# Incorporating Plain English Driller Comments Into Machine Learning Drilling Optimization

Dang TON, Roland HORNE

Stanford University

dangton@stanford.edu, horne@stanford.edu

**Keywords:** drilling optimization, machine learning, neural network, natural language processing

## ABSTRACT

Drilling optimization by use of data from past drilling records is a challenging task. A sufficient amount of data from multiple sources of information is required, and the data must be standardized and properly recorded. Unfortunately, such requirements can be difficult to meet in the case of geothermal wells. This study has been an attempt to use modern data science techniques to create machine learning models that predict rate-of-penetration from past drilling records. It was necessary to overcome the limitations commonly encountered in geothermal drilling records. In addition, rate-of-penetration is not the only criterion used in drilling optimization, so other optimization objectives were also considered. In this study, machine learning models that can help to reduce nonproductive time by predicting potential tripping/problems were developed. These models were found to have satisfactory prediction accuracy to be useful in real life situations. This study also investigated the problem of incorporating "plain English" textual-type entries in drilling records into machine learning models. These textual-type entries often carry useful information about the condition of the well that may not be available elsewhere in the data records. However, the plain English remarks are recorded in nonstandardized ways, and vary greatly depending on who is making the notes. This makes the task of incorporating textual-type data into any machine learning model an expensive and time-consuming operation. We found that Bidirectional Encoder Representations from Transformers (BERT) can provide a solution for incorporating textual data into machine learning models effectively. Using textual information, together with the standardized drilling records, was found to improve the quality of predictions in most cases.

## 1. INTRODUCTION

Drilling operations use highly intricate equipment in a very uncertain and hostile downhole environment, while needing to keep the costs to a minimum. Due to the high cost of renting and operating a drilling rig, reducing the total drilling time always has high priority when talking about drilling optimization. Typical optimizations include increasing rate-of-penetration (ROP), and reducing nonproductive times (due to unscheduled tripping and problems, etc.) However, there is no clear relationship between the hundreds of parameters sent from the rigs and the optimizing parameters. Compounded with the complexities in downhole lithology and wellbore, drilling optimizations can be unreliable and depend strongly on who is making the prediction.

Machine learning (ML) offers a promising solution to the intricate problem of drilling optimization. Modern ML methods like artificial neural network (ANN) not only can accurately describe complex relationship between rig data and optimizing parameters, but can also process nonnumeric data like written text, sound or image recordings. Coupled with the ability to process inputs with missing data, modern ML offers a new perspective on drilling optimization where classical statistical methods have failed before. With modern ML, the target of autonomous drilling in complex formation with minimal prior knowledge may become a reality in the future.

A collection of drilling records from different geothermal field in the United States and Iceland, each with different geological properties, was used in this study. A preliminary analysis of the collection of drilling records is the main topic of Section 2. Section 3 will discuss about the deep neural network model developed to predict rate-of-penetration, and potential future tripping/problems. The problem of integrating text data into the predictions is also in the scope of this study, and will be discussed in Section 4.

## 2. DATASET

The dataset used in this study is a part of the EDGE project (Carbonari, et al., 2021), which is a research project aimed at developing machine learning strategies in geothermal drilling optimization under the support of the EDGE Program of the US Department of Energy (DOE) Geothermal Technologies Office. The EDGE project aims to build a database of geothermal drilling data and from there develop optimization schemes based on machine learning and deep learning methodologies. Currently, the dataset includes data from 113 wells from different geothermal projects developed over the past 30 years. These data represent vastly different geologic and operational settings.

For each well, records are collected, averaged daily, and tallied by record number. All collected records are stored in a relational database for the ease of query and modification. Each relational table in the database is related to other tables by using WellID and Record Number as keys. There are a total of 63 tables, each corresponding to a different source of information. Sources of information include drilling rigs, drill string, drill bit, mud logging, etc. The high variety in information sources is very beneficial to the process of ROP optimization as modeling the subsurface environment accurately requires detailed surface measurements but also detailed wellbore measurements.

### 2.1 Exploratory data analysis

By selecting frequently collected features from the database, it is possible to do exploratory data analysis (EDA) on the collected data to make assessments about their quality and find relationships between variables. Because naturally occurring variables will have the tendency to be distributed under some common probability distributions (Gaussian, Exponential, etc.), any feature that is unusually distributed is worth examining.

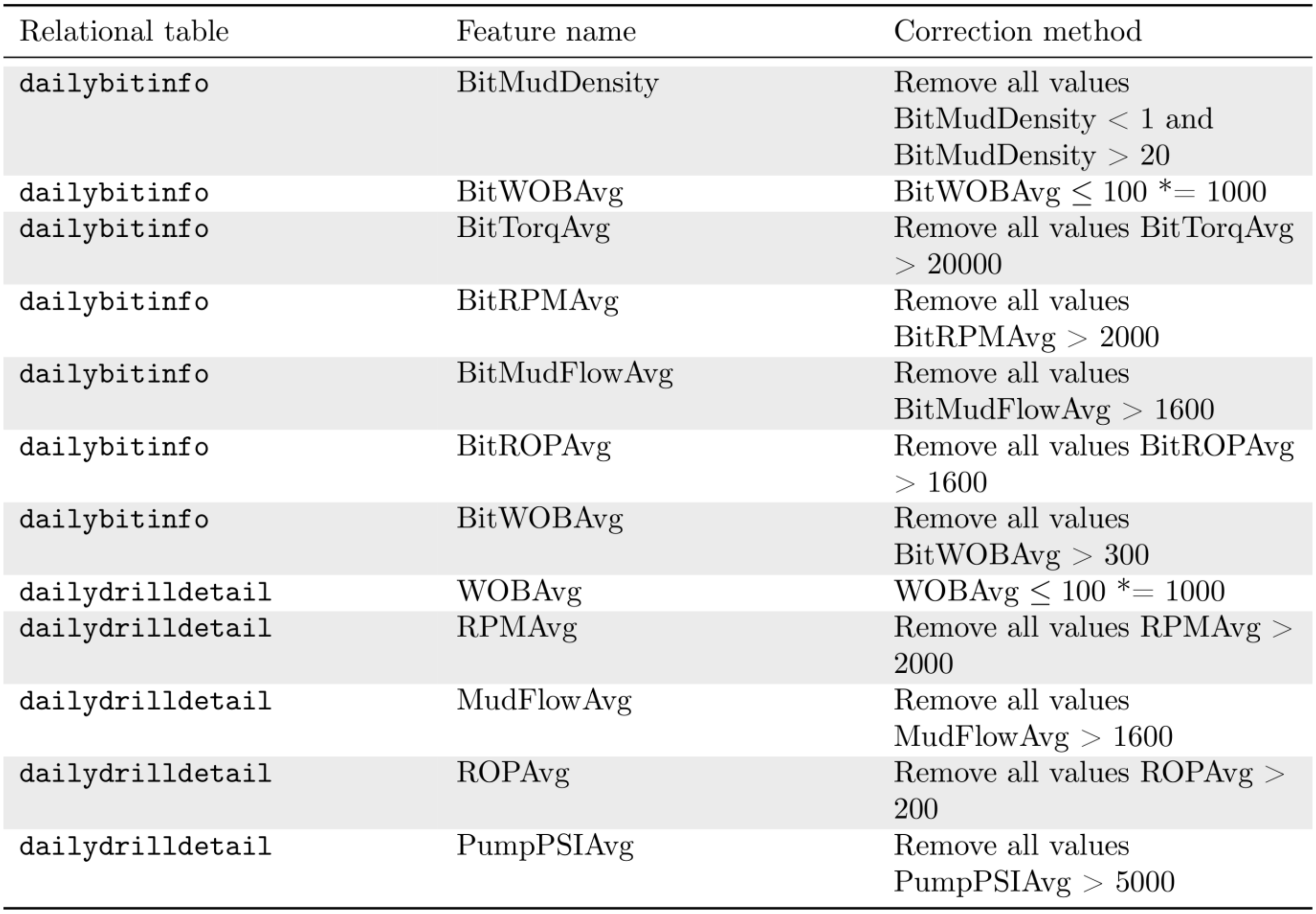
It is easy to see from Figure 1 and Figure 2 that while some features exhibit common distributions (MudFlowMax, AnnVelocityDC, AnnVelocityDP, BitHrs), some features do have a spike in the histogram around the zero value. This is an unexpected behavior, which indicates these features may have outliers or wrong units. Another unexpected problem is that some features have negative values when they should be positive (BitFootage, BitROPAvg).

Histograms of some example features that exhibit spikes around the zero value are in Figure 3 and Figure 4. It can be seen that while most recorded values of BitWOBAvg and BitTorqAvg data are in the [0-1000] range, some values go up to 35000, or about three orders of magnitude difference compared to typical values. A theory for this behavior is that the different drillers mixed up the units when recording the values: some recorded BitWOBAvg as lbs while some as klbs, etc. An easy fix to this problem is noticing the original units for BitWOBAvg and BitTorqAvg are lbs and lbf-ft. Therefore, any value under 100 is incorrectly recorded and should be multiplied by 1000.

The results of the proposed correction method are in Figure 5 and Figure 6. It is observed that in Figure 5, the proposed correction method turns the BitWOBAvg distribution into a Gaussian distribution, which indicates that our theory likely is correct. However, the opposite happened in Figure 6, where the proposed correction method does not significantly change the distribution shape, which may indicate that our theory is not applicable for BitTorqAvg. In addition, because torque values in drilling can vary significantly when drilling when factoring in stuck pipe, lost circulation and mud motor, BitTorqAvg should be left as is. Another problem with BitTorqAvg is that some extreme values (20000, 35000, etc.) only appear a few times in the entire dataset. These datapoints can be considered extreme outliers and removed.

By following the same procedures as BitWOBAvg and BitTorqAvg, improper and extreme outlier values can be identified and fixed/filtered out as seen in Table1.

**Table 1: Correction methods**



| Relational table | Feature name | Correction method |
|---|---|---|
| `dailybitinfo` | BitMudDensity | Remove all values BitMudDensity $< 1$ and BitMudDensity $> 20$ |
| `dailybitinfo` | BitWOBAvg | BitWOBAvg $\leq 100$ *= 1000 |
| `dailybitinfo` | BitTorqAvg | Remove all values BitTorqAvg $> 20000$ |
| `dailybitinfo` | BitRPMAvg | Remove all values BitRPMAvg $> 2000$ |
| `dailybitinfo` | BitMudFlowAvg | Remove all values BitMudFlowAvg $> 1600$ |
| `dailybitinfo` | BitROPAvg | Remove all values BitROPAvg $> 1600$ |
| `dailybitinfo` | BitWOBAvg | Remove all values BitWOBAvg $> 300$ |
| `dailydrilldetail` | WOBAvg | WOBAvg $\leq 100$ *= 1000 |
| `dailydrilldetail` | RPMAvg | Remove all values RPMAvg $>$ 2000 |
| `dailydrilldetail` | MudFlowAvg | Remove all values MudFlowAvg $> 1600$ |
| `dailydrilldetail` | ROPAvg | Remove all values ROPAvg $>$ 200 |
| `dailydrilldetail` | PumpPSIAvg | Remove all values PumpPSIAvg $> 5000$ |

After all improper values in the dataset are taken care of, the next step is to find if there is any underlying structure in the dataset. This is crucial as if some features are incorrectly recorded, or the features have no correlation to each other, there will be no or very weak underlying structures and the modeling job will be much harder. A commonly used analysis method for this step is principal component analysis (PCA), which is a dimensionality reduction method so the data can be visualized. However, for a dataset with a large number of dimensions like ours, there are superior dimensionality reduction methods. In this study, t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008) was used for dimensionality reduction.

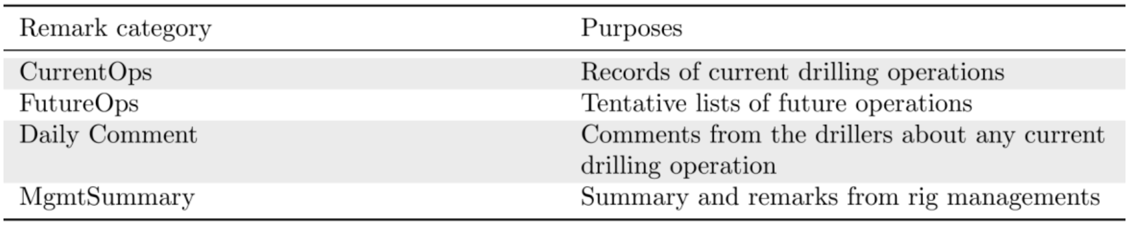
An interesting feature from Figure 7 and Figure 8 is the amount of separation between dailydrilldetail data and dailybitinfo data. Each geothermal field is more cleanly delineated in the t-SNE using dailybitinfo compared to the one from dailydrilldetail. This indicates the data collected from dailybitinfo is of higher quality for ML modeling purpose. However, each geothermal field can be easily identified from t-SNE graphs from dailydrilldetail data, showing that there is a strong underlying data structure even in the dailydrilldetail data. Finally, the results from t-SNE analysis confirmed that there is data sufficiency for further rate-of-penetration modeling.

