Proceedings of the ASME 2022
41st International Conference on Ocean, Offshore & Arctic Engineering
OMAE2022
June 5 – June 10, 2022, Hamburg, Germany

OMAE2022-78423

UNSUPERVISED MACHINE LEARNING: A WELL PLANNING TOOL FOR THE FUTURE

Peter Batruny[1], Tim Robinson[2],

[1]PETRONAS, Kuala Lumpur, Malaysia
[2]Exebenus, Stavanger, Norway

ABSTRACT

In recent years, the industry has sought insights from abundant data generated by drilling operations. One of the key focus areas is the rate of penetration (ROP) which impacts costs directly, and emissions indirectly. Previous work has succeeded in predicting and optimizing ROP, however was limited to specific fields and small-scale applications. This limitation stems from unobserved information between different fields or operations that often impacts model usability. This paper provides a new way of well planning by leveraging the power of unsupervised machine learning to deliver higher drilling efficiency, lower costs, and less uncertainty.

Unsupervised machine learning techniques are used to infer information about drilling operations from real-time data that is not directly measured. Certain well types seem to be separable in low dimensions, based on qualitative interpretation of clusters visualized in 2D, and according to analyzing cluster membership based on which wells the data came from, and by associating common characteristics of wells within clusters using external information.

This work introduces a novel approach to collect, separate, and extract value from data that is otherwise unused. Data synthesized by Variational Autoencoders could be used for enhanced well planning, sold as a standalone product, or shared between industry players with reduced privacy concerns, increasing the wealth of data available.

Keywords: Unsupervised Machine Learning, Drilling Operations, Artificial Intelligence

NOMENCLATURE

BHA	Bottom Hole Assembly
MT	Mill Tooth
PDC	Polycrystalline Diamond Compact
RSS	Rotary Steerable System
TD	Total Depth
TCI	Tri-cone Insert
UCS	Unconfined Compressive Strength
ML	Machine Learning
VAE	Variational Autoencoder
WITSML	Wellsite Information Transfer Standard Markup Language

1. INTRODUCTION

Drilling operations generate millions of data points daily. This comes in the form of Wellsite Information Transfer Standard Markup Language (WITSML). WITSML changed the game by allowing service providers and operators to communicate in a standard medium. This enabled real time operation monitoring, allowing intervention and decision making to be made based on real time information such as lithology, downhole drilling dynamics, and formation pressure [1]. The value of WITSML data is not only in its real time application, but also in the vast amounts of time-based or depth-based data points and their associated attributes. This historical data has been used for benchmarking, continuous improvement, reference as offset wells, and database building for future applications. However, sorting through millions of data points with more than 10 attributes each is not a task that can be performed by a human within a reasonable time frame. This results in data that is diligently compiled, but not to be used again [2]. Maintaining large databases with decades of stagnant data is wasteful and expensive. Thus, the drilling industry was quick to adopt machine learning techniques to process the large amounts of available data and provide meaningful insights, extracting every ounce of value.

The application of machine learning techniques to drilling projects can be summarized into two main parts: supervised machine learning, and unsupervised machine learning. A supervised machine learning model is trained on existing data with attributes labelled by the user. The model is trained to find a relationship between a labelled input vector, and a labelled output vector. This model can then be used to simulate the outcome of different combinations of input that have not yet occurred. Supervised machine learning allows users to impart their knowledge onto the machine, providing a sort of guidance until the model can identify relationships with sufficient fidelity.

In the oil and gas industry, this technique has been successfully used to predict and optimize drilling rate of penetration [3], predict formations based on real time log data [4], assist in seismic interpretation [5], and predict stuck pipe risks [6,7], among others. To successfully apply supervised machine learning techniques and obtain insights from drilling data, the user still must go through a lengthy process of cleaning, segregating, and labelling the data before it is used as input. For the process to be truly automated, the data from the WITSML feed should be vetted, segregated, and labelled by a program ahead of its use in a supervised machine learning model.

Unlike supervised techniques, unsupervised machine learning does not need data to be labelled by a user [8]. Popular unsupervised machine learning models are K-means, self-organizing maps, hierarchal clustering, and random forest [9]. While varying in accuracy [10], all these techniques aim to extract hidden attributes and correlations from data without a guiding hand. From data through WITSML feed alone, is it possible to identify and group wells with similar bottom hole assemblies (BHAs), drilled similar formations, or used similar rigs? Once that is possible, a labelled dataset for classification modelling can be constructed based on the cluster memberships assigned to each datum. This data can then be used in a supervised model to generate output, completing the automation cycle from rig to machine learning model.

Wells can be drilled on land, offshore in shallow water, or offshore in deep water. These wells can be vertical or directional. Directional wells can use a bent motor or a rotary steerable system (RSS). Land wells can have nothing in common with each other than the lack of sea water. An ultra-deepwater well can drill the same formations using the same bottom hole assembly as a shallow water well, but 2000 meters deeper. Due to the many different possible combinations, grouping similar wells becomes a multivariate problem as opposed to a bivariate one, clustering via unsupervised machine learning models would be an elegant solution. A hypothesis on what is defined as "similar" and "cluster" wells is therefore also needed. While it is difficult to anticipate what the machine will define as a cluster, a thought experiment on what may influence the decision can help verify the results of the model. The next section provides a brief explanation of the influencing factors identified by the authors.

Water depth increasing from zero (land wells) to more than 3000 meters (ultra-deepwater) governs the complexity and challenges faced in drilling [11]. Inherently deeper wells since targets are beneath columns of water resulting in longer strings and higher hook loads. Lower fracture gradients since seawater gradient (0.45 psi /ft) is less than overburden gradient (1 psi/ft), resulting in more casing strings [12]. Deepwater wells are also subject to shallow hazards such as shallow gas and shallow water flow which if encountered, may cause the well to be abandoned. These challenges are specific to deepwater and thus two main clusters may emerge: Deepwater in one cluster, and shallow water and land wells in another. Being that shallow water wells are mostly below 110 meters [13], water depth becomes a less governing factor.

Geological formation type is another major influencing factor. Formations can have high unconfined compressive strength (UCS) like certain carbonates [14], can be abrasive like some sandstones, or reactive like shales. Each pose their own issues that reflect in the WITSML data. Vibration may manifest in torque reading variations [15], while bit balling may manifest in low rate of penetration (ROP) and high weight on bit (WOB) [3], with longer and longer time between stands as more shale is drilled. A high UCS formation manifests as a sudden drop in ROP with the same amount of time between each stand.

