the model's predictive capability.

On top of our previous work Pacis et al. (2022) from a single transfer learning, we explored continuous transfer learning (CTL) in Alvheim field wells. Due to the inherent uncertainty and dynamics of drilling data, it was no surprise that continuous retraining further reduced the error than a single transfer learning paradigm. Moreover, we investigated the effect of drilled formations and input combinations on model performance.

## 1. Introduction

According to a 2016 study by EIA [1], drilling constitutes 30%–60% of the average cost per well, which varies from $4.9 MM to $8.3 MM for onshore wells and $120 MM to $230MM for offshore wells. Thus, a modest improvement in the duration of drilling a well results in significant monetary savings. Among other factors such as preventing a non-productive time due to equipment failure or poor weather conditions, choosing the optimal drilling parameters to increase ROP is essential in reducing drilling duration.

Many attempts have been made on predicting the ROP. Although with some success [2], traditional physics-based models require frequent recalibration depending on the auxiliary data such as facies types, bit design, and mud properties [3–6]. This is challenging since facies types, in particular, are often unknown prior to drilling and would require correlation to data from nearby (offset) wells, if such wells exist.

Machine learning (ML) models try to address these challenges by using data to find correlations among many drilling variables. A study by Hegde et al. [7] showed an improvement in ROP prediction in accuracy from 0.46 to 0.84 when using random forest. Elkatatny et al. [8] also showed an improvement from 0.72 to 0.94 using an Artificial Neural Network (ANN).

Despite significant improvements in recent years, no ML approach has been widely used for ROP optimization to date [9]. The potential reason could be that the existing ML models are impractical for

* Corresponding author.
*E-mail addresses:* felix.j.pacis@uis.no (F.J. Pacis), aamb@norceresearch.no (A. Ambrus), saly@norceresearch.no (S. Alyaev), rasool.khosravanian@akerbp.com (R. Khosravanian), tron.golder.kristiansen@akerbp.com (T.G. Kristiansen), tomasz.wiktorski@uis.no (T. Wiktorski).
[1] https://stavanger.ai/

https://doi.org/10.1016/j.jocs.2023.102100
Received 15 March 2023; Received in revised form 11 May 2023; Accepted 15 June 2023
Available online 5 July 2023

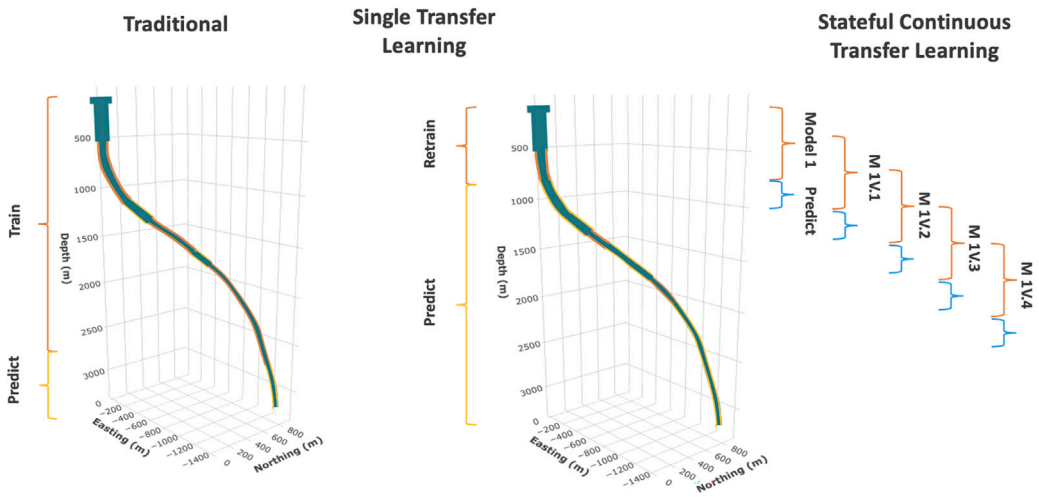


**Fig. 1.** Data utilization for traditional vs. transfer learning approaches for well data. The traditional approach uses most of the well data for training. Single transfer learning retrains a pre-trained model using a small portion of the well data. In stateful continuous transfer learning, as soon as the pre-determined well data comes, it retrains the most recent retrained model, giving several model versions for a single well.

real-time ROP prediction tasks. Developing an ML ROP model is a multidimensional problem that does not revolve solely around prediction accuracy. Higher predictive capability comes at the cost of substantial data requirements, computational constraints, and generalization capability. From a practical perspective, tackling these constraints would be desirable for several reasons.

First, the need for large datasets for training a model for every well would limit the value creation. ANN training, such as Elkatatny et al. [8] and Abbas et al. [10], would require 70% of data for training; rendering these methods essentially not applicable in real scenarios since only a fraction of a well can benefit from such approach, see Fig. 1.

Second, ML models presented by O'Leary et al. [9], Mantha et al. [11], and Hegde et al. [7] require a priori knowledge on the formations being drilled. However, this information is rarely available prior to drilling the hole. This is problematic for wells drilled in new areas where offset wells do not exist yet.

Third, ROP prediction is a real-time regression problem. Unlike physics-based models that only require pre-identification of parameters, the ML requires training before deployment. Hence, one should consider the online computation requirements.

Fortunately, ROP ML models' issues are not foreign in other domains. Deep Learning models, in general, suffer from overfitting due to insufficient training data [12]. Transfer learning (TL) is an active research field in Deep Learning that deals with reusing a model trained from a more general task, termed base model or pre-trained model, to another specific tasks, termed target model. TL techniques have been proven successful in many domains such as computer vision and natural language processing [13].

In this paper, we present the application of TL to ROP prediction. To our knowledge, this is the first application of TL in the context of drilling. We train base models using real, simulated, and combined data from previously drilled wells. Then, we reconfigure each model by freezing some model parameters to limit the number of trainable parameters. Each reconfigured base model is retrained using a small fraction of target-well data, yielding a target model. This way, a high-quality target model is already available from the drilling operation's early stage, see Fig. 1. The performance of our TL models is compared to both the base models and models trained only for the data from the new well. Furthermore, we showed the benefits of leveraging transfer learning to continuously retrain the ROP model using the most recent data using the strategies learned from Pacis et al. [14].

The paper is an extension of previous works [14,15] and is organized as follows. In Section 2, we briefly discuss the concept of TL. In Section 3, we describe the datasets and proceed with the experimental setup, including the model architecture, input data, and the method for training and retraining. We also provide an end-to-end sample application of the TL approach. Section 4 presents the results of the single and continuous transfer learning paradigm. Section 5 concludes the paper.