### 2.2 Nonnumerical data

In additional to standard drilling records described above, there are also nonstandard drilling records in the database. These records are in textual information format, rather than numerical format and need to be processed separately. One such feature that carries important drilling information is the "OpsGroup" column. The "OpsGroup" column encodes what kind of drilling operations happen during a drilling day, from a predefined categorical format: "DRILL" for normal drilling, "TRIP" for tripping, "PROBLM" for problems, and "OTHER" for other operations (i.e. "DRILL"-"DRILL"- "TRIP"-"TRIP"). This feature is important as the content of the daily operations are not transparent from the daily average numerical records alone (i.e. it is hard to infer whether trippings occur or not solely from the numerical records). This will also serve as indicator of nonproductive times, which sometimes are correlated to significant cost-saving opportunities.

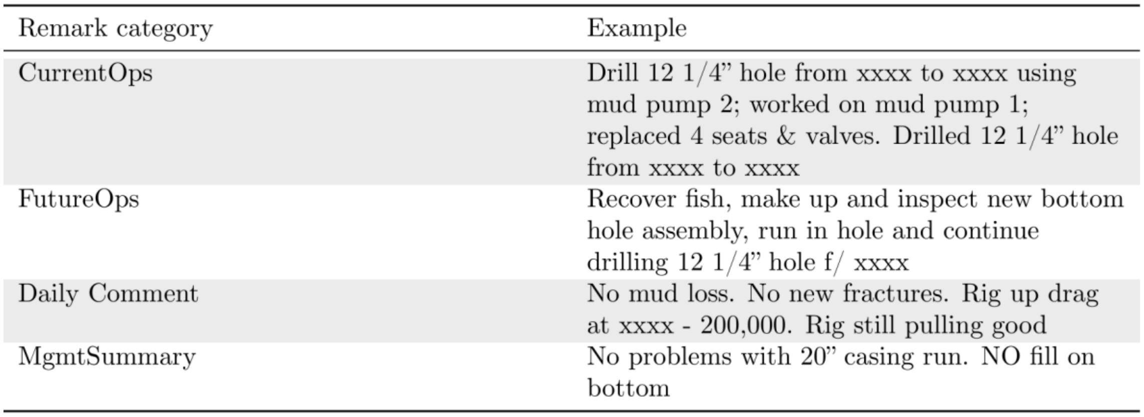
Additionally, the records also contain daily remarks from the drillers in the "Description" column. This feature includes remarks that contain detailed descriptions of what happened that day which cannot be easily described using numbers. There are four different categories that these remarks belong to, depending on the purposes of the remarks (Table 2 and Table 3).

**Table 2: Drilling remarks' categories**



| Remark category | Purposes |
|---|---|
| CurrentOps | Records of current drilling operations |
| FutureOps | Tentative lists of future operations |
| Daily Comment | Comments from the drillers about any current drilling operation |
| MgmtSummary | Summary and remarks from rig managements |

**Table 3: Sample of remarks from the "Description" column**



| Remark category | Example |
|---|---|
| CurrentOps | Drill 12 1/4" hole from xxxx to xxxx using mud pump 2; worked on mud pump 1; replaced 4 seats & valves. Drilled 12 1/4" hole from xxxx to xxxx |
| FutureOps | Recover fish, make up and inspect new bottom hole assembly, run in hole and continue drilling 12 1/4" hole f/ xxxx |
| Daily Comment | No mud loss. No new fractures. Rig up drag at xxxx - 200,000. Rig still pulling good |
| MgmtSummary | No problems with 20" casing run. NO fill on bottom |

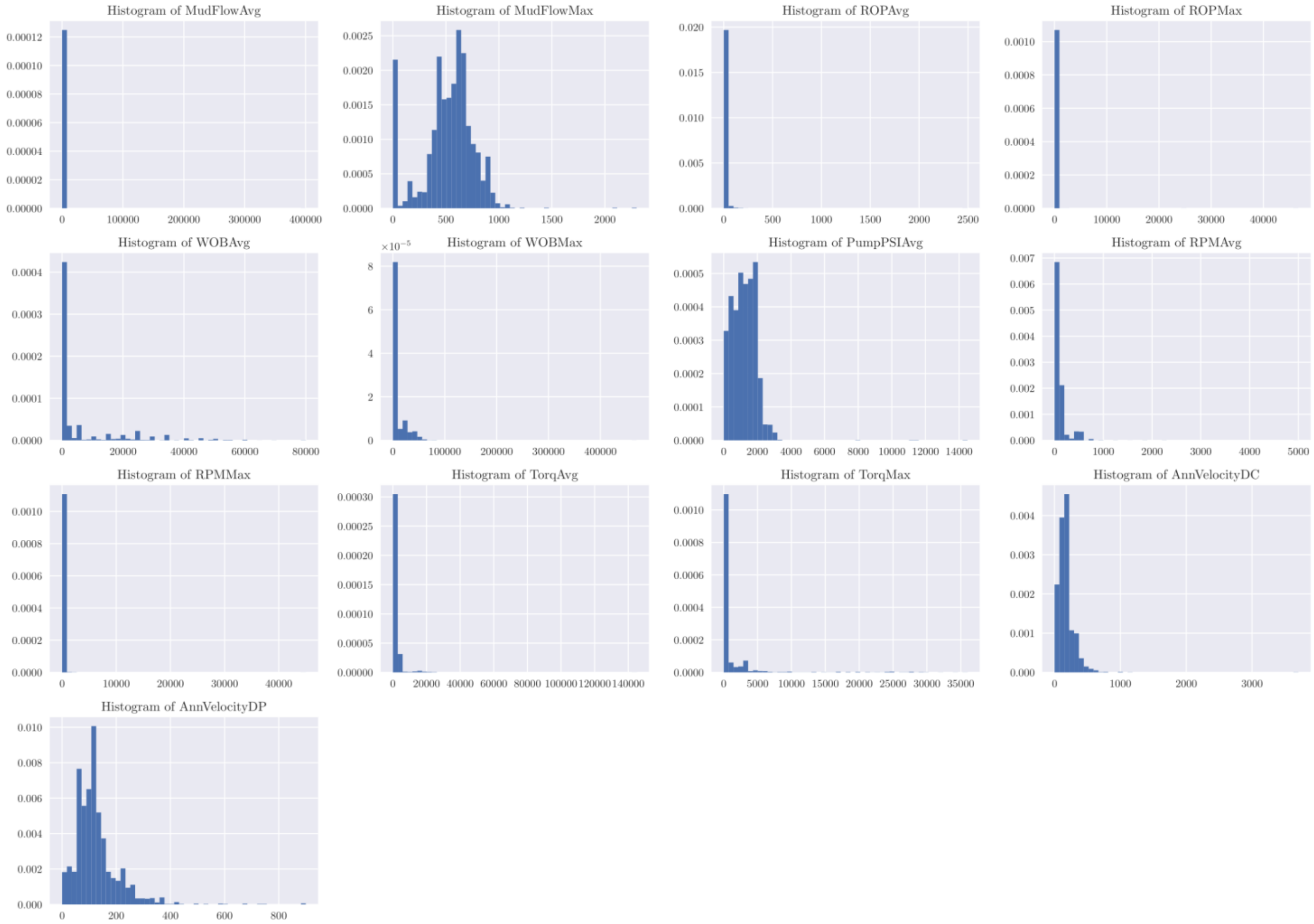


**Figure 1: Histogram of dailydrilldetail data**

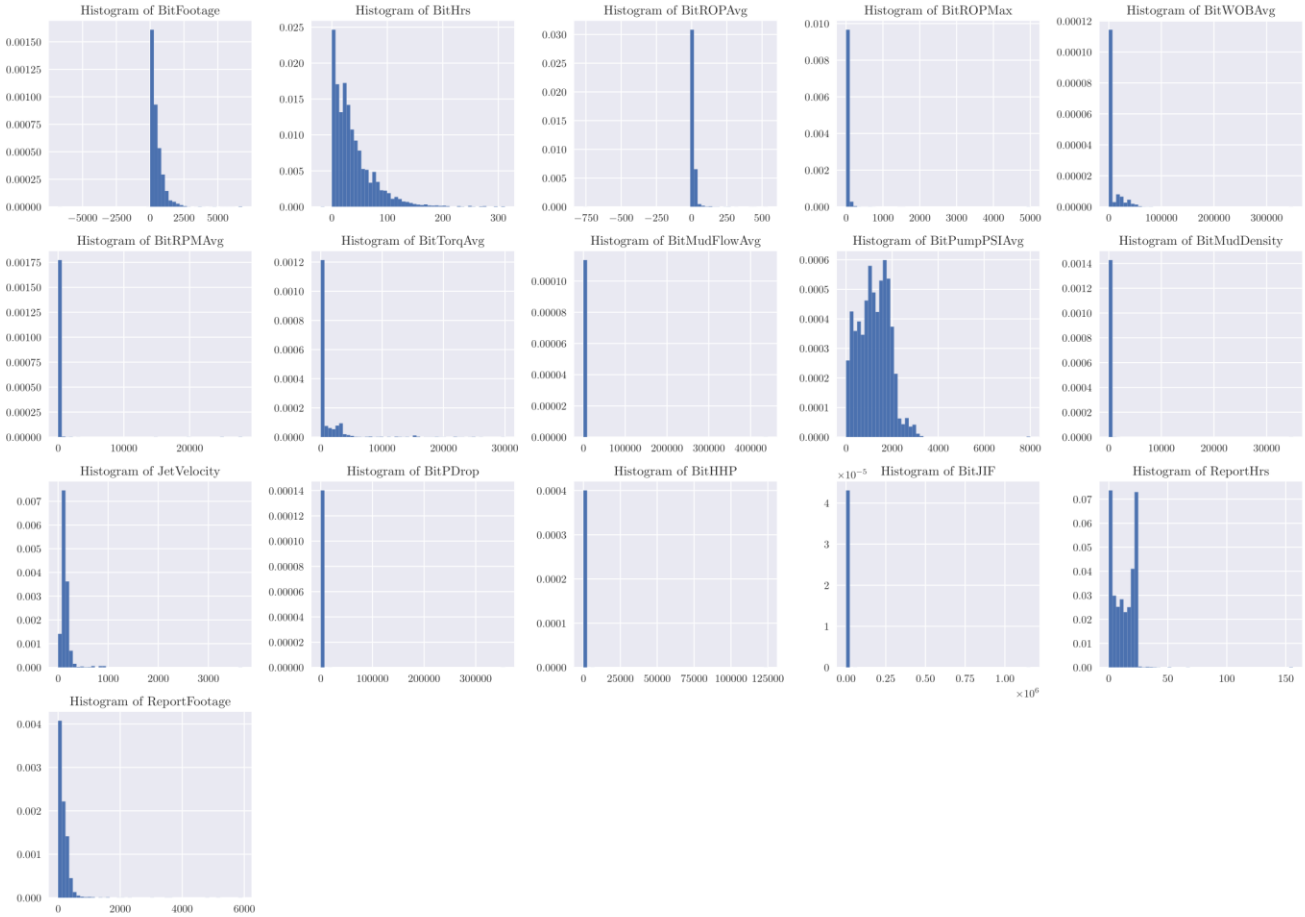


**Figure 2: Histogram of dailybitinfo data**

The lithology of the drilling formations is also useful in creating an accurate model. However, accurate lithological information is hard and expensive to acquire in advance. Therefore, other sources of information are usually used as proxies for lithology information. In our dataset, the rock compositions and properties from the mud-shakers, which are recorded in a "Litho" column, can serve as a proxy. However, the problems of processing textual data encountered with the "Description" column also apply to the "Litho" column. In fact, the records from mud-shakers are even harder to understand compared remarks from the drillers (Table 4).

**Table 4: Sample of daily records from the "Litho" column**

| |
|---|
| 40-60% Phyllite, 40-60% Silstone and Clay |
| xxxx-xxxx 0-20% Phyllite 20-40% Siltstone 40-80% Clay, xxxx-xxxx 0-20% Qtz Vng 30-50% Phyllite 20-50% Silstone 20-40% Clay, xxxx-xxxx 0-20% Phyllite 20-40% Silstone 40-60% Clay |
| Mod ylsh brn; firm to hd; sbrnd ctgs; aphnc w rd glassy incl; com qtz & calct fill amyg; tr clystn |
| Rhyolite/ Basalt; lt-med gry, prplish gry, hd, subblky-subang ctngs, aphnc-sli porh, (qtz, calct, chalcedony) fillied amyg |

It could be very beneficial for the modeling process if the model can incorporate textual information in additional to the conventional numerical information. However, despite the wealth of information that the textual data contains, extracting that information is not easy as the remarks are records in written English with no standardized structure. Conventionally, manual processing is the only way to extract useful information out of the remarks, which is highly time- and resource-consuming. Even then, it is hard to incorporate the extracted information into the ML model to use together with other drilling records. Hence, these kinds of textual data are very often ignored in ML modeling. This study will discuss about how to incorporate these remarks effectively without manual processing in Section 4.