The intention of a BHA is to make hole in an efficient manner. The main types of BHAs are rotary, motor, and RSS. A rotary BHA is a simple, cost effective BHA with limited directional capabilities. To make hole, WOB and RPM must be applied. A positive displacement motor with a bend generates RPM from flow rate [16], therefore it is possible to make hole without any surface RPM, commonly called sliding. Sliding is used with a bent motor to steer since surface RPM cannot be applied. Thus, a unique signature of a motor BHA is the presence of ROP and absence of RPM in WITSML data. An RSS is a rotary BHA with the ability to steer while maintaining RPM. When compared with wells drilled with mud motors, the RSS tool results in more efficient hole cleaning, reduced the risk of stuck pipe, and smoother weight transfer to bit [17]. A combination of RSS and zero bend motor can be applied to achieve higher RPM at bit while maintaining directional control and higher surface RPM.

The bit is the first point of contact with the formation. Bits come in different types such as PDC, diamond impregnated, mill tooth, and TCI. Commonly used in the industry are PDC bits that shear the formation upon contact, and mill tooth bits that crush the formation. A combination between the two bits, called hybrid bit combines elements of both and is used for interbedded formations. Pessier [18] has shown that PDC, Mill Tooth, and hybrid bits all have different Torque and WOB signatures for the same ROP. In wells with more than usual casing strings such as deep-water wells or HPHT wells, HEWD has emerged as a cost-effective way to drill and enlarge the hole in tandem. However, the main issue with HEWD is vibration, that can manifest in different ways on surface readings, but primarily through erratic torque [19].

After examining the intricacies and complexities of different well, BHA, bit and formation type signatures in surface readings, one can see the value that clustering through unsupervised machine learning can bring.

The final objective in this unsupervised learning work was to generate synthetic data that resembles real, measured drilling data as closely as possible. This has many applications, such as for well planning, addressing privacy concerns when sharing potentially sensitive data, or real-time operational benchmarking against expected conditions based on other known wells. An example of this is generating synthetic examples conditioned on some characteristics chosen by drilling engineers; this could be by explicitly conditioning on a subset of drilling parameters such as rotary speed, rate of penetration or torque, and then generating all of the remaining drilling parameters in a statistically

consistent, multi-variate, manner to match those conditions. Alternatively, if using algorithms involving learning of *latent representations* or *encodings,* where different types of wells or data scenarios are grouped and separable in a lower dimensional latent space, then data instances corresponding to particular types of wells can be generated by sampling from specific groups or clusters in the latent space. Clustering algorithms can be applied to these latent groupings to investigate their properties if pre-existing data labels are not available. By understanding the expected distributions of each drilling parameter under a given set of conditions, whether discovered by unsupervised learning or defined by engineers (or both), appropriate choices can be made for BHA or bit designs, or other drilling apparatus.

2. MATERIALS AND METHODS

The following surface measurement variables (with short forms) were used for the unsupervised learning applications described in this section:

- Mud Flow In (MFIA)
- Mud Density In (MDIA)
- Hookload (HKLA)
- Bit Depth (DBTM)
- Surface rotary torque (TQA)
- Rate-of-Penetration (ROPA)
- Surface rotary speed (RPMA)
- Standpipe pressure (SPPA)
- Weight-on-bit (WOBA)

These measurements were chosen based on their routine availability; the analytical techniques described in this work are not limited solely to these example (continuous) variables, or even tabular datasets with continuous variables in general. As only drilling was considered, bit depth and hole depth are equivalent here. A drilling dataset containing ~1.4 million observations from 28 independent wells was used for the unsupervised learning applications described in this work. A variety of well types were present in the data, including deep water, shallow water, and land-based wells, both with and without use of downhole mud motors. The wells were also geographically diverse, with data coming from Central and South America, West Africa, the Middle East and Southeast Asia.

2.1 Identifying patterns in data for labelling purposes

It is often difficult for humans to perceive and interpret data in more than three dimensions, so unsupervised techniques can augment human experts in these respects. Once these groups have been identified, they can act as labels on existing datasets, and applied to building classification models which can in turn be used for labelling new observations lacking labels.

The drilling data was grouped in nine dimensions using the Python implementation of HDBSCAN [20], a density-based hierarchical clustering method that offers superior performance to space-partitioning methods such as k-means. For visualization purposes, the data was projected into two dimensions using Principal Components Analysis, by retaining the first two principal components obtained for the sample dataset.

By investigating the characteristics of clusters such as drilling parameters, or by considering the cluster memberships using prior knowledge about the wells where sample data was taken from, the clusters corresponding to particular well types of interest can be determined. To illustrate this concept, the 28 wells were pre-grouped according to whether they were ultra-deep-water wells, shallow-water wells, or land-based wells. However, this knowledge was withheld from the clustering algorithm in order to test whether the groupings could be discovered independently via an unsupervised learning approach.

If a useful clustering can be found and labels added to the clusters after inspecting their properties, it is then relatively easy to sort new, unlabelled, data instances into the clusters. This can be done directly using HDBSCAN's "approximate_predict()" function, or by alternatively constructing a supervised multi-class classification model if some clusters need to be further consolidated into other groups revealed. The input data to the classifier is the same information used for unsupervised learning, while the target variable would be the previously inferred class label values.

2.2 Synthesizing drilling data

The classifiers described in the previous section for labelling data are an example of *discriminative* models; as their name suggests, these are used to discriminate between different types of data. Another group of ML models are *generative* models, which model the joint probability distribution of a dataset. This allows new "synthetic" instances to be drawn from the model distribution, such that their statistical properties are consistent with the original dataset. Well-known examples of generative ML models include Variational Autoencoders (VAEs) [21] and Generative Adversarial Networks (GANs) [22]. GANs are very useful when it is difficult to define an objective function for the generator. More comprehensive background information on generative models can be found in Chapter 20 of Ian Goodfellow's *Deep Learning* book [23].