## 2. Transfer learning

Following the notations by Pan and Yang [13], Transfer Learning mainly involves a domain $D$ and Task $T$. The domain, denoted by $D = \{X, P^X\}$, includes two components: a feature space $X$ and a marginal probability distribution $P^X$, where each input instance is denoted by $x \in X$. On the other hand, the task, denoted by $T = \{Y, f(\cdot)\}$, includes all possible labels $Y$ and a predictive function $f(\cdot)$ that predicts a corresponding label using unseen instances $\{x*\}s$. For a two domain scenario, given a source domain $D_s$ and learning task $T_s$, a target domain $D_t$ and learning task $T_t$, where $D_s \neq D_t$, or $T_s \neq T_t$, TL leverages learned knowledge from $T_s$ to improve the $T_t$ predictive function. Subscripts s and t here corresponds to source and target, respectively.

The most common TL technique is fine-tuning [16]. In the context of ANN, fine-tuning involves reusing the whole network or freezing certain hidden layers before updating the network weights during retraining for the target task. Fine-tuning works based on the premise that Deep Learning models learn different features at different layers. Thus, reusing a pre-trained model for a target task allows better performance with less training time by starting from "near truth" parameters than training a new model with randomly initialized parameters. More so, supervised training of feedforward networks does not impose any explicit condition on the learned intermediate features. Neural network training is non-deterministic which converges to a different function every time it is run. Using a pre-trained network reduces the variance of estimation process [17].

TL has been widely used both in computer vision and Natural Language Processing [13,18]. This is apparent from the proliferation of pre-trained networks e.g., VCG-16 [19], XLNet [20], GPT-3 [21] using large datasets e.g., ImageNet[2], Giga5[3], and Common Crawl Dataset[4], and reused in domains where data is expensive or hard to obtain. For

[2] https://www.image-net.org
[3] https://catalog.ldc.upenn.edu/LDC2011T07
[4] https://commoncrawl.org/the-data/

**Table 1**
Description of datasets.

| Well name | Hole size (in) | Depth range (m) | Dataset type | Dataset source type | Test case # |
|---|---|---|---|---|---|
| F-1 A | 8.5 | 2602–3682 | Train & Val. | Sim. & Real | |
| F-1 B | 12.25 | 2603–3097 | Train & Val. | Sim. & Real | |
| F-1 B | 8.5 | 3097–3465 | Retrain & Test | Real | 4 |
| F-1 C | 12.25 | 2662–3056 | Retrain & Test | Real | 2 |
| F-1 C | 8.5 | 3067–4094 | Train & Val. | Sim. & Real | |
| F-11 A | 8.5 | 2616–3762 | Train & Val. | Sim. & Real | |
| F-11 B | 12.25 | 2566–3197 | Train & Val. | Sim. & Real | |
| F-11 B | 8.5 | 3200–4771 | Retrain & Test | Real | 3 |
| F-15 A | 17.5 | 1326–2591 | Retrain & Test | Real | 1 |
| F-15 A | 8.5 | 2656–4095 | Train & Val. | Sim. & Real | |
| F-9 A | 12.25 | 489–996 | Train & Val. | Sim. & Real | |
| F-9 A | 8.5 | 1000–1202 | Train & Val. | Sim. & Real | |
| Marcellus Shale | 8.75 | 1974–4405 | Retrain & Test | Real | 5 |
| Alvheim well A | 12.25 | 1928–3360 | Retrain & Test | Real | 6 |
| Alvheim well B | 12.25 | 2156–3490 | Retrain & Test | Real | 7 |

example in medical imaging, Shin et al. [22] fine-tuned AlexNet [23] — a pre-trained network using ImageNet dataset [24] with more than 14 million images belonging to around 20 thousand categories. They successfully achieved 85% sensitivity at 3 false positive per patient in thoraco-abdominal lymph node (LN) detection and interstitial lung disease (ILD) classification. Another successful application, Bird et al. [25] used a simulated scene from a computer game to train a model and resulted in an improvement for the real-world scene classification task.

Pre-trained networks also catalyzed the recent advances in Natural Language Processing (NLP). For example, Devlin et al. [26] introduced Bidirectional Encoder Representations from Transformers (BERT), which can be fine-tuned with adding an output layer to create state-of-the-art models for a wide range of tasks. Successful applications of BERT include text summarizing [26], modeling clinical notes and predicting hospital readmission [27], and machine reading comprehension [26].

The success of TL is apparent from its ubiquitous applications. This motivated websites, such as Hugging Face[5] and Model Zoo[6], which provide a platform to access many open-sourced pre-trained networks with ease.

TL has yet to be explored and applied broadly in the oil and gas domain. Since well-annotated datasets are expensive and difficult to obtain in the oil and gas industry, TL can be used to make rapid progress in this domain [28].

## 3. Experimental setup and data

### 3.1. Methodology

TL requires the base model to be trained from several unique wells to improve their generalization capability. We freeze selected layers in these base models to keep the original weights and allow some to be trainable. These reconfigured layers are then fine-tuned using a pre-determined percentage of data from target wells. Hyperparameters during fine-tuning are carefully chosen to prevent vanishing or exploding gradients. This happens when the distribution of retraining data is entirely different and the learning rate is too high; this impairs the base model's performance. In addition, a new model is also trained using the same retraining data. All these models are then tested using the remaining data from the target well.

### 3.2. Datasets

We used well data from four sources: real field data from Volve, Marcellus shale and Alvheim, and synthetic data. Table 1 summarizes the datasets. The well name, hole size, hole depth range, and the data source type for each dataset are provided for reference.

In general, when drilling an oil and gas well, large diameter holes are drilled first, followed by smaller holes until they reach the predefined target. This is done to maintain well integrity, particularly when transitioning to a new geologic formation. Drillers use different drill bits, bit designs, and drilling fluid properties at each new hole size. This is similar to drilling an entirely new well from an engineering perspective. Thus, we produce independent datasets by segregating the data according to hole size from each well. These datasets contain recorded real-time drilling parameters such as hookload, stand pipe pressure, hole depth, weight on bit, mud weight, and rotations per minute. The frequency of these measurements varies for every well depending on the equipment used.