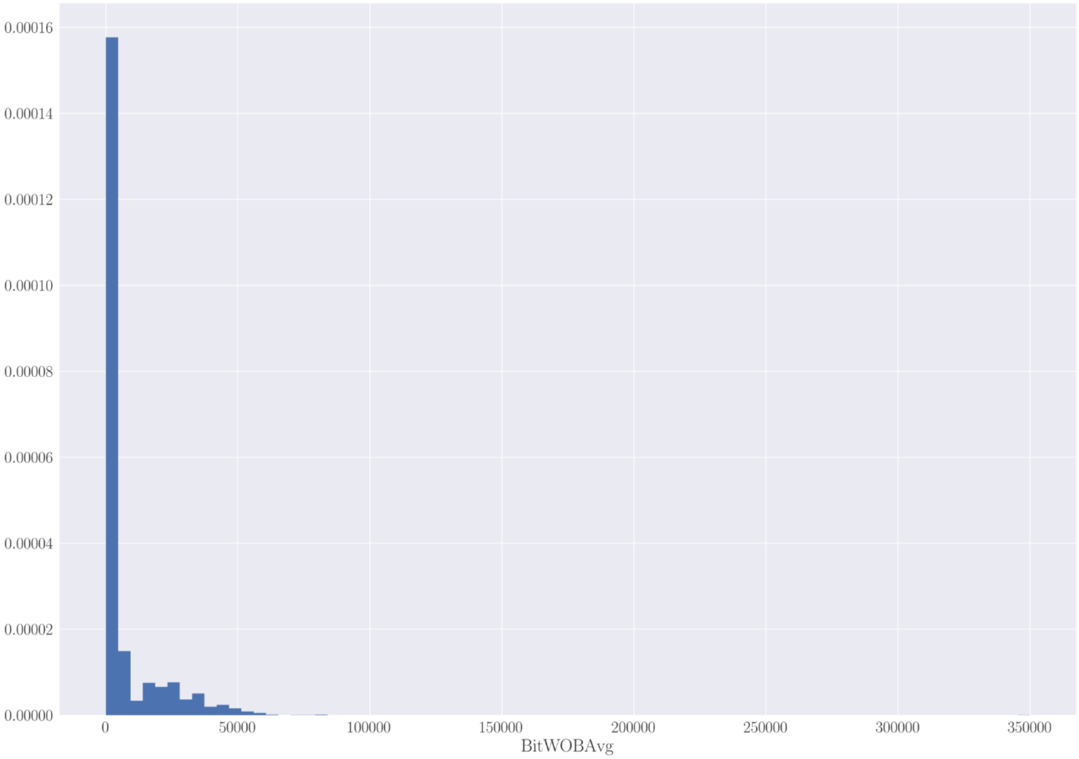


**Figure 3: Histogram of BitWOBAvg data**

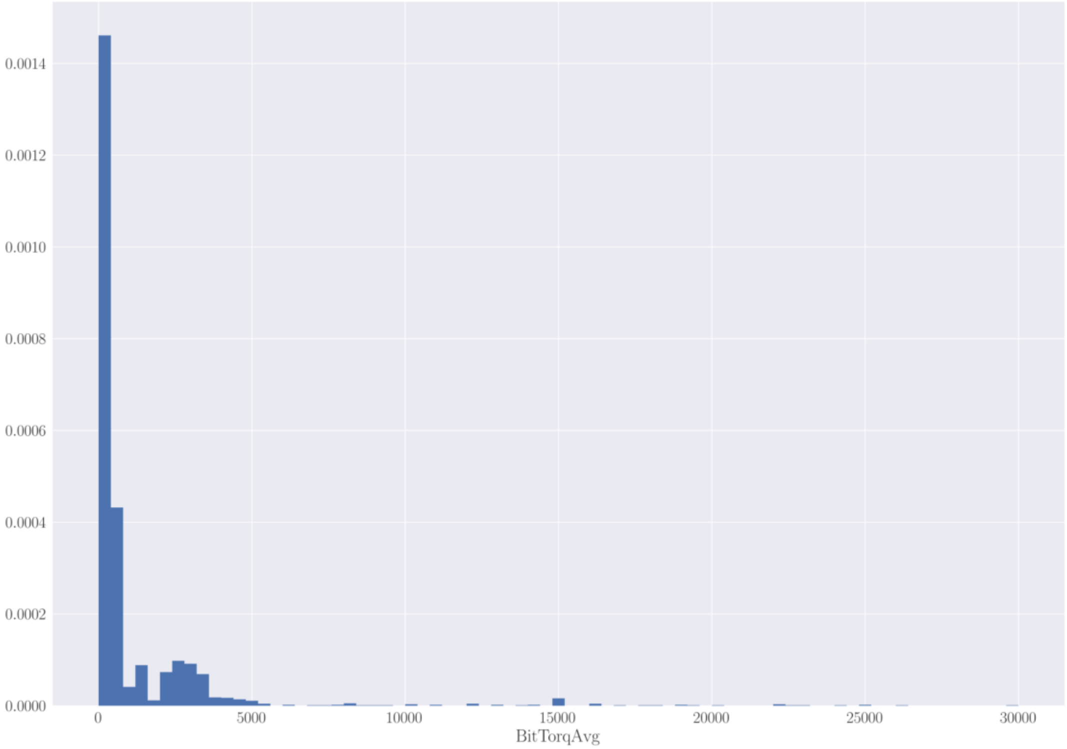


**Figure 4: Histogram of BitTorqAvg data**

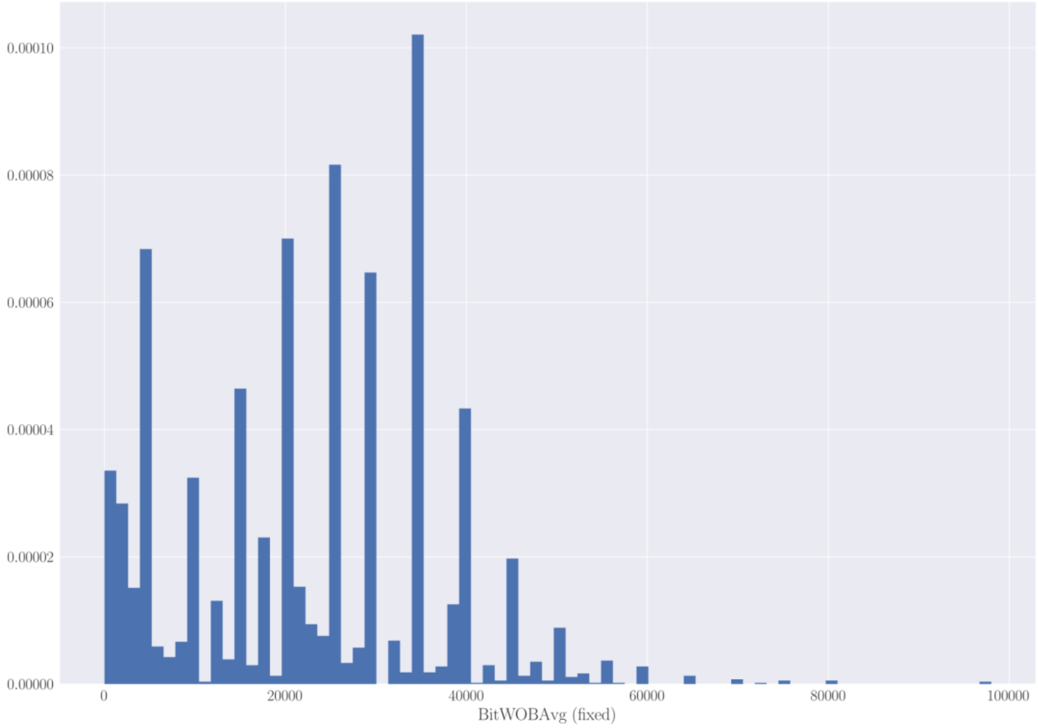


**Figure 5: Histogram of BitWOBAvg data (fixed)**

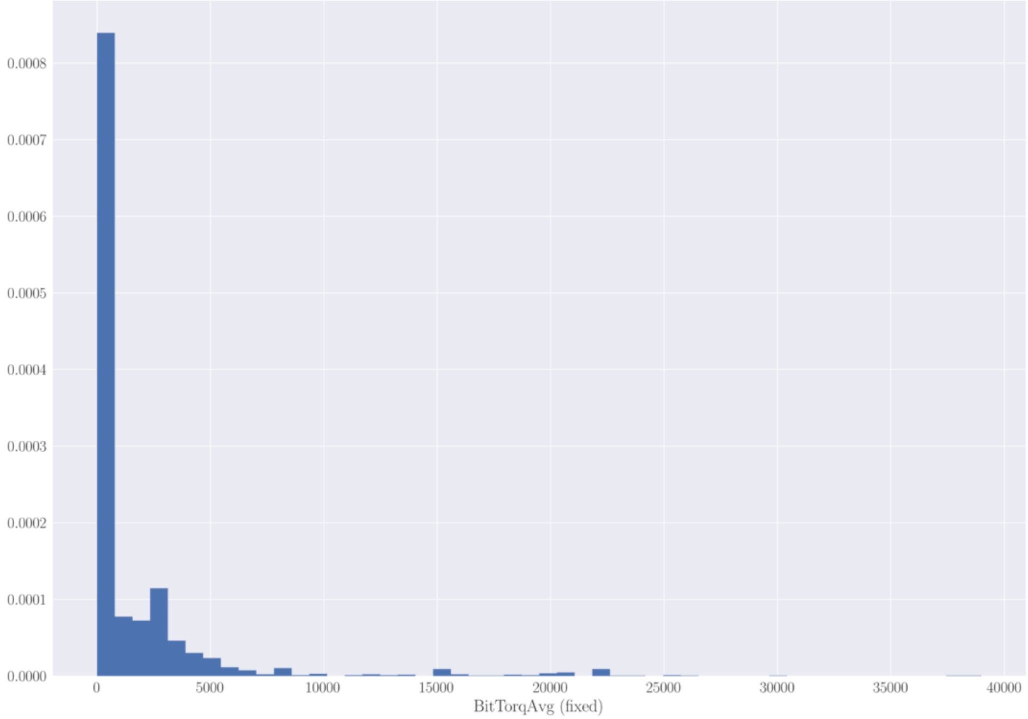


**Figure 6: Histogram of BitTorqAvg data(fixed)**

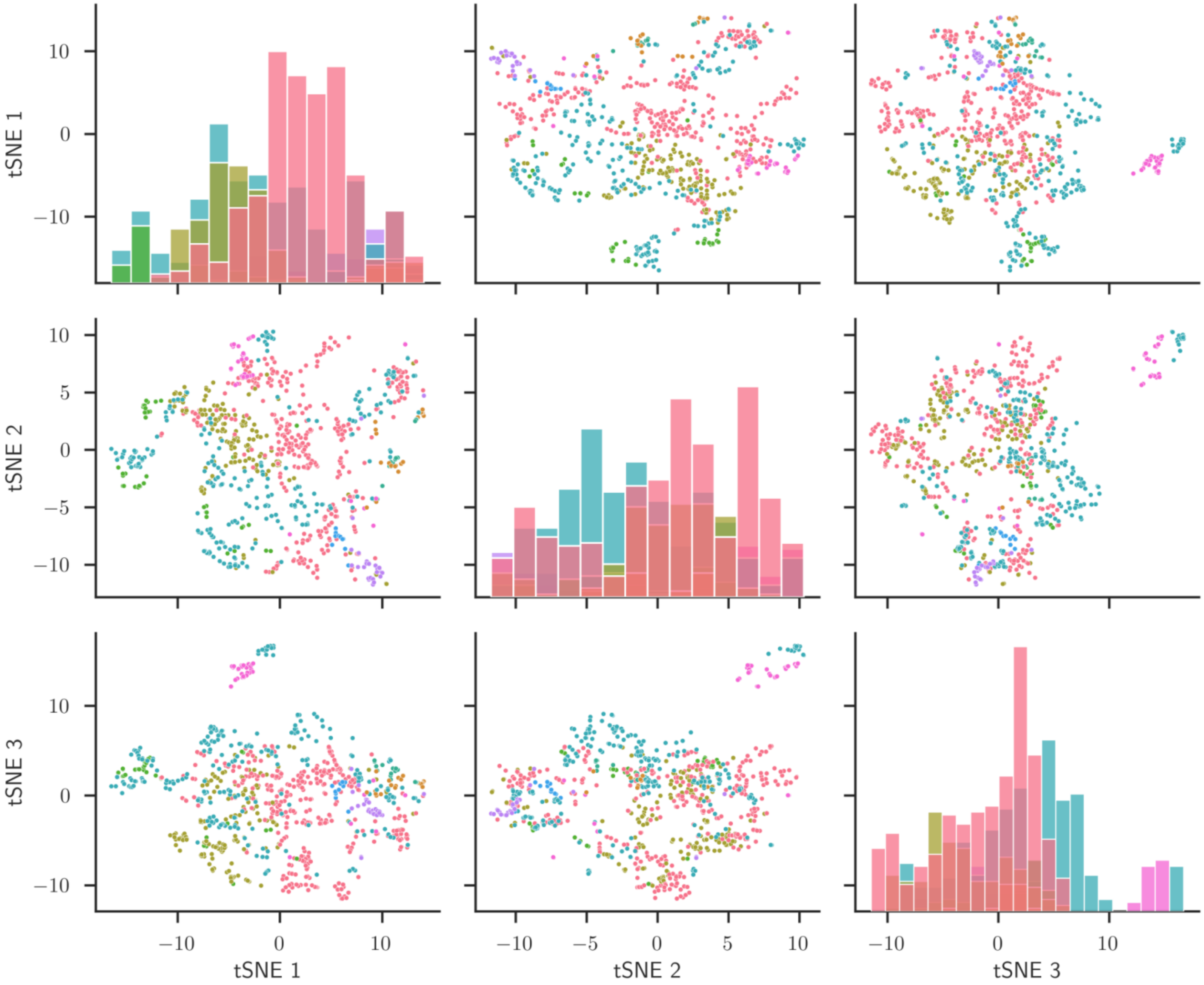


**Figure 7: t-SNE results on dailydrilldetail data. Each dot represents a datapoint and each color represents a different geothermal field**

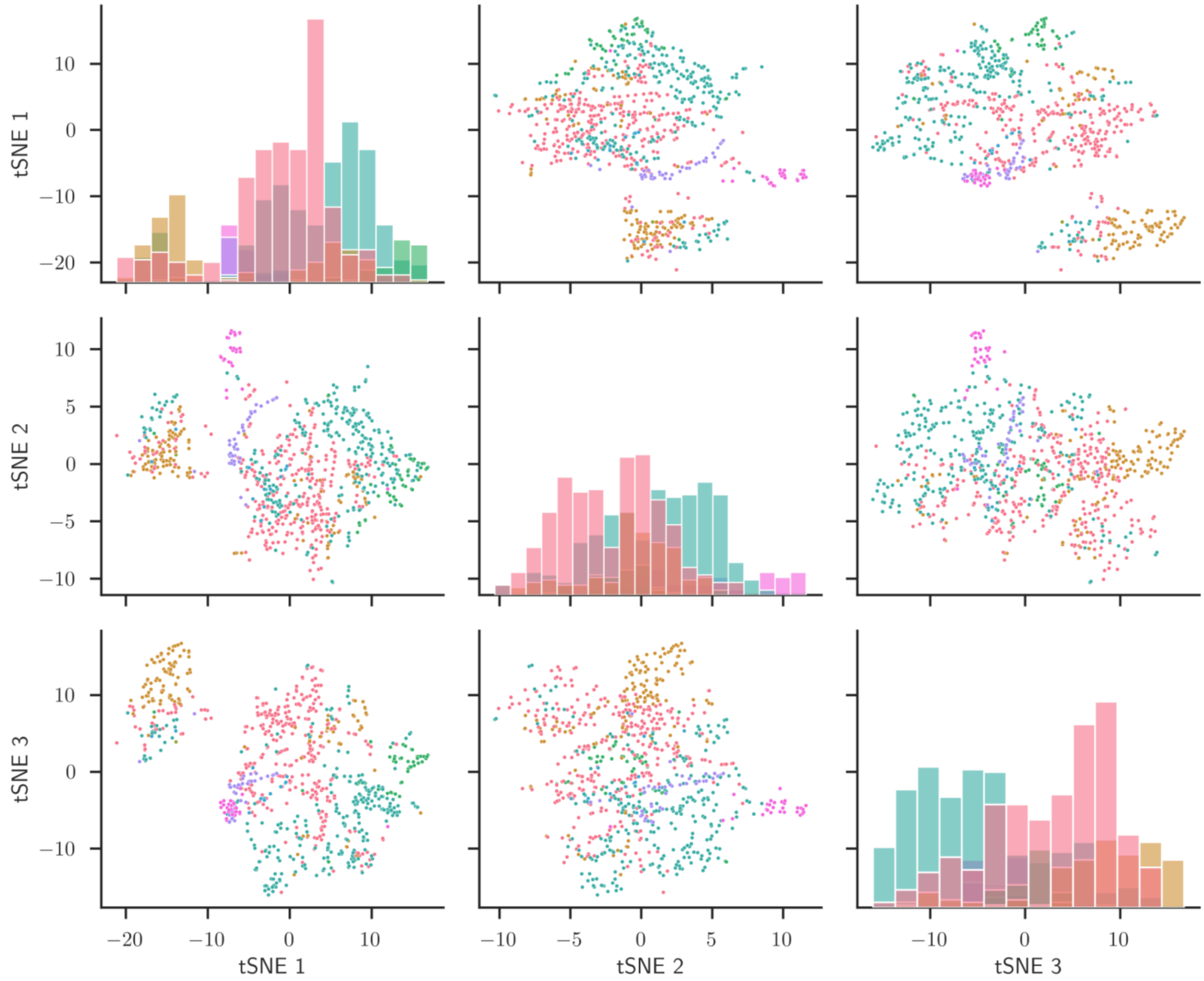


**Figure 8: t-SNE results on dailybitinfo data. Each dot represents a datapoint and and each color represents a different geothermal field**

## 3. MODELING WITH DEEP NEURAL NETWORKS

Since 2010, deep neural network (DNN) methods have gained remarkable improvements, and sometimes outperform humans in some tasks. From the humble beginning of AlexNet (2012) with merely 66 million parameters to the modern GPT-3 (2020) with more than 17 billion parameters, DNN sizes have grown astoundingly in the last decade, and so have their capabilities. AlexNet started out with simple object classification tasks, but now GPT-3 is capable of holding natural language conversation with humans (Brown, Mann, Ryder, Subbiah, et al., 2020). This remarkable advance strongly corresponds to how complex the network is, i.e. the number of layers in the network.

However, simply stacking layers upon layers does not really make a DNN powerful. DNNs are notoriously hard to train properly, coupled with the fact that adding more parameters to the network tends to overfit the training dataset. It took research and experimentation to arrive at modern DNN architecture.

This section discusses the dual-path dual-branch deep residual neural network (DPDBN) architecture used in this study: the methodologies and the results of using DPDBN in well optimizations.

### 3.1 Dual-path dual-branch deep residual neural network

In this study, a new neural network architecture, called dual-path dual-branch neural network (DPDBN) was used. The DPDBN architecture is composed of multiple building blocks (Figure 9), each building block can be formally defined as:

Let:

$x_\ell = [z_1, z_2, z_3, \ldots, z_{\ell-1}, y_{\ell-1}]$

$r^1 = f_\ell^1(x_\ell); r^2 = f_\ell^2(x_\ell)$

$F_\ell(x_\ell) = [z_\ell, y_\ell]$

Then:

$$[z_\ell, y_\ell] = \begin{cases} [\alpha g_\ell^1(r^1) + (1-\alpha) g_\ell^2(r^2), y_{\ell-1} + \beta h_\ell^1(r^1) + (1-\beta) h_\ell^2(r^2)], & \text{in forward phase} \\ [\gamma g_\ell^1(r^1) + (1-\gamma) g_\ell^2(r^2), y_{\ell-1} + \delta h_\ell^1(r^1) + (1-\delta) h_\ell^2(r^2)], & \text{in backpropagation phase} \\ [\mathbb{E}[\alpha] g_\ell^1(r^1) + (1-\mathbb{E}[\alpha]) g_\ell^2(r^2), y_{\ell-1} + \mathbb{E}[\beta] h_\ell^1(r^1) + (1-\mathbb{E}[\beta]) h_\ell^2(r^2)], & \text{in validation} \end{cases}$$

where $x_\ell$ is the aggregated output of the previous block, ${f_\ell}^1$, ${f_\ell}^2$, ${g_\ell}^1$, ${g_\ell}^2$, ${h_\ell}^1$, ${h_\ell}^2$ are any feed-forward neural network , and α, β, γ, δ are variables sampled from a uniformly random distribution in the range [0-1]