In this work, VAEs were applied to generate synthetic drilling data, using the same dataset of surface measurements used for clustering in the previous section for training. Models were implemented using Keras, a Python library offering a high-level interface to Google's Tensorflow package for Deep Learning. VAEs consist of two components, an *encoder* and a *decoder*; the encoder network compresses a multidimensional input vector into a constrained, lower dimensional latent representation, while the decoder attempts to reconstruct the latent space values into a vector matching the input as closely as possible. When a useful representation is learned by the VAE, the positions of encoded points in the latent space should be meaningful in some way. A two-dimensional latent space was used for modelling drilling data. Architecturally, VAEs resemble conventional autoencoders, neural networks with a "bottleneck" structure for compressing an input vector into lower dimensions and reconstructing it to match the original input [23, Chapter 14]. However, VAEs learn a probability distribution for each of the

latent variables, rather than point estimates. This can be interpreted as assigning high probability mass to latent samples which could have generated the input vector, rather than a single point estimate for the most likely sample values. From these latent distributions, samples can be randomly drawn and then reconstructed into synthetic data examples.

To better understand the formulation of VAEs, we follow Kingma's and Welling's original VAE paper [21]. Consider a dataset $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ consisting of N instances $\boldsymbol{x}^{(i)}$. We assume there is an unobserved continuous random variable $\boldsymbol{z}^{(i)}$ which is involved in generating $\boldsymbol{x}^{(i)}$ from a random process parametrized by $\boldsymbol{\theta}$, where $\boldsymbol{z}^{(i)}$ is sampled from a prior distribution $p_\theta(\boldsymbol{z})$, and a conditional distribution (likelihood) $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ generates $\boldsymbol{x}^{(i)}$. Both distributions are assumed to be parametric, and that their probability density functions are differentiable with respect to both $\boldsymbol{\theta}$ and $\boldsymbol{z}$. In order to generate data, we need to understand the characteristics of $\boldsymbol{z}$ given the information $\boldsymbol{x}$ available; from Bayes Theorem, we have

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{z})p_\theta(\boldsymbol{x}|\boldsymbol{z})}{p_\theta(\boldsymbol{x})}\,, \tag{1}$$

where $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is the posterior density, and $p_\theta(\boldsymbol{x})$ is the marginal distribution of $\boldsymbol{x}$. However, solving for $p_\theta(\boldsymbol{x})$ is often intractable, so variational inference is required to estimate it. Hence, we approximate the true posterior density $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ using another distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with parameters $\boldsymbol{\phi}$, which is defined to be a multivariate Gaussian with a diagonal covariance structure for tractability. $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ can be represented by a neural network (the encoder) which outputs a distribution of $\boldsymbol{z}$ values which could have generated $\boldsymbol{x}$, and similarly $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is also represented by another neural network which outputs $\boldsymbol{x}$ which could have been generated from $\boldsymbol{z}$. However, the need to randomly sample values from the latent space poses a problem for fitting the parameters of these neural networks via back-propagation updates. This is solved using the "reparameterization trick", for Gaussian distributed latent variables, we write $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma^2} \odot \boldsymbol{\varepsilon}$, where $\boldsymbol{\mu}$ is the vector of means and $\boldsymbol{\sigma^2}$ is a vector of variances characterizing the latent distributions, and $\boldsymbol{\varepsilon}$ contains elements randomly sampled from a unit Gaussian $\mathrm{N}(0,1)$ and is element-wise multiplied with $\boldsymbol{\sigma^2}$. The encoder network thus outputs two vectors with dimensions set by the number of latent variables, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma^2}$, which can be used as part of an optimization objective with respect to the neural network parameters, while enabling random samples to be drawn from the latent distributions as inputs to the decoder. A simplified schematic of a VAE summarizing the above formulation is given in Figure 1.

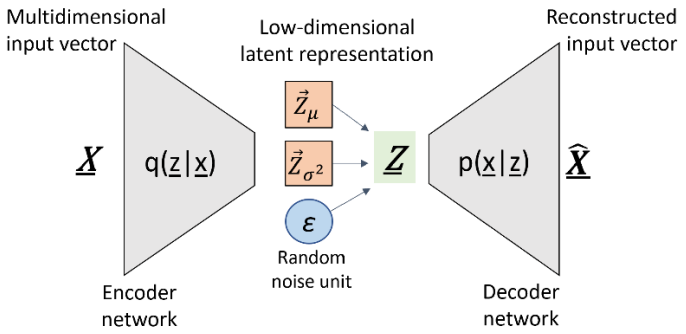

Figure 1: SIMPLIFIED SCHEMATIC OF A VARIATIONAL AUTOENCODER. IN THIS WORK, A TWO-DIMENSIONAL LATENT SPACE WAS USED, HENCE Z IS A VECTOR WITH TWO COMPONENTS.

VAEs are regularized through enforcing conditions on the distributions of the latent variables, typically by requiring them to resemble unit Gaussian distributions as priors. This is achieved by modifying the autoencoder's loss function to include a Kullback-Leibler (KL) Divergence term, which quantifies the similarity between two probability distributions. For distributions of continuous random variables P and Q, the KL-divergence is given by

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx\,, \tag{2}$$

where $p(x)$ and $q(x)$ are the probability densities of P and Q. For a VAE, P and Q are the prior distribution $p_\theta(\boldsymbol{z})$ and the approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. For unit Gaussian priors and assuming that covariance matrix for the latent variables has non-zero values only on the diagonal elements, the KL-divergence for a single sample can be written as

$$D_{KL} = -\frac{1}{2}\left(\sum_{i=1}^{n_z}(1 + \log(\sigma_i^2)) - \sum_{i=1}^{n_z}\mu_i^2 - \sum_{i=1}^{n_z}\sigma_i^2\right), \tag{3}$$

where the summations are done over the latent dimensions and symbols have their usual meanings. This simplifying assumption encourages VAEs to learn disentangled latent representations when the KL-divergence term is more strongly weighted [24], which can be a desirable property.

The KL-divergence contribution to the loss function encourages smoothness in the latent representations obtained from training, such that points sampled from a particular region of the latent space should yield decoded values which are fairly close to each other, a desirable characteristic for generative models. The VAE loss is therefore given in by

$$\mathcal{L}_{VAE} = D_{KL} + \beta\mathcal{L}_{recon}\,, \tag{4}$$

where $\mathcal{L}_{recon}$ is the reconstruction loss and β is a tuneable hyperparameter. Commonly used choices for reconstruction loss

are the familiar cross-entropy or mean squared error (MSE); cross-entropy was used in this work, as data was scaled into the 0.0-1.0 range for all variables using min-max scaling. Note that some formulations of the VAE loss in the literature may apply β to the KL-divergence term instead (for example Higgins et al. [24]), however equivalent effects can be realised through appropriate choices of β. A value of 7.0 was used for β in the VAE used for generating synthetic drilling data in this work.