#### 3.2.1. Volve

In 2018 Equinor publicly shared raw real-time drilling data from 20 wells found in the Volve field in the North Sea [29], together with well logging data, surveying data, drilling reports, and other auxiliary information. Pre-processed Volve drilling logs can be found in a public data repository [30]. For this paper, we selected drilling data from 7 wells and separated them according to the hole size. In total, we compiled 12 independent datasets for the experiment. Volve data has an average sampling frequency of 0.4 Hz, corresponding to a time step of 2.5 s.

#### 3.2.2. Marcellus shale

Marcellus shale is the most prolific natural gas-producing formation from the Appalachian basin in the United States [31]. A site owned and operated by Northeast Natural Energy, LLC provides several horizontal wells drilled in the Marcellus shale [32]. A specific long horizontal well spanning 2431 m, with an average measurement frequency of 0.176 Hz or 5.67 s time step, was chosen for the current study. This well data allows testing the models' generalization and re-usability outside Volve data.

#### 3.2.3. Alvheim

Located in the central part of the North Sea, the Alvheim area consists of the Boa, Vilje, Volund, Bøyla, and Skogul fields. AkerBP provided well data from two specific wells for this study. These infill mid wells were drilled to access attic oil remaining above existing producers. While drilling these wells, complex hole instability issues were encountered, and relative to the Volve well data used, Alvheim wells have higher variation in the ROP; thus, it was chosen for the CTL approach viability. Unfortunately, due to privacy issues, only limited information is published, and well names are anonymized.

[5] https://huggingface.co
[6] https://modelzoo.co/

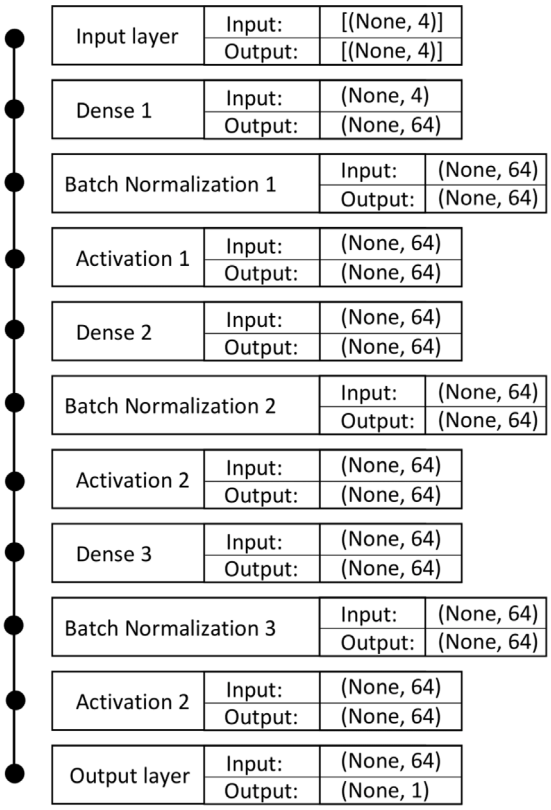


**Fig. 2.** Model architecture.

### 3.2.4. Synthetic data

To provide additional training data and investigate the feasibility of using simulated training data for the TL application, we generated eight synthetic datasets using a state-of-the-art drilling simulator which includes advanced hydraulics, mechanics, and heat transfer models [33]. The well architecture, trajectory, drilling mud properties, drill string configuration, and formation properties were based on the drilling reports extracted from the Volve public database [29]. The drilling set points (top-drive rotary speed, weight on bit, and flow rate) used as input to the simulations were based on the values from the Volve recorded drilling logs compiled by Tunkiel [30]. The simulation outputs were stored as time-series with a time step of 1 s.

## 3.3. Setup of experiments

### 3.3.1. Single transfer learning

Our model starts with an input layer, which receives four input parameters, followed by three successive pairs of dense and batch normalization layers. By embedding normalization as part of the model architecture, this prevents internal covariate shift [34] and causes a more predictable and stable behavior of the gradients [35], allowing higher learning rates without the risk of divergence [34,35]. In addition, batch normalization eliminates the need for Dropout [12] for regularization [34]. We use rectified linear unit [36] as activation function. Finally, the output layer is a single-output dense layer with mean squared error as the loss function. A complete and detailed network structure is shown in Fig. 2.

To predict ROP, we used stand pipe pressure (SPP), weight on bit (WOB), mud weight (MWin), and top drive rotations per minute (RPM). We based these inputs from the setup described in Ambrus et al.'s work [37]. In general, when choosing our input parameters, two considerations were in place: first, despite using an ANN, the choice of input parameters should still reflect the physics of the drilling process.

**Table 2**
Training and validation data shapes.

| Data source type | TrainX | TrainY | ValX | ValY |
|---|---|---|---|---|
| Real | (333400,4) | (333400,1) | (83350,4) | (83350,1) |
| Simulated | (649503,4) | (649503,1) | (162376,4) | (162376,1) |
| Combined | (982903,4) | (982903,1) | (245726,4) | (245726,1) |

Second, selected inputs must always be available. During drilling, hundreds of parameters and metadata are recorded in real-time [38]. The inclusion of many drilling parameters as inputs to the ANN could be helpful but at the same time dangerous when one or more of these parameters are missing for the current well due to sensor failure or they were not necessarily recorded during the operation. Although one might infer the missing values, this would increase the model's prediction uncertainty when there are many inferred values.

Eight datasets were selected to build the base models out of the 12 available well sections from Volve. These were carefully selected to ensure that they contain values of the upper and lower boundaries of each input and output parameter. For example, the dataset with the highest ROP and lowest ROP values should be among the chosen eight.

To avoid overfitting the model, the first 80 percent of each well section is concatenated into the training dataset, whereas the remaining 20 percent are used for validation. This was done to all the three data source types — real, simulated, and combined. The shapes of concatenated training and validation data are shown in Table 2.

Each drilling parameter varies in range wildly depending on the units of measurement. When left unscaled, some features would dominate the training making it difficult for the neural network to learn the underlying patterns in the data. For example, during drilling, the rotation of the drilling tool is measured in RPM (revolution per minute), which would give a range of two to three-digit values, and for weight on bit measured in kg could reach five-digit values, the weight-on-bit would have more effect on the training parameters than the RPM. However, this is not true, particularly when drilling softer formations where the RPM is more important than the weight-on-bit. In essence, by scaling the input parameters, each can affect the model parameters during the backpropagation [39]. Thus, the data were scaled using a MinMaxScaler from Scikit-Learn [40] before passing to the model. This removes the harmful effects of having different value ranges for every input variable by scaling all of them to a (0,1) range. The introduction of batch normalization further stresses the importance of scaling the input.