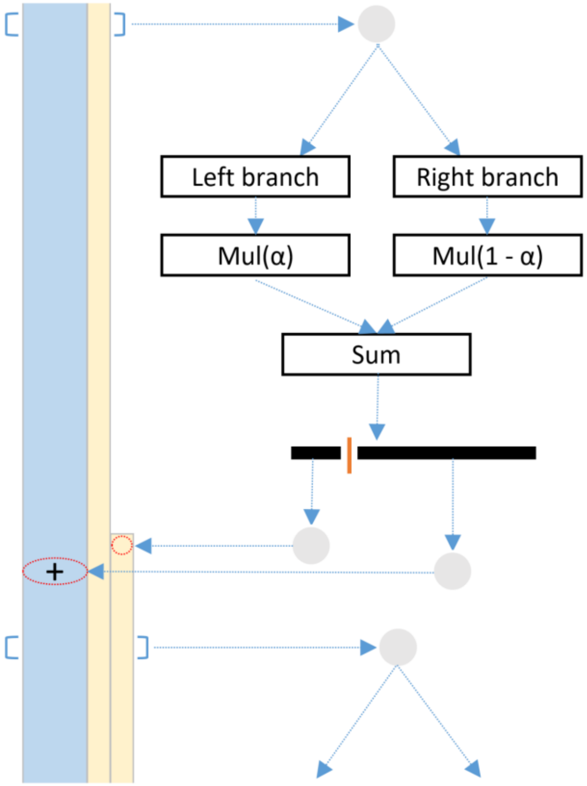


**Figure 9: Dual-path dual-branch Network building block**

### 3.2 Methodology

3.2.1 Rate-of-penetration modeling

As discussed before, the ML model developed to help drill a successful well will have ROP as the main criterion of optimization. For that purpose, a ML model based on DPDBN has been developed. The target is to create a ML model that is able to accurately predict ROP from the drilling records.

The daily-averaged drilling records described in Section 2 serve as inputs, and the daily-averaged ROP will be the target. However, these inputs and outputs cannot be used directly as they are due to multiple problems in the original records, as uncovered in the second section of Section 2. Therefore, some data preprocessing works are needed to make the data ready for the later modeling part. The data preprocessing work done on the inputs includes the follow steps:

1. Take care of improper values
2. Fill missing values or extreme outliers with nan
3. Scale the features to have mean of zero and standard deviation of one
4. Create the feature mask

5. Replace missing values or extreme outliers with zero
6. Concatenate scaled data with the feature mask to create the final input

The first step was described in detail in Section 2. In addition, to prevent data leaking from the inputs to the outputs, some features must be removed from the inputs. The removed data includes features that record footage or total drilling time of the bits (BitFootage, BitHrs, etc.). In Step 2, any missing data, or any extreme outlier deemed by the previous step, is replaced with a NAN value. The purpose of this step is to remove the influences of any extreme outlier, and make the data easier to work with in the later steps. The third step is scaling the data, which is a requirement for most neural network models. One of the most common scaling methods is standard scaling, which transforms the data so that each feature has the mean of zero and standard deviation of one. Standard scaling was also the chosen scaling method for this project due to its effectiveness and simplicity. After scaling, a feature mask is created. A feature mask is a table which has the same dimension as the scaled data, where a value of one in the table indicates a missing/outlier value in the original data, and value of zero indicate the data is present. The purpose of the features mask is to inform the locations of missing data so the model can react accordingly. Using a feature mask is a strategy that helps a ML model deal with incomplete information, which can also be used to describe the dataset used in the project. Without this feature mask, any row with missing data has to be dropped, which may reduce the size of data by up to 75 %. Finally, the input is the simple concatenation of the scaled-data and the feature mask.

Considering the purpose of this model, which is to help drillers to model and maximize the ROP, the model can try to predict the next day ROP from the current drilling record, giving drillers ample warning times for potential problems. Hence, the model will be trained with drilling record at time $t-1$ and asked to predict the ROP at time $t$.