2.3 Well Planning Applications

Statistical distributions for each drilling parameter are summarized based on well type. A recommendation based on the range and statistical behaviour of the synthesized data for each drilling parameter is made at the well design stage to choose the optimum equipment rating to cater for the future well and minimize overdesigning of equipment. Since the expected rotary speed and weight on bit are known ahead of time from the synthesized data, optimum BHA and bit design can be chosen in the conceptual planning stage of the well. Flow rate, mud weight, and standpipe pressure will aid in the optimum selection of mud pumps for the rig, minimizing emissions. Surface torque will define the drill pipe type needed for the planned well, and mud weight will aid in casing design and setting depth. These parameters are usually defined during the detailed design stage after models have been built and run. However, with synthesized data, these parameters are readily available whenever the drilling engineer starts the planning stage. The synthesized data also aids in defining technical specifications when contracting for future well operations.

3. RESULTS AND DISCUSSION

3.1 Clustering and labelling

A drilling dataset clustered using all 9 continuous variables is shown in Figure 2, using HDBSCAN as the clustering algorithm and visualized by via a scatter plot of the first two principal components. Deepwater and ultra-deepwater wells are clearly separated from the shallow water and land wells using downhole motors. Analysis of the cluster membership of the blue cluster revealed it was made up completely by a single deep water well (albeit the shallowest of the deep-water wells), while the orange cluster contained a combination of the two ultra-deepwater wells from West Africa, and a deep water well in Southeast Asia. The pink, red and green clusters all contained drilling observations from shallow water wells, while the purple cluster was dominated by samples from land-based wells where downhole mud motors were used.

A multi-class classifier was constructed based on the clustering shown in Figure 2, using the XGBoost Python package [25]. The pink, red, brown, and green clusters, which all contained observations from shallow water wells, were aggregated together for the purpose of this exercise. The "noise" cluster (samples not allocated to a cluster) was also used as an "unclear" class label. 50,000 samples were used to train the classifier, and ~20,000 samples were used for validation. The class balances in the validation dataset are described in Table 1.

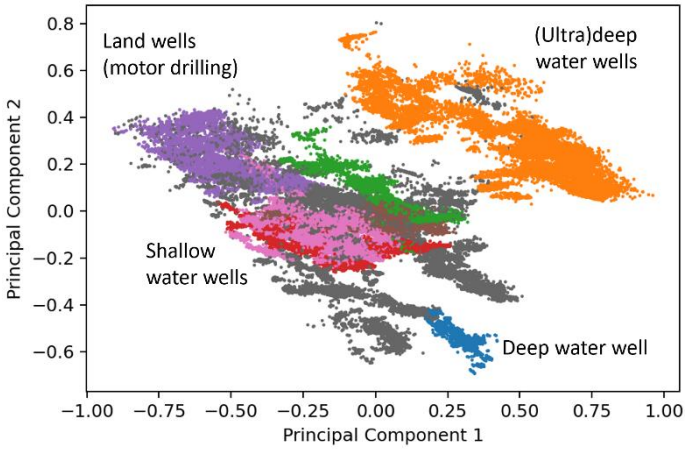

Figure 2: RESULTS FROM CLUSTERING DRILLING DATA IN 9 DIMENSIONS USING HDBSCAN AND VISUALIZED USING THE FIRST TWO PRINCIPAL COMPONENTS. DARK GREY POINTS HAVE NOT BEEN ALLOCATED TO A CLUSTER.

Class	Clusters in Fig. 3	% of samples
Shallow water	Pink, red, green, brown	32.4
Land	Purple	9.6
Ultra-deep water	Orange	21.5
Deep water	Blue	9.7
Unclear	Dark grey	26.8

Table 1: CLASS INFORMATION FOR THE DATASET USED TO VALIDATE THE TOY MULTI-CLASS CLASSIFIER USED FOR LABELLING DATA BASED ON TYPE OF WELL.

A classification accuracy (the proportion of examples correctly labelled) of 0.971 was achieved on the toy validation dataset, without performing hyperparameter optimization. This demonstrates that once the useful groupings have been identified via Unsupervised Learning, these groupings can be used for labelling new observations with high accuracy, unlocking additional value from oil companies' datasets.

3.2 Synthetic drilling data

In order to generate useful synthetic data, a VAE must be able to map the original data into some meaningful latent representation using the encoder, and then reconstruct, from points in the latent space, values of the input variables resembling the original inputs as closely as possible. A method for testing whether a meaningful latent representation has been obtained is to check how observations from pre-selected groups are distributed once encoded. Here, observations were grouped based on whether they came from deep water (DW), shallow water (SW) or land-based wells. The encoded observations are visualized as points in a scatter plot (Figure 3), where some separation between the different types of wells can be observed, with a smooth progression from land wells to shallow water and

to deep water. There is a clear difference between deepwater and the other well types. This can be attributed to shallow water being up to 100m of water, whereas deepwater can be between 500 and 3000 m.

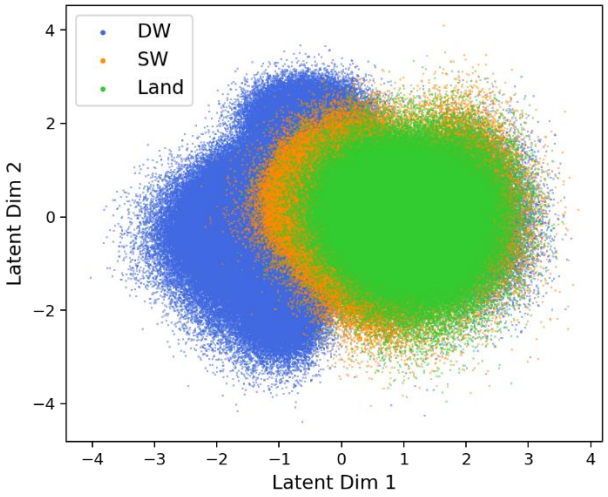

Figure 3: SCATTER PLOT OF DRILLING DATA COMPRESSED INTO THE 2-D LATENT SPACE OF A VARIATIONAL AUTOENCODER, WITH B=7.0.