Scaling the output is unnecessary since ROP is a function of the input, and the network will learn to map the input to the correct output during training. Since the final activation function constrains the model's output, we did not use any activation function on the last layer — it was only a single output-dense layer.

Three separate runs for each data source type were conducted to build base models while keeping the model's hyperparameters the same. In particular, batch size, which is the number of samples that is propagated through the network before updating the internal model parameters, was chosen to be 10000. This is relatively small, around 1 to 3 percent of each training dataset, to increase variation in batch statistics, thereby enabling better model generalization during the retraining process [41]. An early stopping callback was placed to cease training when the validation loss stops improving after 100 epochs. This allows us to generate two distinct base models for each data source type: one base model with the best validation loss and another based on the training loss. Altogether, we train six base models.

The four remaining well sections from real Volve data and the Marcellus shale horizontal well are used for retrain and test data. A sensitivity analysis was done by creating independent datasets with different retrain: test data ratios. These vary from 30:70, 20:80, 10:90, and 5:95, where the smaller partition corresponds to retraining data.

**Table 3**
Number of trainable and non trainable parameters.

| Model configuration | Trainable parameters | Non-trainable parameters |
|---|---|---|
| Base Model | 8897 | 384 |
| Zero Frozen Layers | 8513 | 768 |
| One Frozen Layer | 8257 | 1024 |
| Two Frozen Layers | 4161 | 5120 |

Similar to the data preprocessing used for building the base model, each dataset is split sequentially and the values are scaled to a (0,1) range.

During retraining, we kept a similar model architecture to the base models, except that some layers were frozen. This allows us to retrain the model using smaller training datasets since fewer parameters are retrainable, and at the same time, model parameters are not initialized randomly. In addition, since models are pre-trained, a low learning rate is needed to reach the global minima. In our case, we used a learning rate of 0.0001 for all instances, with the exception of test case 4 that used $10^{-9}$. Maximum epochs are set at 150000 for tests. Similar to training the base model, we set up an early stopping at 50 epochs based on the training loss.

These base models are reconfigured in three ways: freezing the first dense layer, first and second dense layers, and keeping all dense layers unfrozen. This gives us 18 reconfigured base models for retraining. All batch normalization layers were frozen in all these configurations to prevent the risk of vanishing or exploding gradients. In this context, freezing a layer means keeping the parameters learned during the initial training stage. Table 3 shows the number of trainable and non-trainable parameters for each configuration.

A randomly initialized new model with similar model architecture and hyperparameters was trained for every retraining data configuration. This is to compare the performance of fine-tuning a pre-trained model with that of a new model trained from scratch on the same dataset.

#### 3.3.2. Continuous transfer learning

Otherwise stated, similar fine-tuning techniques from the single transfer learning were implemented. One difference in data preparation is that instead of using a percentage as the unit of amount of data, we expressed it by the number of drilling pipe stands, similar to our previous work [14]. A pipe stand consists of two to three pipe joints and is considered a unit. The drilling stands connect the drilling bit to the top of the drilling rig. In addition, pipe stands are hollow where the drilling fluid passes through. We used pipe stands as a data unit for reproducibility since each well varies in length and sensor measurement frequency; thus, using percentage could be misleading. Retraining data to test data ratio are fixed in 10:1 stands, i.e., the first ten drilling stands were used for retraining, and the model was tested on the 11th stand. Then, this $n$th stand window moves until we exhaust all the available data. All hyperparameters are the same as with single transfer learning. On top of our previous work [15] we performed sensitivity analysis on the optimal input parameters. For this paper, all CTL experiments are performed on the Alvheim field since it has more variation in ROP than Volve field wells.

### 3.4. Model evaluation

#### 3.4.1. Single transfer learning

We have six unique base models from previous sections, wherein each was reconfigured in three configurations based on the number of frozen layers. This gives us 18 unique models on top of the base models plus an entirely new trained model. In total, for every retraining data configuration, e.g., one test well, with unique retrain:test ratio, we tested 25 different models.

Model performance is evaluated by computing the mean absolute error ($MAE := L_1$) and root-mean-square error ($RMSE := L_2$) for every test data configuration:

$$L_k = \left( \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|^k \right)^{\frac{1}{k}} \tag{1}$$

where $N$ is the number for data points, $\hat{y}_i$ is the true value, and $y_i$ is the predicted value. In addition, we kept a summary of a moving window MAE by dividing the test data into ten equal windows with the exact count of data points and computing MAE at each window. This enables us to measure prediction quality for various test data sizes. It is also important to emphasize that every well section data has a varying frequency of measurements and size, e.g., 30 percent of the well F-1C 12.25 in. section contains fewer data points than the F-11B 8.5 in. section.

#### 3.4.2. Continuous transfer learning

CTL models were evaluated similarly to the STL, except that the CTL was evaluated on the current single stand. This is contrary to STL, where it was evaluated on all the remaining well data excluded for model retraining. Fig. 1 displays the test data for a given well in STL and CTL approach.

### 3.5. Example of usage

#### 3.5.1. Single transfer learning

As discussed in Section 3.3, we train six base models then reconfigure by freezing layers. Subsequently, we derive four different datasets from well F-15 A 17.5 in data. Each dataset differs on the retrain and test ratio as described previously. Each of the 18 reconfigured base models is then fine-tuned using retraining data from every dataset. In addition, we train an entirely new model using similar retraining data. From here, we have a total of 25 distinct models — 6 base models, 18 reconfigured base models, and one new model. These 25 models are then used for predicting ROP on the test datasets. Having 4 data split ratios from well F-15 A 17.5 gives us a total of 100 test runs. For every test case, MAE is recorded.

#### 3.5.2. Continuous transfer learning

Each well dataset is subdivided into stands. Starting from the simulated base model, we fine-tune the model using the first ten stands and test on the 11th stand. We then move the window of retraining and testing stand until we exhaust all the data for a given well. After the initial retraining, succeeding models utilize the newly retrained model as its new base rather than starting from the simulated base model. This is in accordance with our results from Pacis et al. [14] that in most cases, stateful continual learning (which retrains the model from one state to the next as more data becomes available) showed the most robust results. Thus, for a well drilled with n stands, there will be n-11 unique retrained models. For every test case, MAE is recorded.