Conventionally in machine learning, the training set and the validation set are constructed by randomly selecting datapoints from the original dataset. This is to ensure that both the training and the validation data have the same underlying distributions, which is an important assumption when using them to gauge the network performances/generalizations. However, considering how a ROP model would be used in drilling, the model is trained on drilling records of known wells so that it can accurately model the ROP of any new similar well in the same area. This means that a random train/validation splittingscheme is not suitable as there is no information about the new well. To deal with this problem, another train/validation splittingscheme was used; rather than splittingthe information randomly, the train/validation is chosen well-by-well (i.e. there are drilling records of well 10, 11, 12, 13, 14, and 15; then the records of wells 10, 11, 12, and 13 will be included in the training set while the records of wells 14, and 15 will be included in the validation set). This train/validation splitting scheme mimics how drillers would use a ROP model in production. Both approaches were considered in the study, with the train/validation ratio of 3:1.

### 3.2.2 Other than rate-of-penetration modeling

As discussed at the end of Section 2, there are other important parameters to the success of a well. Constructing a model that can give future warning about potential problems and nonproductive times is also beneficial to the drillers. Rather than ROP, this model will predict what operations will happen tomorrow (the "OpsGroup" feature). In this project, a model to predict the two leading causes of nonproductive times ("TRIP" and "PROBLM") was developed. This model takes the drilling record at time $t-1$ and predicts whether "TRIP" or "PROBLM" happens at time $t$. This will give drillers ample warning time to prevent/remediate potential nonproductive periods. For this purpose, the inputs, the train/validation splitting scheme, and the models are similar as in the ROP modeling process. The only difference is the output, which will be coming from the "OpsGroup" column.

## 3.3 Results and discussion

### 3.3.1 Rate-of-penetration modeling

From Figure 10, it can be seen that in random train/validation splitting, the ROP model performed reasonably well on the drilling records dataset. The model achieves an $R^2$ score of 0.56 on the training set and 0.54 on the validation set. The 95% confidence interval of the model on Figure 11 also indicates good prediction quality.