The influence of the KL-divergence term in the VAE's loss function on the smoothness of the latent space distributions can be seen by training multiple VAEs with various weighting factors β applied to the reconstruction error term. Otherwise, the VAE models were identical and used the same training datasets. For low β values, the KL-divergence is increasingly significant, resulting in latent space distributions more closely resembling 2D Gaussians, as seen in Figure 4. The plotted samples have been coloured according to the type of well they came from, with the same colour scheme as in Figure 3 (blue points are from deep water wells, orange points from shallow water wells, and green points from land-based wells using downhole mud motors). As β increases and reconstruction error becomes more important, more complicated structures begin to emerge in the latent space, with reduced smoothness and greater separation between samples from different well types. For β values of 6.0 and 8.0, the ultra-deepwater wells (four different wells, 2 of them from the same field) appear to separate into 3 distinct clusters.

The authors chose to highlight differences based on water depth for this study, due to the ease in separating wells in this manner to use for reference; however, other classifications can be used such as BHA type, inclination, or formation hardness.

A useful quality check to apply to synthetic data is to consider the distributions of each variable in the synthetic data; these should be similar those of the original dataset, as the goal with synthetic data is for it to be indistinguishable from the corresponding real data. In Figure 5, comparisons between the real and synthetic data are shown for each of the 9 variables obtained from the surface measurements. The synthetic data shown in the distributions was generated based on randomly sampling 250,000 points from a 2D Gaussian distribution in the latent space, centered on (0,0) and with standard deviations of (1.1, 1.0), then using these as inputs to the VAE's decoder, before rescaling the generated examples back to their original units.

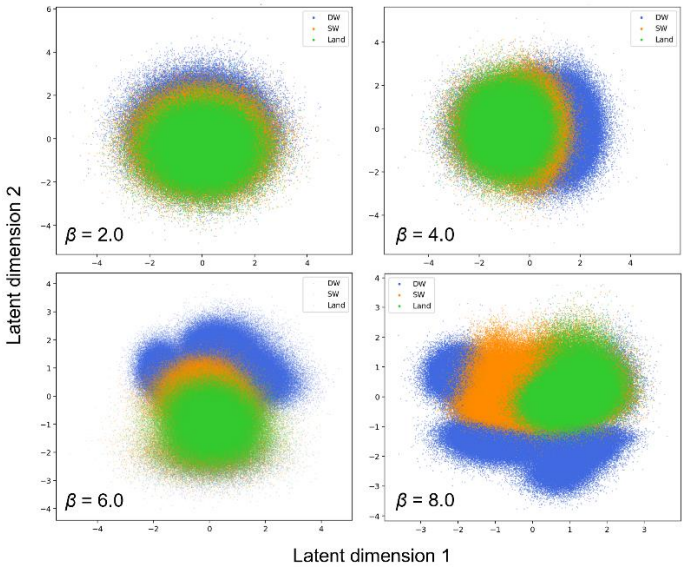

Figure 4: VISUALIZATION OF OBSERVATIONS IN THE 2D LATENT SPACE FOR VARIATIONAL AUTOENCODERS TRAINED WITH DIFFERENT β WEIGHTING FACTORS APPLIED.

While the synthetic distributions do not perfectly match the original data, most of the features of the original distributions are captured. This is a promising result, considering most of the original distributions are multimodal, skewed, or heavy-tailed – far from Normal distributions. The synthetic Mud Density In, Hookload and Bit Depth distributions capture multiple modes from the original distributions, including some of the heavy tails. For Mud Flow In, the primary mode near ~850 gpm was correctly captured by the generative model, however the secondary mode near ~600 gpm was not. A similar observation can be made for the top-drive rotary speed distributions, where the synthetic distribution has a single primary mode and long heavy tail. The Weight-on-Bit distribution features (skew, primary mode) are mostly reproduced, though the extent of the long tail does not fully match between the synthetic and real data distributions.

The synthetic data is also "smoother" than the real data, providing another potential application of VAE's. When rig data feed is run through the function, the output is not only an automatic classification of the well from real-time data, but also a smoother data distribution with less "noise". The drilling engineer can then work with filtered data closely resembling the real data with fewer outliers or discontinuities in the data.