## 4. Results and discussion

In Section 4.1.1, we analyze the results on well F-15 A 17.5 in and compare the best models from several methods, which includes fine-tuning, training of an entirely new model, and direct use of the base model. This section's results obtained are representative of both Volve and Marcellus shale test cases. Data from the other four wells can be found in the Appendix. In Section 4.1.2, we provide recommendations based on the results of five test cases. In Section 4.1.3, we provide results on the generalization capability of the approach. Section 4.2 discusses the results of the continuous transfer learning approach. Input sensitivity and the effect of formation on the model are also presented.

**Table 4**
Test Case 1: F-15A 17.5 in.

| Frozen layers | BM loss type | Source type | Retrain (%) | Train (%) | MAE | RMSE |
|---|---|---|---|---|---|---|
| 0 | Validation | Simulated | 30 | N/A | 3.674 | 4.886 |
| 2 | Validation | Simulated | 5 | N/A | 4.104 | 5.238 |
| 1 | Validation | Combination | 30 | N/A | 4.165 | 5.656 |
| 2 | Validation | Simulated | 30 | N/A | 4.234 | 5.562 |
| 1 | Validation | Simulated | 30 | N/A | 4.268 | 5.87 |
| *New Model | Training | Real | N/A | 30 | 5.061 | 5.845 |
| *Base Model | Validation | Simulated | N/A | 80 | 9.091 | 10.767 |

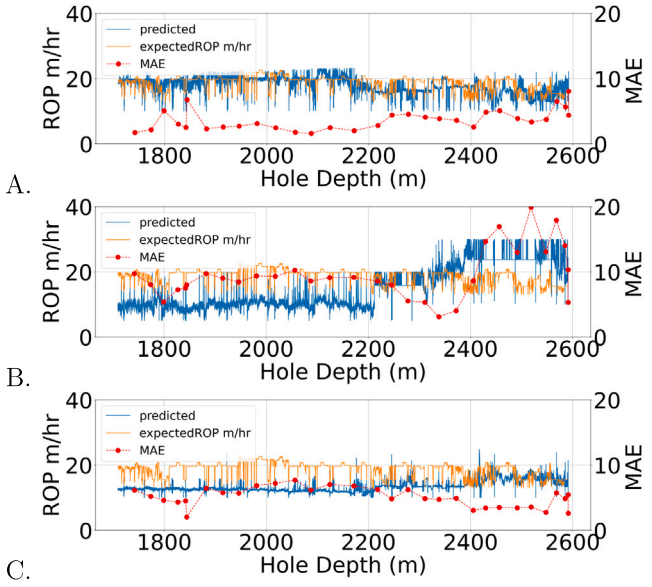


**Fig. 3.** ROP predicted by different models for well F-15 A 17.5 in. A. Fine tuned model with TL. B. Base (pre-trained model) without fine-tuning. C. Newly trained model only using the data from the current well. Orange and blue lines refer to expected and predicted ROP values, respectively. Red markers are the computed MAE moving window e.g., one red marker is the MAE of the previous 2500 observations. All data are plotted against hole depth on the $X$-axis. Fine-tuned model performed best among other models with an MAE of 3.674.

### 4.1. Single transfer learning

#### 4.1.1. 3-Way-comparison

After testing 100 models, we plot the predicted vs. expected ROP values plus a moving MAE window. In each plot, $X$-axis represents the hole depth, $Y$-axis to the left is the ROP with m/hr unit, and $Y$-axis to the right is the MAE. A, B, and C plots in Fig. 3 show the best model among fine-tuned base models, base models, and new models, respectively. Model configurations and metadata of these models can be found in Table 4.

Retraining the base model reduces the MAE by 59.6% and 27.4% vs. using the base model and training an entirely new model, respectively. A relatively close RMSE to MAE also indicates that the ROP error disperses equitably across the data. Despite the base model not being trained with the same 17.5-inch hole size, it outperforms other models by tuning with the current well data. This is also on top of the fact that model A has fewer trainable parameters. Although not seen on the plot, the second-best model overall has an MAE of 4.104, despite only using 5% retraining data and 55% fewer trainable parameters compared to training a new model. Furthermore, both of the best two models were pre-trained using simulated data.

#### 4.1.2. Recommendations based on all test cases

**Training data source type.** Four out of the five test cases suggest that training with simulated data provides better result than training with real data in terms of MAE. One explanation for this could be that predictions are less noisy since the simulated data is deterministic; thus, it produces better results when re-trained on a small section of the test set.

**Base Model loss type.** Four out of the five test cases suggest that the best model should be based on the best validation loss rather than training loss. This is expected since the early stopping based on the validation loss helps reduce overfitting on the training data. Although the best retrained model in test case 3 was obtained using the training loss criterion, the base model using the validation loss criterion does not come far behind when considering the MAE.

**Number of Frozen Layers.** Four out of the five test cases suggest that fine-tuned models always perform the best. Case 4 performed just as well as the base model. Although, there was no clear relation between the number of frozen layers and the MAE. Paradoxically, increasing the number of frozen layers also increased the retraining time by 43 up to 247 percent. Thus, from retraining time perspective the ROP prediction problem benefits more from a pre-trained network without frozen layers. Another observation is that models with zero, one, and two frozen layers took an average retraining time of 7, 12, and 15 min, respectively, versus base models' 22 min.

**Retraining data percent.** As mentioned previously, every test well has a different length; therefore, even having the same retraining data percentage, the number of data points would still vary. There is no clear correlation between the number of data points and retraining data percentage for the best fine-tuned model based on the five test cases, although one could say that there could be a slight trade-off between the accuracy of the model and the length of the well to be predicted.

During our experiments we also observed that TL was sensitive to the choice of base model training data and learning rate. However the detailed analysis is out of the scope of this paper.

#### 4.1.3. Marcellus shale: Test outside volve data

We tested the approach on the Marcellus shale dataset to evaluate the generalization and re-usability of the TL approach. This well is entirely distinct from Volve data in terms of well profile (horizontal), type of formation (shale), location (onshore well), and equipment used. This is analogous to recognizing between breeds of dogs and breeds of cats for the computer vision domain. Clearly, the best-retrained model reduced the MAE by 29% and 19% when compared with the newly trained model and base model, respectively. Relative to other test cases from Volve data, computed MAE is higher because of noise and lower measurement frequency in the Marcellus data. On top of improving the MAE, the retrained model only used 20% of retraining data while decreasing the trainable parameters by 10%. Furthermore, this result was achieved by training the base model with simulated data. This demonstrates the potential of using synthetic data generated with a high-fidelity drilling simulator for training the ANN ROP model that can be reconfigured for real operations with a minimal amount of retraining.