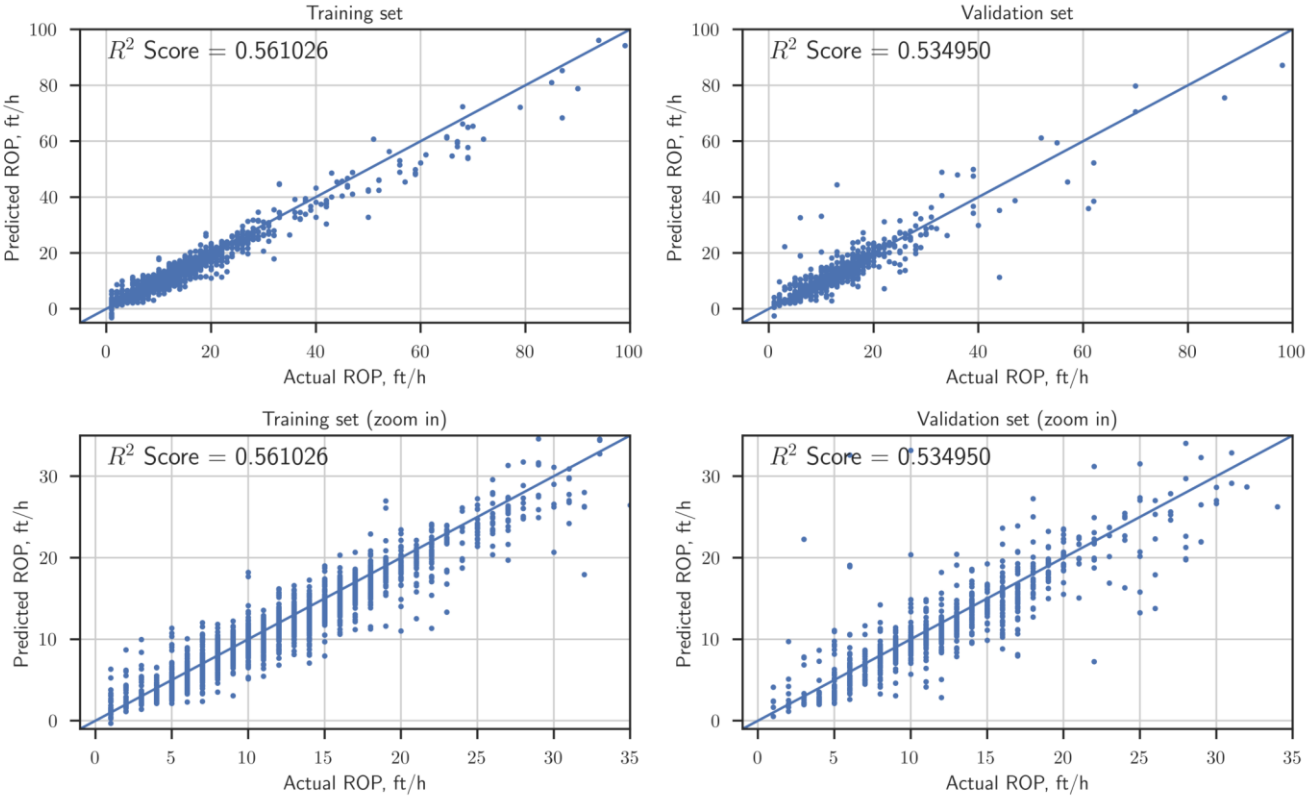


**Figure 10: Correlation plot for ROP modeling, random train/validation splitting**

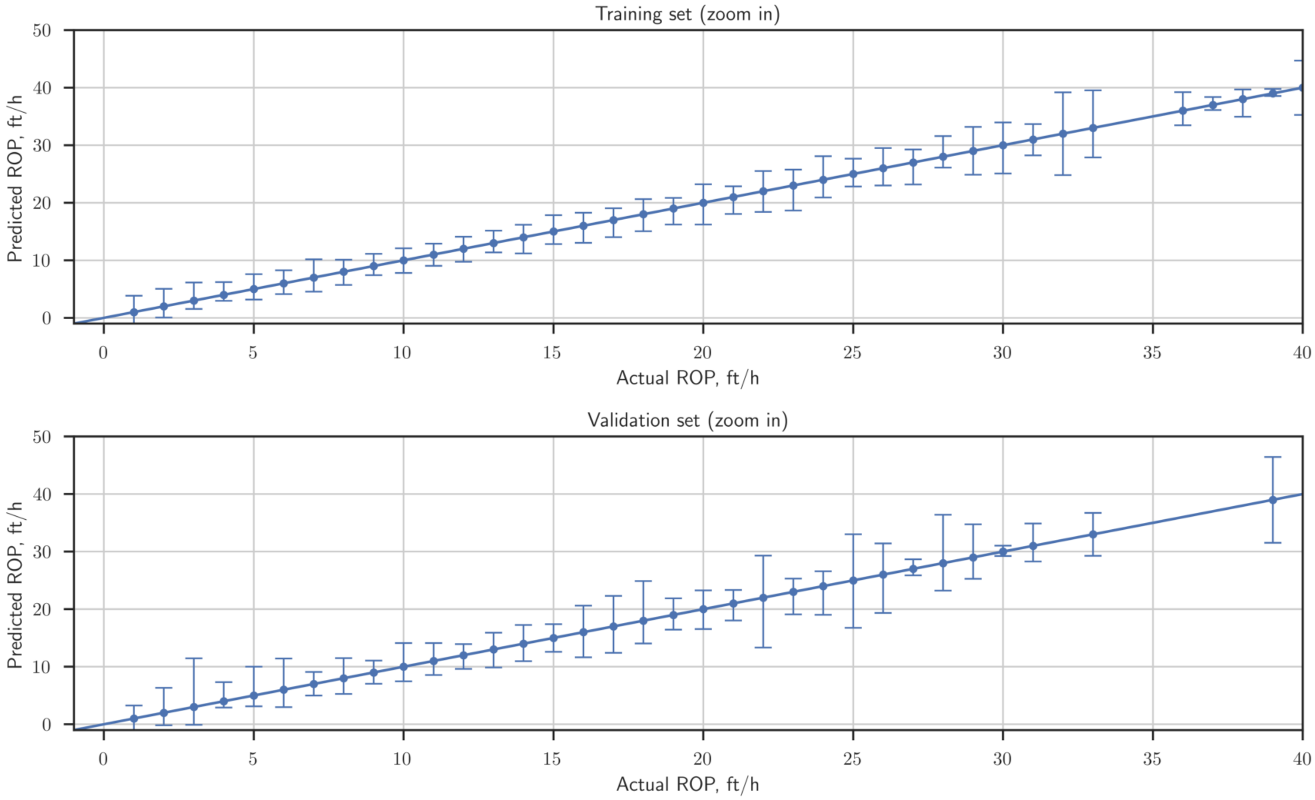


**Figure 11: 95% confidence interval plot for ROP modeling, random train/validation splitting**

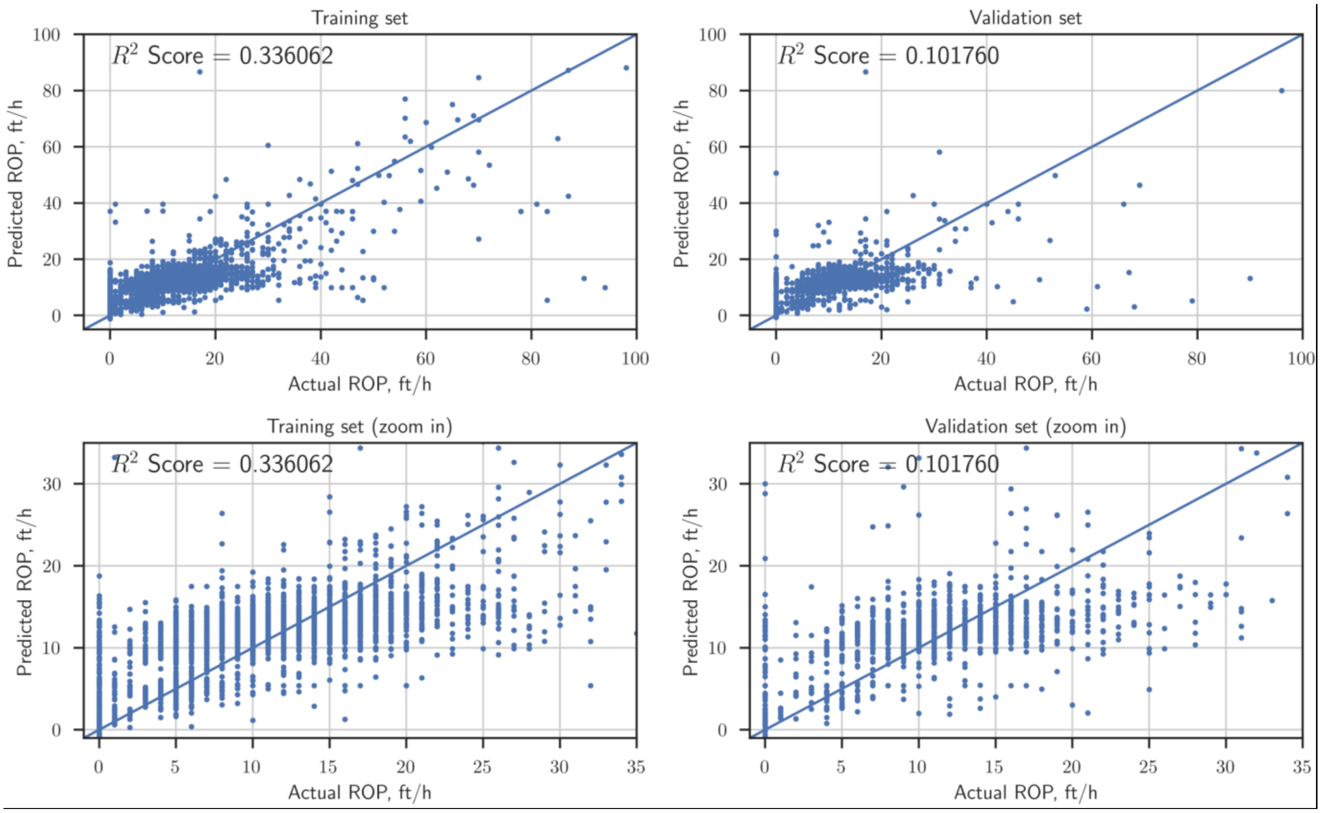


**Figure 12: Correlation plot for ROP modeling, well-by-well train/validation splitting**

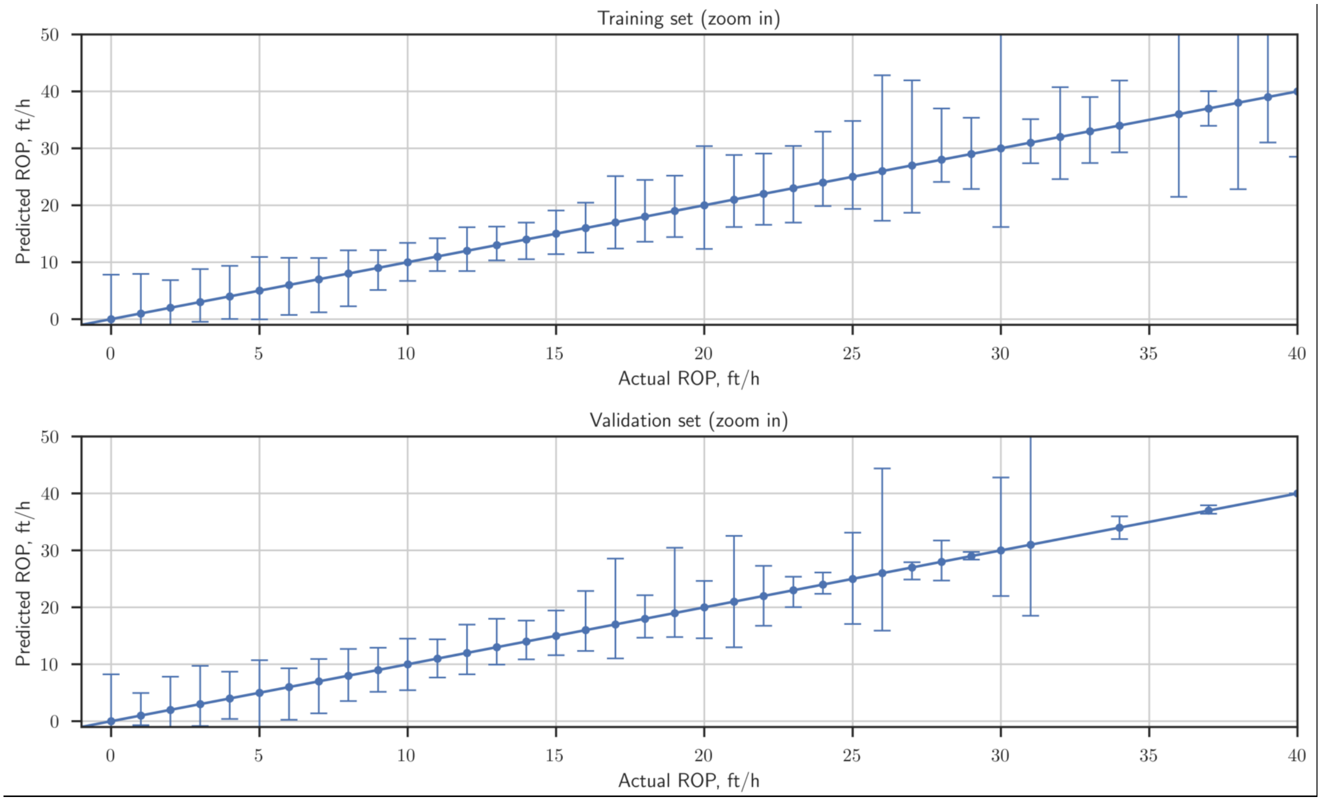


**Figure 13: 95% confidence interval plot for ROP modeling, well-by-well train/validation splitting**

However, switching from random to well-by-well train/validation splitting has a detrimental effect on the results. The model only achieves an $R^2$ score of only 0.34 on the training set and 0.10 on the validation set (Figure 12). The 95% confidence interval of the model on Figure 13 shows the high uncertainties in the predictions (i.e. a real ROP of 10 ft/hr is often predicted to be in the range of 5 ft/hr to 15 ft/hr).

It is clear from the result that even with an identical machine learning model, using random train/validation splitting produced a model that is accurate enough for use in production, while using well-by-well train/validation splitting produced a model with subpar prediction quality. Unfortunately, as discussed before, the model trained on random train/validation splitting data has limited use in production despite its superior accuracy as it requires knowledge from the future. On the contrary, the model trained on well-by-well train/validation splitting has use in production, but its accuracy was not sufficient.

### 3.3.2 Other than rate-of-penetration modeling

In contrast to the poor results with ROP modeling, modeling with other than ROP information did produce a model that is usable in production. In Figure 14, the left confusion matrix is from “TRIP” prediction model, while the right confusion matrix is from “PROBLM” prediction model.

The left matrix indicates a really good result as the model predictions are correct most of the time, especially in when tripping happens, with ratio of True positive to False positive ratio nearly equal to 3.5:1.

On the other hand, with the right confusion matrix, while it shows good results when forecasting “PROBLM” in the cases where problem does not happen, the model does poorly when problem does happen. When problem does happen, the ratio of True positive to False positive ratio is 1:1. In other words, if the model forecasts problem in the future, it is just as reliable as flips of a coin.

Unfortunately, although “problem” does not happen in many drilling processes, any instance of it can result in extensive nonproductive time and consequently significant costs. Therefore, it is important that the model be able to predict possible future “problem” correctly rather than the opposite. The final result is a model that is useful in production. The trip prediction model, with its accurate prediction, is able to give drillers at least one day in advance warning, which can result in proper preparation and reduction of nonproductive time.

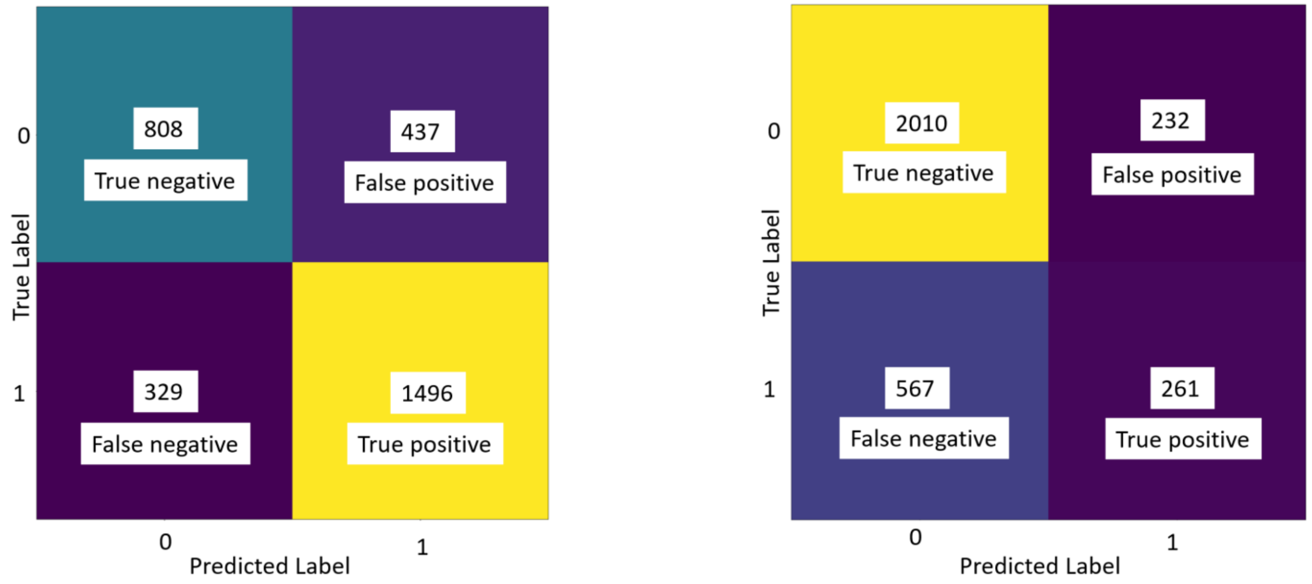


**Figure 14: Confusion matrix for tripping and problem predictions: left for tripping predictions, right for problem predictions. One indicates tripping/problem did happen, zero indicates tripping/problem did not happen.**

## 4. NATURAL LANGUAGE PROCESSING

In additional to impressive results in computer vision that made self-driving vehicles a reality, modern deep neural network also revolutionized the field of natural language processing (NLP). The most modern NLP models, with the help of deep neural networks, are capable of NLP tasks that exceed the highest expectation of a decade ago. With GPT-3 (Brown et al., 2020) or RoBERTa (Liu, et al., 2019), researchers have created NLP models that are capable of holding natural conversations with humans, or writing an essay from a single sentence prompt that is indistinguishable from human writings.

This section will discuss the usage of natural language processing (NLP) in the drilling modeling process. More specifically, this section shows how to prepare the data and train the BERT model, and how to integrate the new results into the results obtained from Section 3.

### 4.1 Bidirectional Encoder Representations from Transformers

A significant breakthrough came in 2017 with the Transformer architecture. On top of the improved quality, Transformer also brings an order of magnitude of improvements in training speed when dealing with NLP models. As of 2022, the Transformer model is the preferred architecture for most NLP tasks.

A new and very important concept in the Transformer is the concept of self-attention. Formally, the concept of self-attention can be described with a concept of query-key-value attention mechanism, where query, key, and value are three vectors which can be defined as:

- Query: $Q = W_q \times [\ embedded\ input\ ]$, vector the attention mechanism is looking from
- Key: $K = W_k \times [\ embedded\ input\ ]$, vector the attention mechanism is looking to
- Value: $V = W_v \times [\ embedded\ input\ ]$, the value of the attention

where $W_q$, $W_k$, and $W_v$ are three learnable matrices with same dimension. There are a query, a key, and a value matrix corresponding to every embedded element in the input. Then the self-attention can be defined as:

$$Attention(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d}}) \cdot V$$

where $K^T$ is the transpose of the key matrix and $d$ is is the dimension of $K$. This concept is illustrated in Figure 15.

One major benefit of the self-attention head model is it can easily expand, both in the horizontal direction and the vertical direction. If the model is expanded horizontally, then it is called a multihead self-attention model where each head produces a different self-attention. If the model is expanded vertically, then it is called a multilayer self-attention model where the previous head's self-attention is the input for next self-attention head. The Transformer encoder-decoder model takes full advantage of the self-attention mechanism by expanding the self-attention head vertically and horizontally (Figure 16).