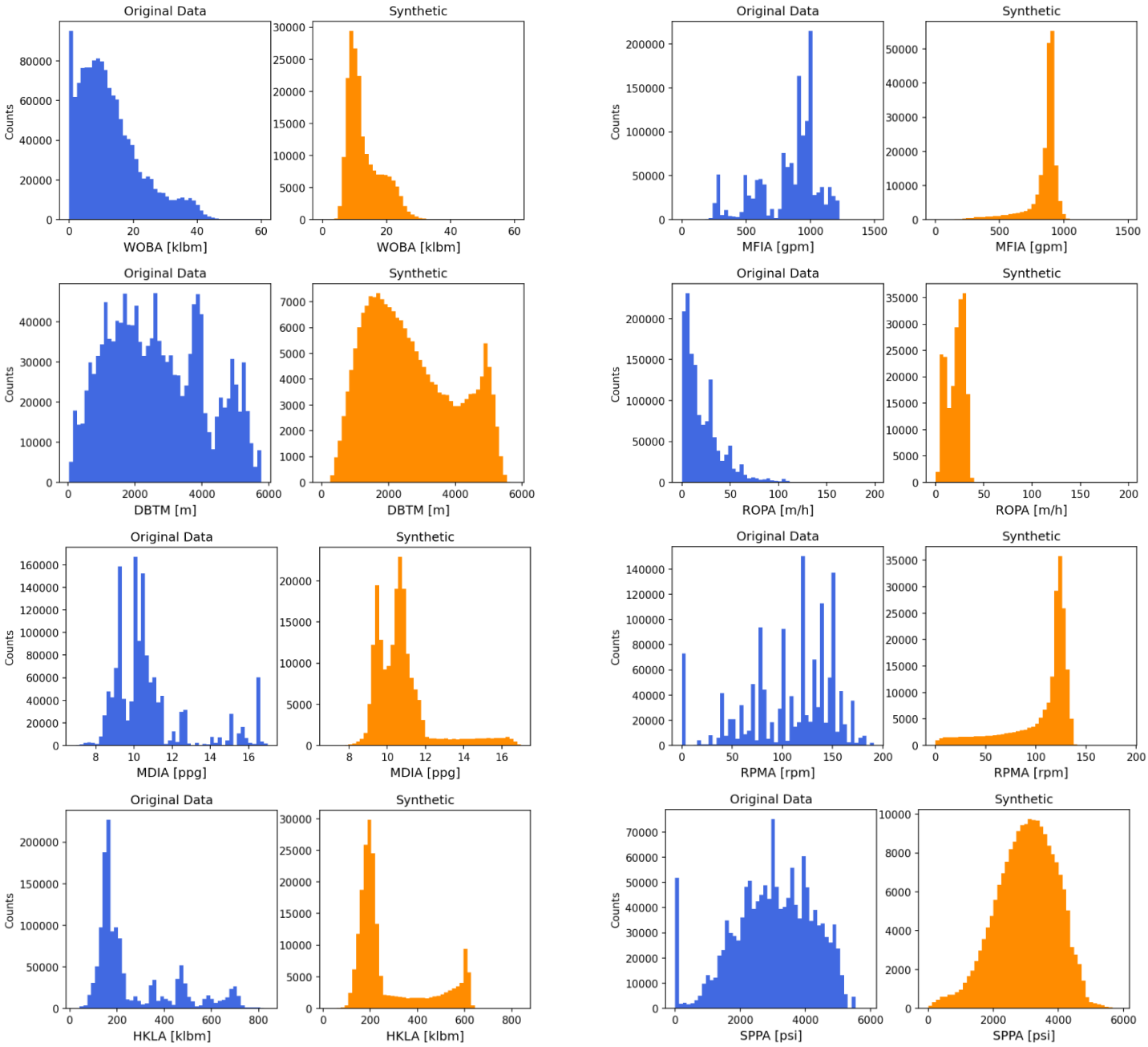

Figure 5: COMPARISON OF DISTRIBUTION OF EACH DRILLING PARAMETER BETWEEN THE SYNTHETIC DATA SET AND THE ORIGINAL DATA SET.

It is instructive to investigate the characteristics of data sampled from different regions of the latent space (Figure 3), to see how the distributions of the generated data vary by moving around the latent representations. To illustrate this, data was sampled from Gaussian distributions, with standard deviations of 0.2, and centered on points in a 5x5 grid. 10000 samples were taken from around each grid point. The grid's center point was at (0, 0), with corners located at (-2.5, 2.0), (-2.5, -2.0), (2.5, 2.0), (2.5, -2.0), and regularly spaced points between these. In Figure

6, the bit depth distributions generated from sampling around each grid point are shown. The layout of the plots matches the latent space grid points. The bit depth distributions were observed to smoothly change in most cases when moving around the grid, a desirable property for generative models, encouraged by the KL-divergence term in the VAE objective function. Furthermore, the distributions in the left-hand side of the image represent samples with on larger bit depths compared to those on the right, consistent with the separation between ultra-deep water and land-based wells seen in Figure 3.

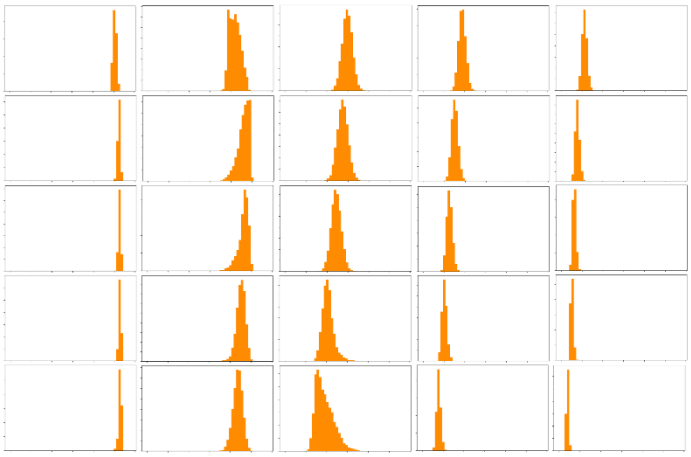

Figure 6: DISTRIBUTIONS OF BIT DEPTH FOR EXAMPLES GENERATED BY RANDOMLY SAMPLING FROM GAUSSIAN DISTRIBUTIONS CENTERED ON DIFFERENT POINTS IN A GRID IN THE LATENT SPACE.

In addition to each variable having approximately the same distribution in the synthetic dataset as in the original data, correlations between variables should also be preserved for a good generative model. This can be evaluated via a comparison of correlation heatmaps produced from both the original and synthetic datasets (same as used in Figure 5), shown in Figure 7. By inspecting the correlation heatmaps, we can see the (Pearson) correlation structure is preserved quite well in the synthetic data, particularly in terms of the signs of the correlations, although there are some minor differences in magnitudes.

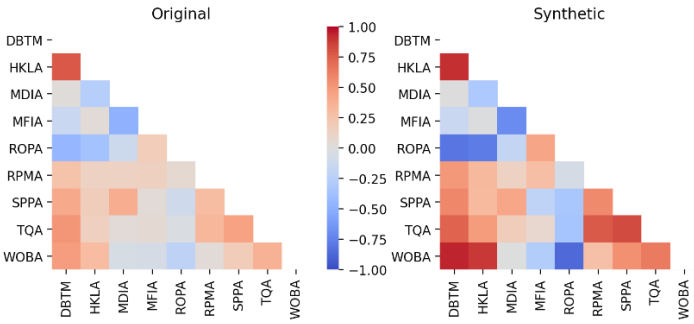

Figure 7: CORRELATION HEATMAPS FOR THE ORIGINAL DATA AND SYNTHETIC DATA.

Another method for assessing a generative model is to train supervised binary classification models attempting to distinguish between real and synthetic data. The optimal case is hence when the classifier is unable to distinguish between the two classes of data; for a balanced dataset, the prediction accuracy would tend towards 0.5, equivalent to a coin toss. This was done using a few different classifiers provided by the scikit-learn Python package [26], namely logistic regression, a Linear Support Vector Classifier (SVC) and a Random Forest Classifier (RFC) using 100 trees and a few different max tree depths, which controls the complexity of individual decision trees in the ensemble. 250,000 samples were generated by the VAE, and 250,000 additional samples were taken randomly from the original drilling dataset (~1.4 million observations), such that the synthetic and real data were equally represented in the dataset. K-fold cross-validation, with K=5, was used to estimate the classification accuracy for the candidate classifiers, with the results summarized in Table 2. A few different max tree depths were tested for the random forest classifier (RFC) to see the effect of model complexity on classification performance. From this, it can be seen that the VAE could successfully fool the logistic regression and Linear SVC models in most cases, which struggle to distinguish between real and synthetic data based on their accuracy scores of ~0.49. However, the random forest classifiers were able to identify differing characteristics between the real and synthetic examples and could correctly identify the samples' origins in most cases. This signifies that there is room for improvement in generative model performance.