### 4.2. Continuous transfer learning

We compared the performance of a continuous and non-continuous training paradigm. The red line in plot A in Fig. 4 is the model retrained only until the 29th stand, while the blue line implements a continuous training paradigm using the most recent ten stands. The red line has a higher error in 9 out of 10 test cases. This shows the benefit of the continuous transfer learning paradigm as observed by Pacis et al. [14].

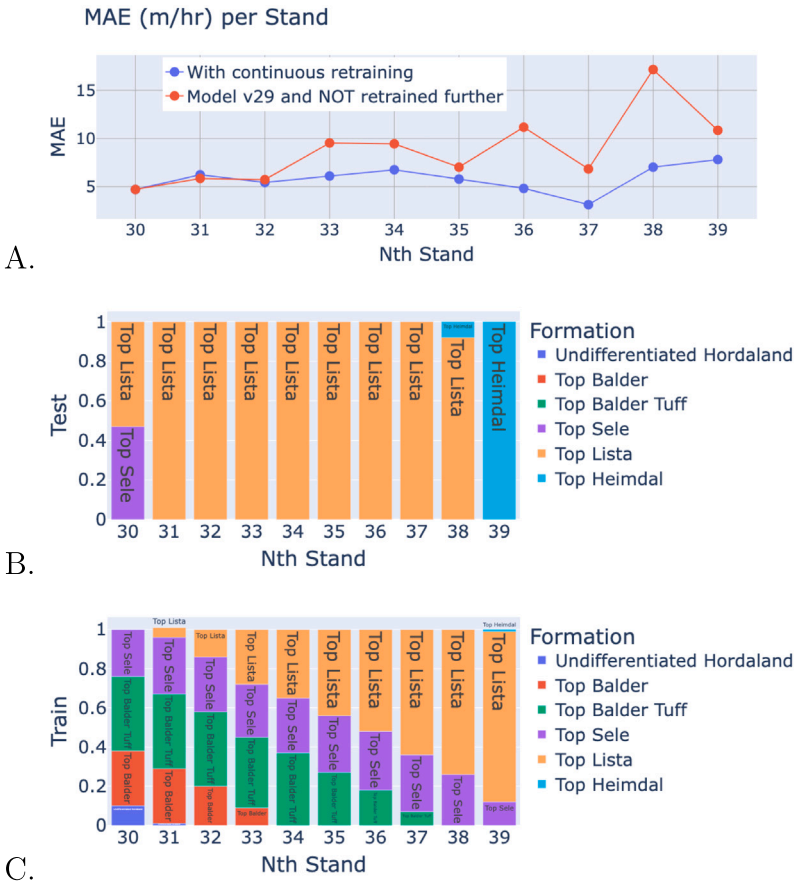


**Fig. 4.** Formation effect.

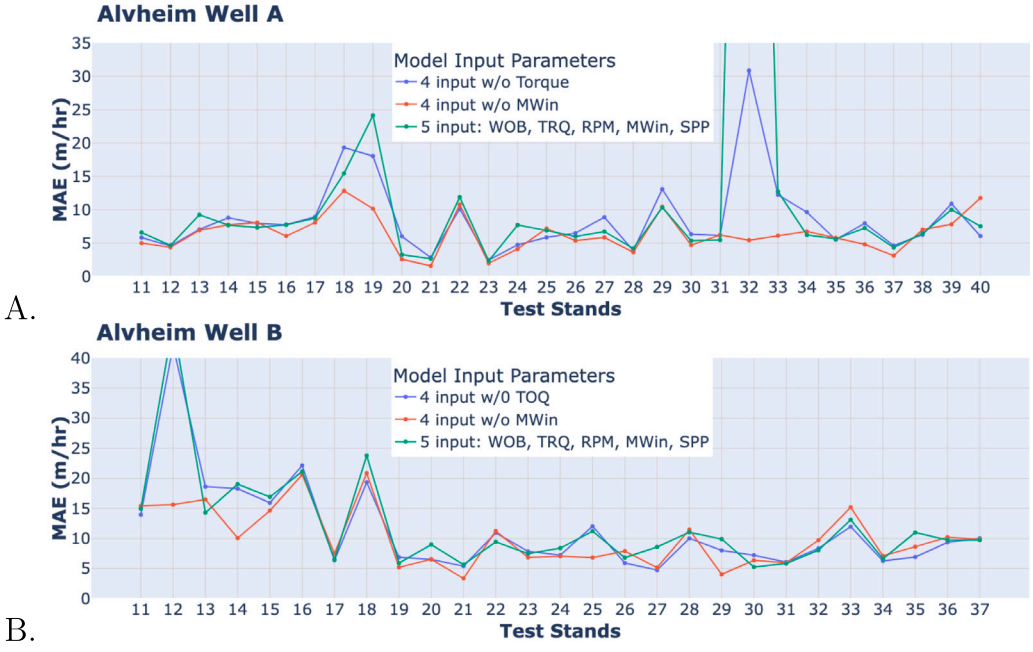


**Fig. 5.** Input sensitivity on Alvheim Well A and Alvheim Well B.

#### 4.2.1. Effect of formation

Charts B and C in Fig. 4 show the composition of formations drilled in testing and training data, respectively. For example, at the 30th stand, the test data comprises Top Sele and Top Lista formation. On the other hand, the corresponding retraining data comprises the ten most recent stands, i.e., 20th to 29th, which drilled four different formations. It is imperative to note that the ratio accounts for the number of data points; hence, it does not directly translate to the actual thickness of the formation. This discrepancy is caused by two factors inherent in drilling data, drilling duration at a specific depth and frequency of measurements. As we drill ahead, it is clear that the ratio of the Lista formation linearly increases upon reaching the Heimdal reservoir formation.

From plot A in Fig. 4, looking at the blue line starting from the 34th stand, the MAE reduced until the 37th stand and increased thereafter. Validating the results on the formation effect, it is clear that all the test cases are Top Lista formation while the ratio of Top Lista on the retraining data increases. Thus, it is no surprise that the fine-tuned models are calibrated and more biased towards the Top Lista formation, as shown by the error trend.

#### 4.2.2. Input sensitivity

Previous work [15] used WOB, RPM, MWin, and SPP as input parameters. We performed sensitivity analysis on the optimal input parameters by including the surface torque. Among these five input parameters, MWin rarely changes and does not reflect the downhole condition. However, it indirectly affects the ROP by aiding in lifting cuttings and ensuring wellbore stability. For these reasons, we decided to experiment with omitting MWin.