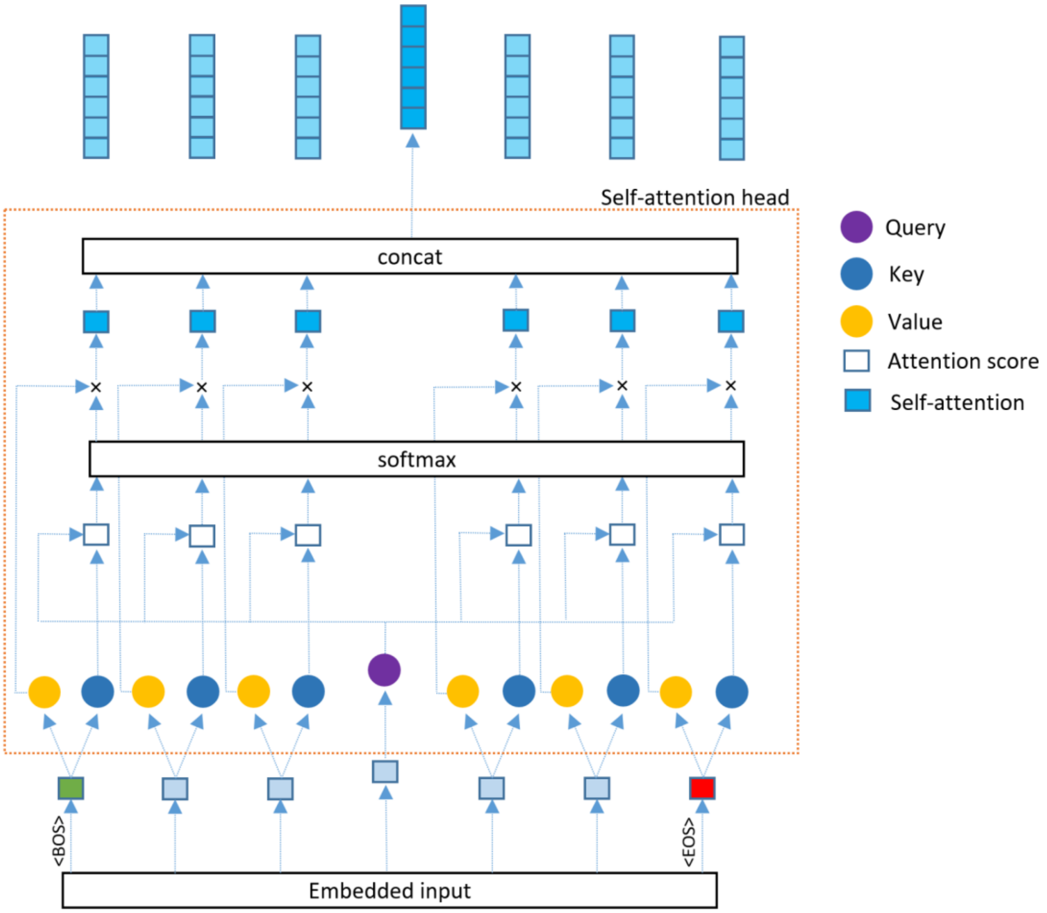


**Figure 15: Self-attention head**

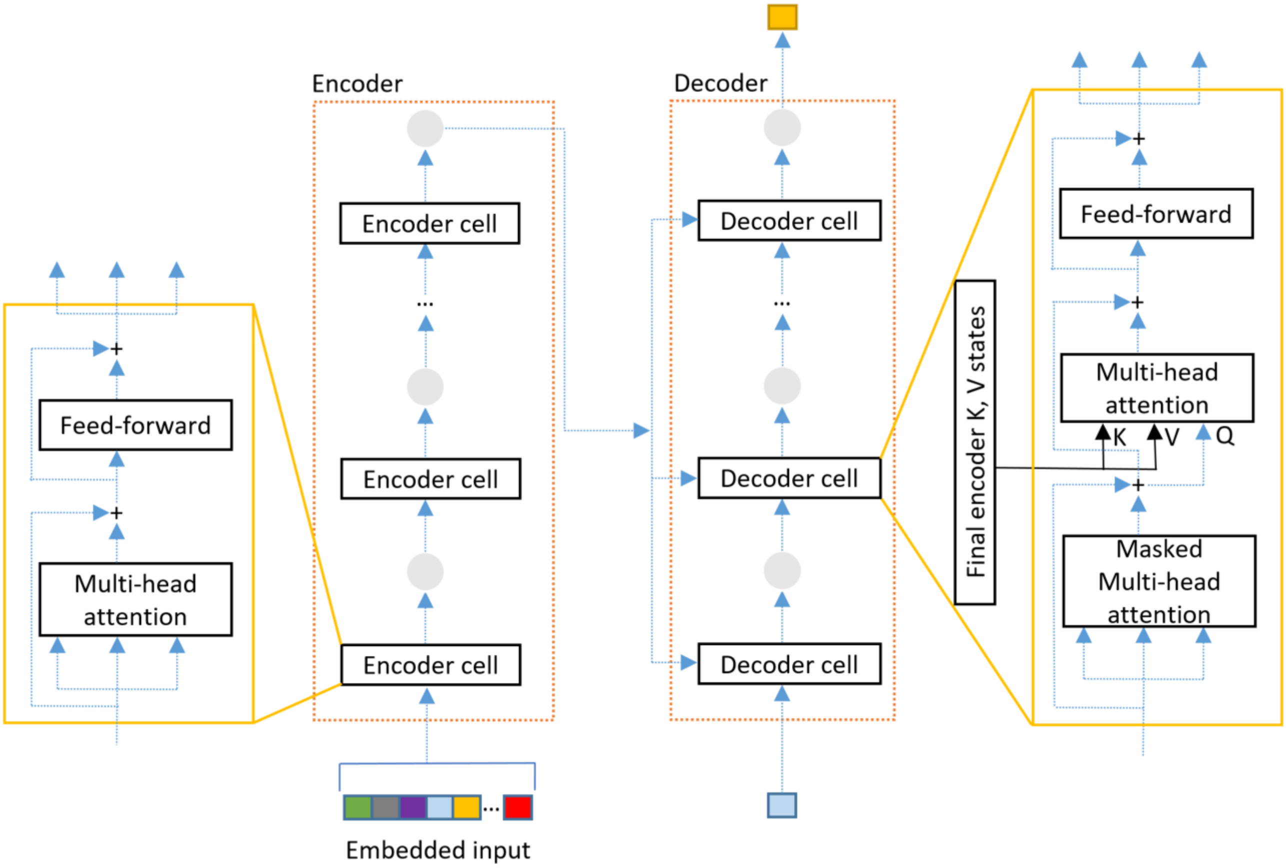


**Figure 16: Transformer architecture**

With the introduction of Transformer, a new era of NLP was opened. Now NLP models can process very long inputs, can be effectively trained on terabyte-size datasets, and can achieve near human performance in some NLP tasks. However, these state-of-the-art NLP models all share some common points: they are all enormous with billions of trainable parameters, and they are all trained on extremely large datasets, which requires an enormous amount of computational power. Therefore, training a model for a specific NLP task until it achieves state-of-the-art performance is something only the biggest organizations can achieve.

Due to the difficulty in training a good NLP model, transfer learning is an active field of research in NLP. The goal of transfer learning is to transfer knowledge and information between different models in order to accelerate the training process. Transfer learning is especially effective when dealing with NLP tasks. Although NLP tasks may be different drastically from each other, most of the time they all have same common requirement to the NLP models built to solve them, which is the ability to read and understand written natural language.

Bidirectional Encoder Representations from Transformers (BERT) (Devlin, et al., 2019) is a Transformer based NLP model with very good results on most NLP tasks and exceptional transfer learning capability. The architecture of BERT is remarkably simple: it is the encoder part of the Transformer architecture (Figure 16).

The main idea is that: to fine tune BERT to the downstream task, the input is fed into the pretrained BERT model, and the output corresponds to the <CLS> token will be used as input for a feed-forward neural network. This feed-forward neural network will have its outputs suitable for the downstream tasks, and will be trained together with the model used in the downstream tasks (Figure 17). In most cases, the pretrained BERT model's parameters are frozen and will not be modified in the fine tuning process.

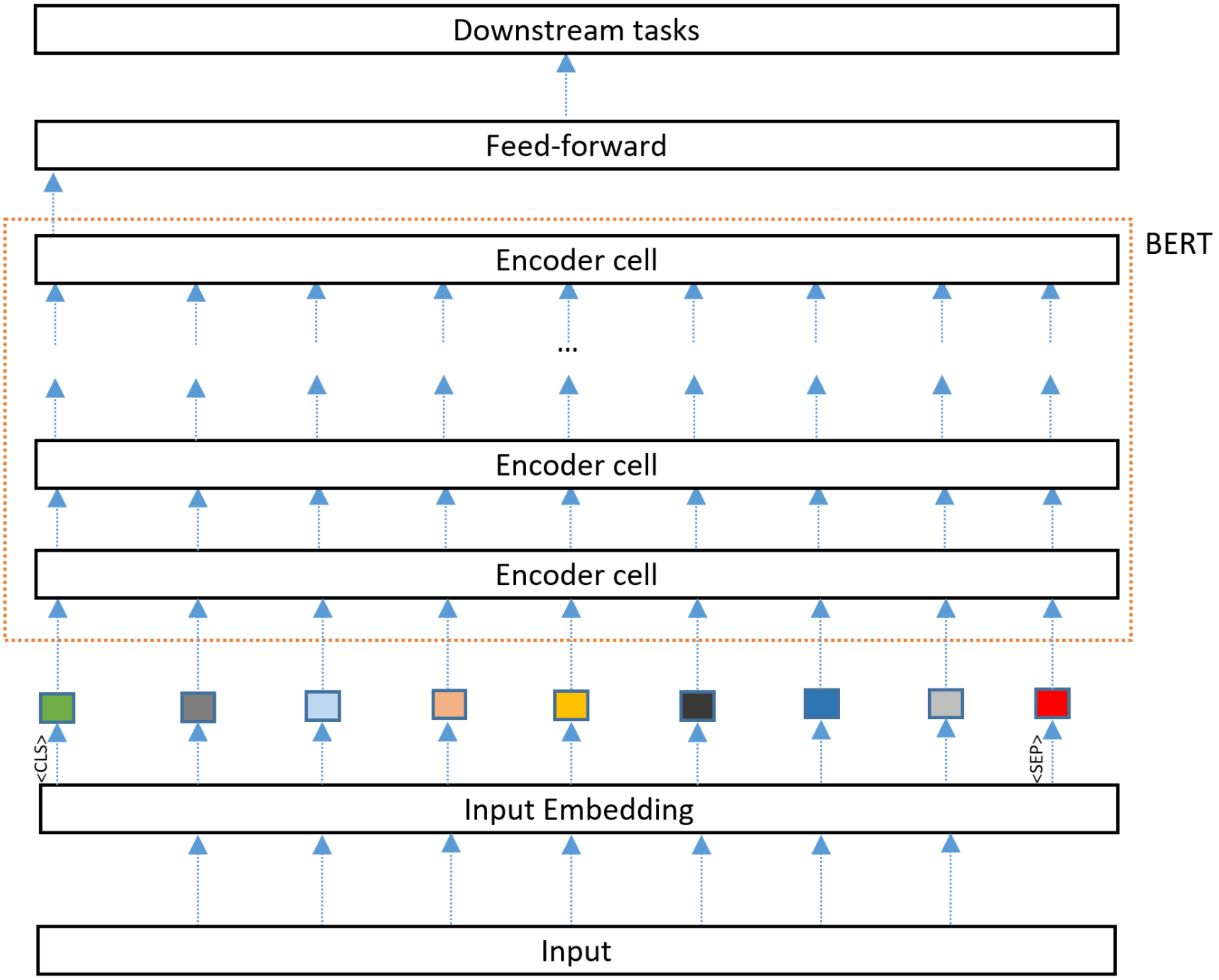


**Figure 17: BERT architecture**

### 4.2 Methodology

4.2.1 Qualification of BERT

In this project, a pretrained BERT model called *bert-base-uncased* was used to process the textual drilling records. This pretrained model is a case-insensitive, 12 layer, 12 attention head, and 110 million parameter model. The neural network is trained on the BookCorpus, a corpus of 11038 unpublished books, and the entire English Wikipedia.

The first task was to qualify the pretrained BERT model, so that is it is able to parse and understand these remarks and lithological comments from the drilling records. As described at the end of Section 2, there are four different categories for the remarks. Together with the lithological type, there are five categories that any piece of textual data can fall into. If the network can truly understand the contents of the remarks/comments, then it should be easy to categorize them into the correct type.

In order to test its performances on the drilling records, the remarks/comments were fed into the pretrained BERT model, then the output corresponding to the <CLS> token was fed to another feed-forward neural network to categorize the source of the input. All the remarks/comments in the drilling records were used as the dataset for this task; 70% of the randomly selected remarks/comments were used for the training, and the rests were used for validation.

4.2.2 Modeling with Natural Language Processing

As discussed at the end of Section 2, in addition to standard numerical records, there is also a wealth of textual data in the daily drilling record. This textual information often provides valuable information that is not available elsewhere. However, this textual information is recorded as written English remarks without any standardized structure. It is very hard to process these records numerically. However, BERT provided an elegant solution to process these textual records.

The numerical representations of the remarks/comments described previously, together with the numerical drilling records, were used as inputs for this task. The new representations of the remarks/comments is concatenated into the final input described in Section 3. Except with the new inputs, the methodology and the goals of the studies described in this section are identical to the methodology and the goals of Section 3: DPDBN is the still the deep neural network architecture used; the target is still to create a ML model that is able to accurately predict the ROP/"TRIP"/"PROBLM" from the drilling records.

### 4.3 Results and discussions

4.3.1 Qualification of BERT

The *bert-base-uncased* was able to achieve an F-1 score of 0.9 on both training and validation dataset, indicating the capabilities to read and understand textual information included in the drilling records. The results show that the BERT model was also able to work with long input (up to 512 tokens) without the problem of "forgetting". It could be concluded from the results that BERT is strong enough to use in latter modeling parts.

4.3.1 Modeling with Natural Language Processing

Direct comparisons between Figure 10 and Figure 18, Figure 11 and Figure 19, Figure 12 and Figure 20, and Figure 13 and Figure 21 show that adding textual information provides only small improvements in the quality of ROP predictions. This contradicts the hypothesis from the end of Section 3 that adding lithological information would improve the quality of ROP predictions. Even with drillers' remarks and lithological information proxies from the mud-shakers, it is not possible to improve the ROP results enough to make the models usable in production.