Classifier	Mean Accuracy	Std. Dev. Accuracy
Logistic regression	0.492	0.002
Linear SVC	0.491	0.002
RFC (max depth=10)	0.992	0.001
RFC (max depth=3)	0.849	0.007
RFC (max depth=2)	0.820	0.002

Table 2: CLASSIFICATION ACCURACY METRICS FOR A SELECTION OF CLASSIFIERS ATTEMPTING TO DISTINGUISH SYNTHETIC DATA FROM REAL EXAMPLES.

3.3 Synthetic data use cases

As the VAE jointly models the variables contained in the drilling data, realistic relationships between the generated drilling parameters are preserved and statistically consistent. Hence, it is possible to place conditions on the synthetic data, such that generated examples match some desired characteristics. One option is to do this indirectly, by sampling in a specific region of the latent space, for example around the green cluster in Figure 3 if the user wants data closely resembling land-based wells with motor drilling. This requires the latent distributions to have been well-characterized. Another option is to place conditions on specific drilling parameters; if the drilling engineer would like to understand what parameter combinations could be associated with ROP values falling into a specified

range, then expected distributions for these can be obtained by generating synthetic drilling data matching the conditions.

This can be done via rejection sampling approaches; by randomly sampling observations from the latent space and reconstructing them, until sufficient examples matching the specified conditions are found (using an appropriate stop condition on the maximum number of iterations. This provides engineers with a view of what sorts of drilling parameters are required for a certain well type or to match desired drilling parameter values and helps them understand what combinations of variables could be expected in given scenarios, which in turn informs planning decisions when preparing a drilling campaign. Drilling engineers can also use the sampled data to benchmark ROP in real-time, or post-drilling against similar operations.

To illustrate the concept of sampling from a specific region of the VAE latent space for well planning purposes, the authors generated a dataset for deep water wells sampling from Gaussian distributions centered on points (-1.0, -0.1) in Figure 3. The generated data is dependent on where in the latent space samples are drawn from prior to reconstruction via the decoder. The locations and extents of the sampling distributions were not tuned, which may have contributed to discrepancies observed between the real and synthetic data in this context. While future work will focus on sampling methods from the latent space to optimize well and drilling parameter representation, the section bellow illustrates how unsupervised machine learning can be a useful tool in well planning. Summary statistics for each drilling parameter for deepwater wells is presented in Table Table **3**.

In the early stages of designing a deepwater well, the drilling engineer would extract a synthetic summary like the one presented in Table 3. Looking at the RPM statistics from Table 3, a 5th percentile of 91 RPM and a 95th percentile of 134 RPM results in a recommendation for a rotary BHA rather than a PDM (due to a lower surface RPM limit for bent motors). A WOB median of 18.6 suggests that the BHA, stabilization, and HEWD system needs to cater for a higher WOB and therefore the neutral point between tension and compression in the drillstring needs to be accounted for. Currently, the drilling engineer would have to manually sift through pages or spreadsheets of individual offset wells to obtain an idea of the parameters used. Synthetic data offers an exponentially quicker solution that samples large amounts of data points within seconds.

However, the novel well planning method proposed is not without its shortcomings. Table 3 shows that the synthetic set of data values consistently lies within the real set of data values. This can create a blind spot for the engineer designing the well. For example, in something as critical as mud weight to control the well, 95th percentile synthetic data is 15% lower than the same percentile taken from real data. It is also important to note that the sampled drilling parameters are highly dependent on the input well data. In this case, all deepwater wells are vertical wells, which would exhibit lower surface torque than deviated wells. Therefore, the bias of the synthetic data to its input data needs to be considered. Future work will focus on generating depth-based synthetic data for a clearer idea of the drilling parameters by hole section. Other classification methods such as deviated vs vertical trajectories, and formation type, can also be considered

Using synthetic data in well planning provides the drilling engineer with large and immediate data visibility compared to conventional methods of data gathering. However, this method should still be combined with traditional well design approaches to account for all scenarios encountered during drilling operations.

Variable	Median		95th percentile		5th percentile	
	Synth.	Real	Synth.	Real	Synth.	Real
RPM	125	110	134	161	91	21
WOBA [klbm]	18.6	19.1	24.7	38.7	10.6	2.5
TQA [kft lbf]	12.5	12.2	17.1	24.9	7.2	1.8
MFIA [gpm]	904	901	966	1092	589	276
MDIA [ppg]	9.7	9.4	14.0	16.5	9.2	8.5
SPPA [psi]	3690	3790	4620	5192	1104	27

Table 3: COMPARISON OF SYNTHETIC DATA TO REAL DATA - DEEPWATER WELLS.

4. CONCLUSION

Several promising use cases of unsupervised learning applied to drilling data and well planning were presented in this work, including automated discovery of interesting groupings in multivariate drilling data using clustering methods, labelling of data instances using these groupings, and generation of synthetic drilling data using Variational Autoencoders (VAEs), which can also identify useful latent representations of drilling datasets. The synthetic data was shown to be difficult to distinguish from real data, showing promising use of machine learning in well planning applications.

ACKNOWLEDGEMENTS

The authors would like to thank PETRONAS for the support, manpower and resources allocated for this project. As well as Exebenus colleagues Dalila Gomes, Jan Kåre Igland, and Olav Revheim for interesting discussions on this topic.

REFERENCES

[1] Pisauri, F., Escudero, F., and Rodriguez, P. E., 2020, "Enabling WITSML Stream Analytics Through Open Source Big Data Technologies,", *Proceedings of SPE Latin American and Caribbean Petroleum Engineering Conference*, Virtual, 27 – 31 July, Paper No. SPE-198963-MS.