Plots A and B in Fig. 5 show the input sensitivity analysis on wells A and B from the Alvheim field, respectively. Both test wells showed that omitting the MWin makes every model more stable (red line). Both test cases showed a similar trend for most of the test stands.

## 5. Conclusions

We presented the application of TL for ROP prediction in oil and gas drilling. We trained, retrained, and tested 100 models for each of the five test wells. Based on MAE evaluation, the TL approach for four out of five test wells outperforms both the newly trained model and the non-fine-tuned base model. The TL was on par with the traditional approach for the fifth well.

We explored the best model configurations based on the five test cases. In most cases, the best results were obtained with the base models trained on the simulated data. Moreover, the validation loss seems to indicate the model's performance on the new well.

During fine-tuning, pre-trained models with zero frozen layers converged faster, although there was no clear relation between the MAE and the number of frozen layers. Despite uncertainty in the optimal number of frozen layers and retraining data percentage, results indicate that transfer learning is valuable in developing an adaptable, reusable, and more general ROP prediction model.

We also demonstrated that transfer learning could be used to continuously recalibrate an ROP model and adapt it to the current downhole drilling condition.

## Declaration of competing interest

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

## Data availability

Data will be made available on request.

## Acknowledgments

This work is part of the Center for Research-based Innovation DigiWells: Digital Well Center for Value Creation, Competitiveness and Minimum Environmental Footprint (NFR SFI project no. 309589, DigiWells.no). The center is a cooperation of NORCE Norwegian Research Centre, the University of Stavanger, the Norwegian University of Science and Technology (NTNU), and the University of Bergen, and funded by the Research Council of Norway, Aker BP, ConocoPhillips, United States, Equinor, Lundin, TotalEnergies, and Wintershall Dea.

## Appendix. Test results

Tables A.5 to A.8 show the best five fine-tuned models, base model, and new model on each test well. Each table has seven columns that state the number of frozen layers, the type of loss used for the base model, the data source type used for training the base model, the retraining data percentage, the training data percentage, MAE, and root mean squared error (RMSE).

**Table A.5**
Test Case 2: F-1C 12.25 in.

| Frozen layers | BM loss type | Source type | Retrain (%) | Train (%) | MAE | RMSE |
|---|---|---|---|---|---|---|
| 0 | Validation | Simulated | 30 | N/A | 5.006 | 6.644 |
| 1 | Validation | Simulated | 30 | N/A | 5.765 | 9.36 |
| 2 | Validation | Simulated | 20 | N/A | 6.416 | 9.008 |
| 0 | Validation | Combination | 10 | N/A | 7.161 | 11.324 |
| 0 | Training | Real | 20 | N/A | 7.522 | 14.608 |
| *New Model | Training | Real | N/A | 30 | 6.211 | 7.845 |
| *Base model | Validation | Real | N/A | 80 | 5.22 | 7.575 |

**Table A.6**
Test Case 3: F-11B 8.5.

| Frozen layers | BM loss type | Source type | Retrain (%) | Train (%) | MAE | RMSE |
|---|---|---|---|---|---|---|
| 0 | Training | Simulated | 10 | N/A | 9.083 | 11.696 |
| 0 | Training | Combination | 20 | N/A | 9.222 | 12.436 |
| 1 | Validation | Real | 10 | N/A | 9.653 | 12.526 |
| 1 | Validation | simulated | 20 | N/A | 9.878 | 13.077 |
| 0 | Training | Combination | 10 | N/A | 9.912 | 12.732 |
| *New Model | Training | Real | N/A | 10 | 10.277 | 12.949 |
| *Base Model | Validation | Simulated | N/A | 80 | 10.271 | 13.097 |

**Table A.7**
Test Case 4: F-1B 8.5 in.

| Frozen layers | BM loss type | Source type | Retrain (%) | Train (%) | MAE | RMSE |
|---|---|---|---|---|---|---|
| 2 | Validation | Real | 30 | N/A | 5.898 | 7.498 |
| 1 | Validation | Real | 30 | N/A | 5.899 | 7.498 |
| 0 | Validation | Real | 30 | N/A | 5.899 | 7.498 |
| 0 | Validation | Real | 20 | N/A | 7.179 | 9.266 |
| 1 | Validation | Real | 20 | N/A | 7.185 | 9.275 |
| *New Model | Training | Real | N/A | 20 | 10.752 | 12.323 |
| *Base Model | Validation | Real | N/A | 80 | 5.9 | 7.498 |

**Table A.8**
Test Case 5: Marcellus shale 8.75 in.

| Frozen layers | BM loss type | Source type | Retrain (%) | Train (%) | MAE | RMSE |
|---|---|---|---|---|---|---|
| 1 | Validation | Simulated | 20 | N/A | 33.151 | 46.631 |
| 2 | Validation | Real | 20 | N/A | 33.444 | 48.093 |
| 0 | Validation | Combination | 20 | N/A | 34.679 | 48.575 |
| 2 | Training | Combination | 30 | N/A | 35.921 | 46.091 |
| 1 | Training | Combination | 10 | N/A | 36.164 | 48.261 |
| New Model | Training | Real | N/A | 30 | 46.42 | 52.231 |
| *Base Model | Validation | Real | N/A | 80 | 41.092 | 49.633 |

## References


[1] Trends in U.S. oil and natural gas upstream costs, 2022, https://www.eia.gov/analysis/studies/drilling/pdf/upstream.pdf. (Accessed 13 January 2022).

[2] C. Soares, H. Daigle, K. Gray, Evaluation of PDC bit ROP models and the effect of rock strength on model coefficients, J. Nat. Gas Sci. Eng. 34 (2016) 1225–1236.

[3] G. Bingham, A new approach to interpreting rock drillability, Tech. Man. Repr. Oil Gas J. (1965) 93.

[4] W. Maurer, The perfect-cleaning theory of rotary drilling, J. Pet. Technol. 14 (1962) 1270–1274.

[5] G. Hareland, P. Rampersad, Drag-bit model including wear, in: SPE Latin America/Caribbean Petroleum Engineering Conference, OnePetro, 1994.

[6] H.R. Motahhari, G. Hareland, J. James, Improved drilling efficiency technique using integrated PDM and PDC bit parameters, J. Can. Pet. Technol. 49 (10) (2010) 45–52.

[7] C. Hegde, H. Daigle, H. Millwater, K. Gray, Analysis of rate of penetration (ROP) prediction in drilling using physics-based and data-driven models, J. Pet. Sci. Eng. 159 (2017) 295–306.

[8] S. Elkatatny, A. Al-AbdulJabbar, K. Abdelgawad, A new model for predicting rate of penetration using an artificial neural network, Sensors 20 (7) (2020).

[9] D. O'Leary, D. Polak, R. Popat, O. Eatough, T. Brian, First use of machine learning for penetration rate optimisation on elgin franklin, in: SPE Offshore Europe Conference & Exhibition, OnePetro, 2021.

[10] A.K. Abbas, S. Rushdi, M. Alsaba, Modeling rate of penetration for deviated wells using artificial neural network, in: Abu Dhabi International Petroleum Exhibition & Conference, OnePetro, 2018.

[11] B. Mantha, R. Samuel, ROP optimization using artificial intelligence techniques with statistical regression coupling, in: SPE Annual Technical Conference and Exhibition, OnePetro, 2016.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[13] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 1345–1359 (2010).

[14] F.J. Pacis, S. Alyaev, A. Ambrus, T. Wiktorski, Exploration of strategies to improve continual learning from irregular sequential drilling data, in: International Conference on Ocean, Offshore & Arctic Engineering, 2023

[15] F.J. Pacis, S. Alyaev, A. Ambrus, T. Wiktorski, Transfer learning approach to prediction of rate of penetration in drilling, in: International Conference on Computational Science, Springer, 2022, pp. 358–371.

[16] D. Sarkar, R. Bali, T. Ghosh, Hands-on Transfer Learning with Python: Implement Advanced Deep Learning and Neural Network Models using TensorFlow and Keras, Packt Publishing Ltd, 2018, pp. 163–165.

[17] D. Erhan, A. Courville, Y. Bengio, P. Vincent, Why does unsupervised pre-training help deep learning? in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, in: JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.

[18] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data 3 (1) (2016) 1–40.

[19] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[20] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Adv. Neural Inf. Process. Syst. 32 (2019).

[21] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, 2020, arXiv preprint arXiv:2005.14165.

[22] H.-C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (5) (2016) 1285–1298.

[23] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.

[25] J.J. Bird, D.R. Faria, A. Ekárt, P.P. Ayrosa, From simulation to reality: CNN transfer learning for scene classification, in: 2020 IEEE 10th International Conference on Intelligent Systems, IS, IEEE, 2020, pp. 619–625.

[26] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[27] K. Huang, J. Altosaar, R. Ranganath, Clinicalbert: Modeling clinical notes and predicting hospital readmission, 2019, arXiv preprint arXiv:1904.05342.

[28] Y. Hajizadeh, Machine learning in oil and gas; a SWOT analysis approach, J. Pet. Sci. Eng. 176 (2019) 661–663.

[29] Equinor, Volve field data (CC BY-NC-SA 4.0), URL https://www.equinor.com/en/news/14jun2018-disclosing-volve-data.html.

[30] A. Tunkiel, Selected work repository, 2020, https://www.ux.uis.no/~atunkiel/.

[31] Independent Statistics and Analysis, U.S. Energy Information Administration, Marcellus shale play: Geology review, 2017.

[32] Marcellus Shale Energy and Environment Laboratory, 2022, http://mseel.org. (Accessed 11 January 2022).

[33] J.E. Gravdal, R. Ewald, N. Saadallah, S. Moi, D. Sui, R. Shor, A new approach to development and validation of artificial intelligence systems for drilling, in: 2020 15th IEEE Conference on Industrial Electronics and Applications, ICIEA, IEEE, 2020, pp. 302–307.

[34] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456.

[35] S. Santurkar, D. Tsipras, A. Ilyas, A. Madry, How does batch normalization help optimization? in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 2488–2498.

[36] R.H. Hahnloser, R. Sarpeshkar, M.A. Mahowald, R.J. Douglas, H.S. Seung, Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit, Nature 405 (6789) (2000) 947–951.

[37] A. Ambrus, S. Alyaev, N. Jahani, T. Wiktorski, F.J. Pacis, Rate of penetration prediction using quantile regression deep neural networks, in: International Conference on Ocean, Offshore & Arctic Engineering, Volume 10: Petroleum Technology, American Society of Mechanical Engineers, 2022, V010T11A010.

[38] A.T. Tunkiel, T. Wiktorski, D. Sui, Drilling dataset exploration, processing and interpretation using volve field data, in: International Conference on Offshore Mechanics and Arctic Engineering, Vol. 84430, American Society of Mechanical Engineers, 2020, V011T11A076.

[39] F. Chollet, Deep Learning with Python, second ed., Manning Publications, 2021.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (Oct) (2011) 2825–2830.

[41] D. Masters, C. Luschi, Revisiting small batch training for deep neural networks, 2018, arXiv preprint arXiv:1804.07612.

**Felix James Pacis** is a Ph.D student in Deep Learning at the University of Stavanger (UiS) where he also finished his master's degree in drilling and well engineering. He also has a B.Sc. degree in Petroleum Engineering from the Palawan State University in the Philippines.

The main objective of their project is to leverage Artificial Intelligence (AI) to build adaptive data-driven models for drilling and positioning wells. Addressing the practical bottlenecks of data-driven models for field implementation is also part of their research.

**Adrian Ambrus** is a senior researcher at the Norwegian Research Center (NORCE). His research focuses on drilling modeling.

**Sergey Alyaev** is a senior researcher at the Norwegian Research Center (NORCE). His interests include Forward Modeling, Applied Machine Learning, Data assimilation, and Real-time decisions.

**Rasool Khosravanian** works at AkerBP ASA where he is part of the Drilling and Wells Digital Domain. His research focuses on drilling and well technology automation and data solutions.

**Tron Golder Kristiansen** Tron works at AkerBP ASA where he is the Chief Engineer of D&W Rock Mechanics.

**Tomasz Wiktorski** is a Professor in data technology at the University of Stavanger. His research focuses on analysis of large time series data combining conventional time series methods and deep learning approaches with data-intensive techniques from other domains. Primary application areas include: biomedical data analysis (in particular: sport watches and other personal monitoring devices); oil and gas drilling process automation and optimization; prediction and optimization in energy and petroleum systems; monitoring and optimization of data centers and cloud infrastructures; other large time series data (e.g. sensor networks, smart city).