[2] Noshi, C. I., and Schubert, J. J., 2018, "The Role of Machine Learning in Drilling Operations; A Review," Proceedings of the SPE Eastern Regional Meeting, Pittsburgh, Pennsylvania, USA, 7 – 11 October, Paper No. SPE-198963-MS.

[3] Batruny, P., Zubir, H., Slagel, P., Yahya, H., Zakaria, Z., and Ahmad, A., 2021, "Drilling in the Digital Age: Machine Learning Assisted Bit Selection and Optimization," Proceedings of the International Petroleum Technology Conference, Virtual, 23 March – 1 April, Paper No. IPTC-21299-MS.

[4] Fadokun, D. O., Oshilike, I. B., and Onyekonwu, M. O., 2020, "Supervised and Unsupervised Machine Learning

Approach in Facies Prediction," Proceedings of the Nigeria Annual International Conference and Exhibition, Lagos, Nigeria, 11 – 13 August, Paper No. SPE-203726-MS.
[5] Zhao*, T., Verma, S., Qi, J., and Marfurt, K. J., 2015, "Supervised and Unsupervised Learning: How Machines Can Assist Quantitative Seismic Interpretation," Proceedings of the SEG Technical Program Expanded Abstracts, New Orleans, Louisiana, USA, pp. 1734–1738, Paper No. SEG-2015-5924540.
[6] Meor Hashim, M. M. H., Yusoff, M. H., Arrifin, M. F., Mohamad, A., Gomes, D., Jose, M., and Ezharuddin Tengku Bidin, T., 2021. "Utilizing Artificial Neural Network for Real-Time Prediction of Differential Sticking Symptoms". Proceedings of the IPTC International Petroleum Technology Conference 2021, Virtual, 23 March – 1 April 2021, Paper No. IPTC-21221-MS.
[7] Meor Hashim, M. M. H. et al., 2021. "Case Studies for the Successful Deployment of Wells Augmented Stuck Pipe Indicator in Wells Real Time Centre". Proceedings of the IPTC International Petroleum Technology Conference 2021, Virtual, 23 March – 1 April 2021, Paper No. IPTC-21199-MS.
[8] Bandura, L., Halpert, A. D., and Zhang, Z., 2018, "Machine Learning in the Interpreter's Toolbox: Unsupervised, Supervised, and Deep-Learning Applications," SEG Technical Program Expanded Abstracts, Anaheim, California, Paper No. SEG-2018-2997015.
[9] Tse, K. C., Chiu, H.-C., Tsang, M.-Y., Li, Y., and Lam, E. Y., 2019, "Unsupervised Learning on Scientific Ocean Drilling Datasets from the South China Sea," Front. Earth Sci., 13 (1), pp. 180–190.
[10] Singh, H., Seol, Y., and Myshakin, E. M., 2020, "Automated Well-Log Processing and Lithology Classification by Identifying Optimal Features Through Unsupervised and Supervised Machine-Learning Algorithms," SPE Journal, 25(05), pp. 2778–2800.
[11] Watson, P. A., Iyoho, A. W., Meize, R. A., and Kunning, J. R., 2005, "Management Issues And Technical Experience In Deepwater And Ultra-Deepwater Drilling," Proceedings of the Offshore Technology Conference, Houston, Texas, USA, 2 – 5 May, Paper No. OTC-17119.
[12] Christman, S. A., 1973, "Offshore Fracture Gradients," Journal of Petroleum Technology, 25(08), pp. 910–914.
[13] Giblon, R. P., and Minorsky, V. U., 1975, "Evolution of the Design of a Jack-up Type Offshore Structure," Marine Technology and SNAME News, 12(01), pp. 50–59.
[14] Fiduk, J. C., 2010, "Analysis of Layered Evaporites Within the Santos Basin, Brazil," Proceedings of the Offshore Technology Conference, Houston, Texas, USA, 3 – 6 May, Paper No. OTC-20938-MS.
[15] Davis, J. E., Smyth, G. F., Bolivar, N., and Pastusek, P. E., 2012, "Eliminating Stick-Slip by Managing Bit Depth of Cut and Minimizing Variable Torque in the Drillstring," Proceedings of the SPE/IADC Drilling Conference and Exhibition, San Diego, California, USA, 6 – 8 March, Paper No. SPE-151133-MS.
[16] Li, J., Tudor, R., Sonego, G., and Varcoe, B., 1999, "Performance of Positive Displacement Downhole Motors under Two-Phase Flow," *Journal of Canadian Petroleum Technology*, 38, pp. 7.
[17] Tribe, I. R., Burns, L., Howell, P. D., and Dickson, R., 2003, "Precise Well Placement With Rotary Steerable Systems and Logging-While-Drilling Measurements," *SPE Drill & Compl* 18(01), pp. 42–49, Paper No. SPE-82361-PA.
[18] Pessier, R., Damschen, M., and Hughes, B., 2011, "Hybrid Bits Offer Distinct Advantages in Selected Roller-Cone and PDC-Bit Applications," *SPE Drill & Compl* 26(01), p. 8.
[19] Batruny, P., Aburto, Z., Slagel, P., Paimin, M. R., Mahran, M., and Cortina, C. J., 2021, "Drilling in the Digital Age: Sensors in Bit and Underreamer Improves Future BHA Design Offshore Mexico," Proceedings of the Abu Dhabi International Petroleum Exhibition and Conference, Abu Dhabi, UAE, 15 – 19 November, Paper No. SPE-207398-MS.
[20] McInnes, L., Healy, J., Astels, S., 2017, "hdbscan: Hierarchical density based clustering", *Journal of Open Source Software*, The Open Journal, 2 (11).
[21] Kingma, D.P. and Welling, M., 2013. Auto-encoding Variational Bayes. arXiv preprint arXiv:1312.6114.
[22] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014, "Generative Adversarial Nets", *Advances in Neural Information Processing Systems*, pp. 2672-2680.
[23] Goodfellow, I., Bengio, Y., and Courville, A. 2016. *Deep Learning*, MIT Press.
[24] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S. and Lerchner, A., 2017, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework", Proceedings of International Conference on Learning Representations, Toulon, France, 24 – 26 April.
[25] Chen, T. and Guestrin, C., 2016. "XGBoost: A Scalable Tree Boosting System", Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, DOI: 10.1145/2939672.2939785
[26] Pedregosa, F., Varoquaux, G., Gramfort, A. et al, 2011, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research* **12**, pp. 2825-2830.