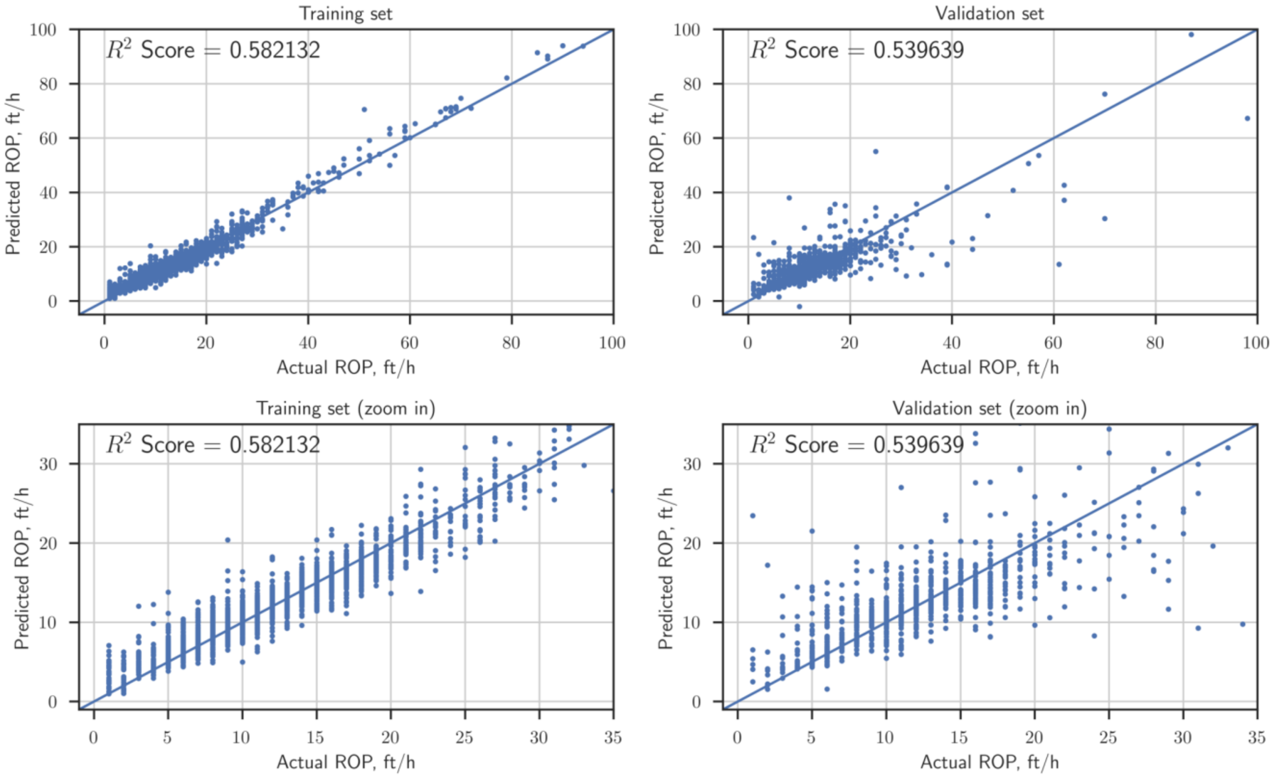


**Figure 18: Correlation plot for ROP modeling with textual information, random train/validation**

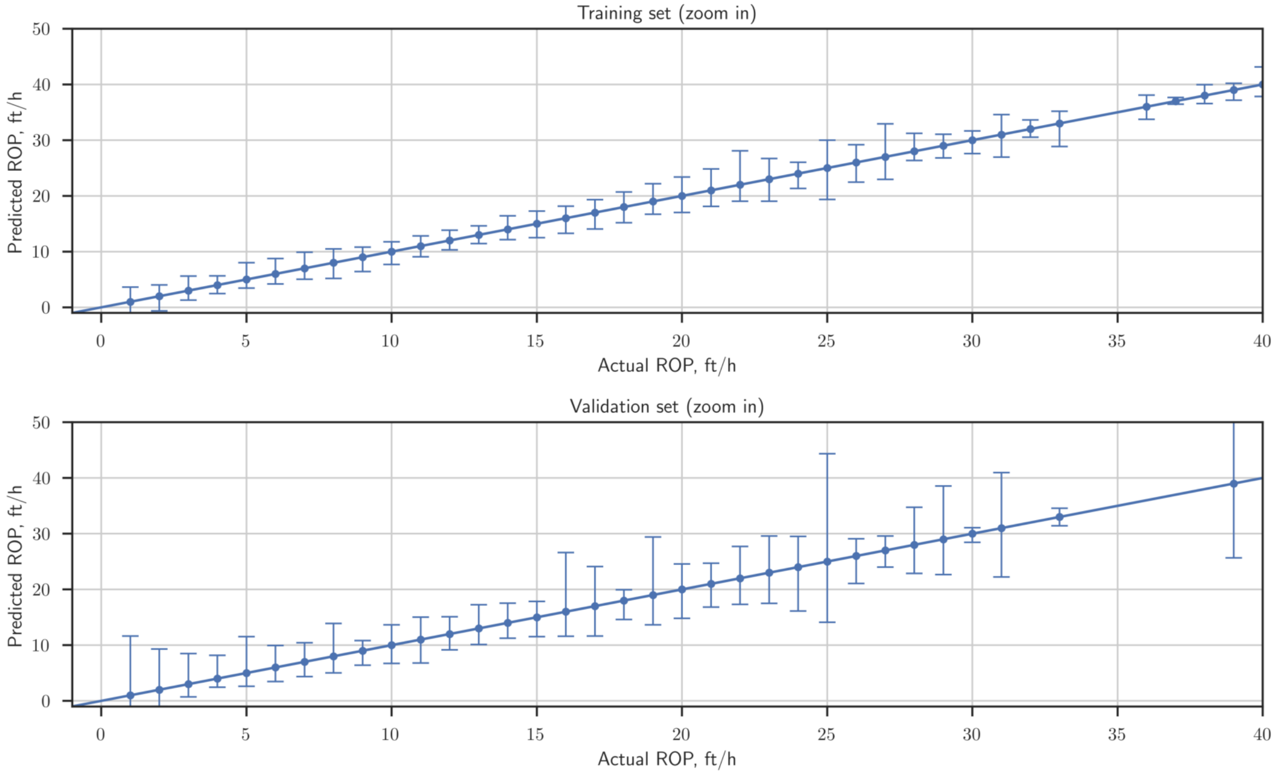


**Figure 19: 95% confidence interval plot for ROP modeling with textual information, random**

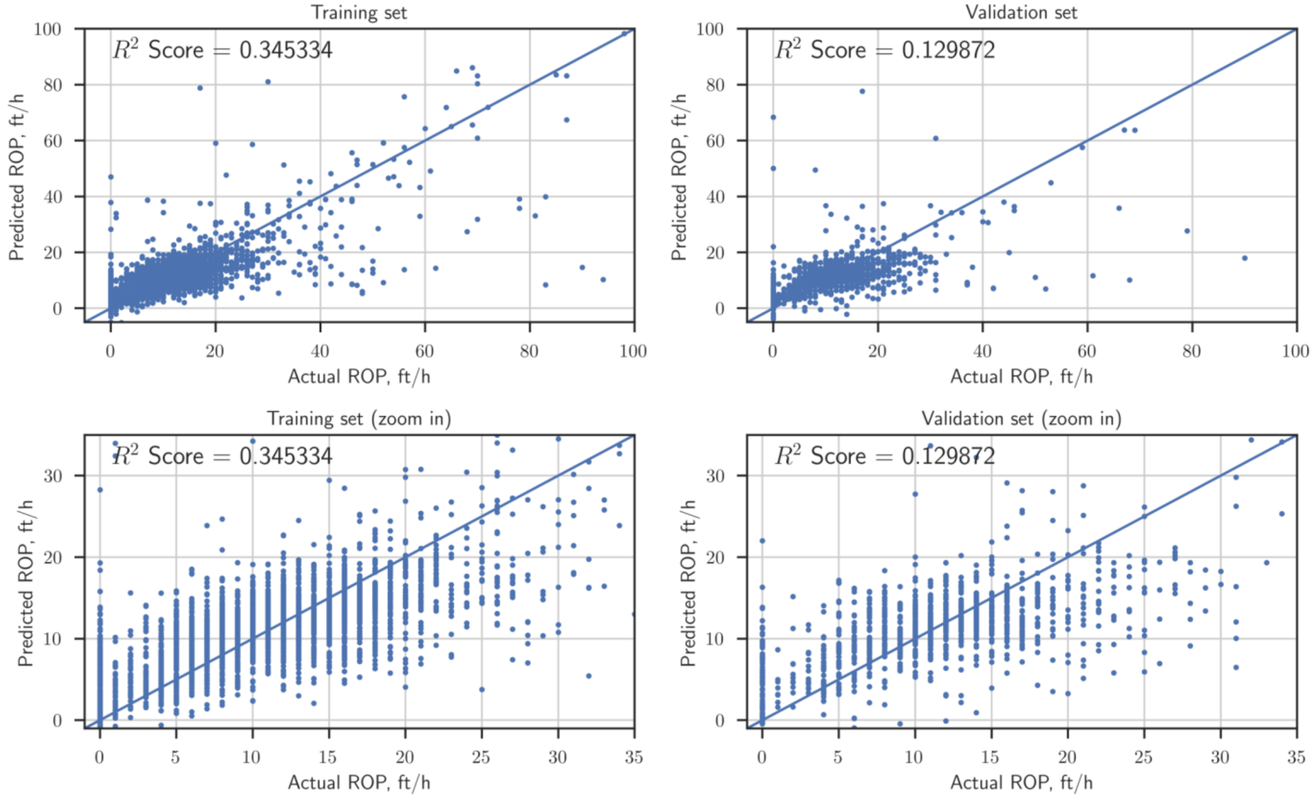


**Figure 20: Correlation plot for ROP modeling with textual information, well-by-well train/validation splitting**

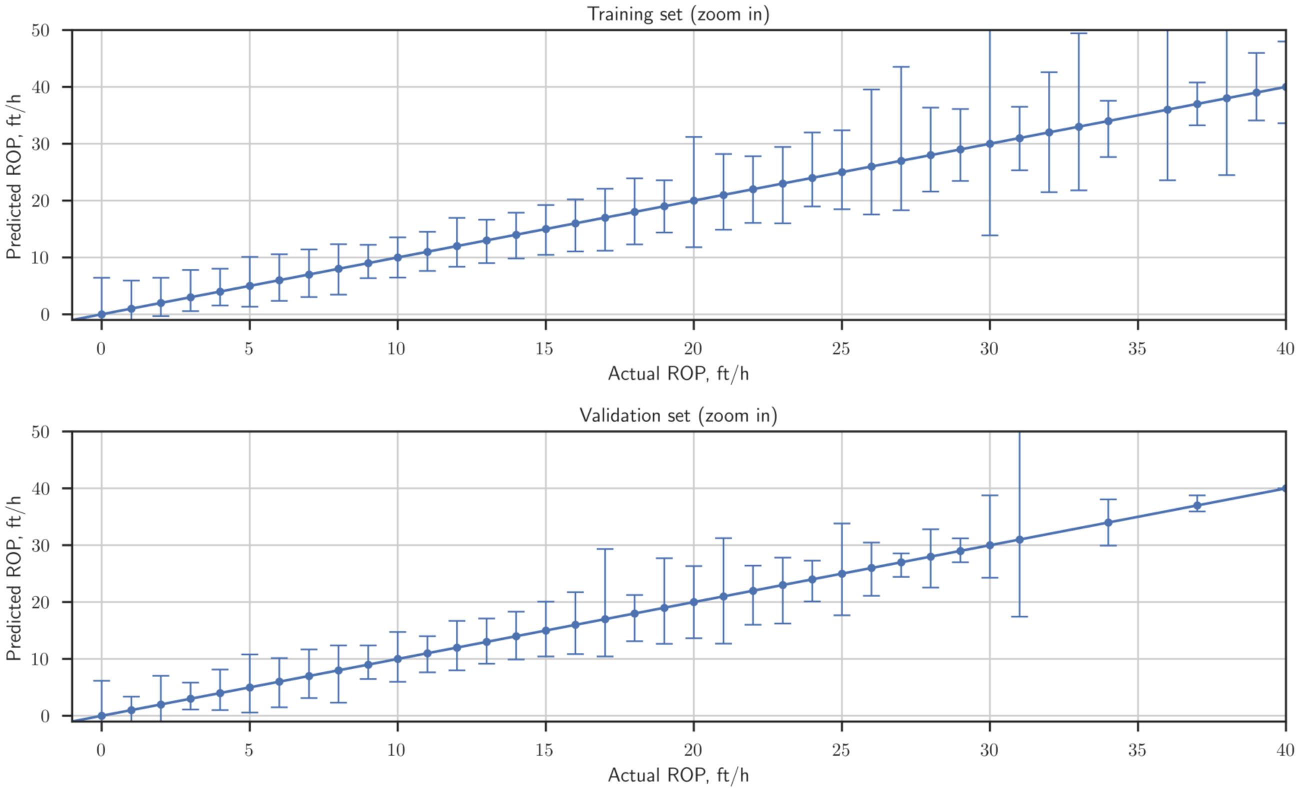


**Figure 21: 95% confidence interval plot for ROP modeling with textual information, well-by-well train/validation splitting**

However, comparing Figure 14 to Figure 22 and Figure 23 provides a completely different picture when using BERT. Figure 22 shows that using BERT also provides small improvements in tripping prediction accuracy. However, Figure 23 shows that BERT helps bring significant improvements in problem prediction accuracy. Without using BERT, the accuracy of predictions if problems do indeed happen is only 50%, which is no better than a random guess. However, with BERT, the ratio of True positive to False positive improved to nearly 2:1. This results in a model that is accurate enough to use in production, which can give drillers ample preparation for possible future problems.

The results in Figure 23 can be explained by the fact that the drillers tend to make remarks/comments about unusual observations encountered when drilling. A future problem will likely correlate to unusual observations in the past (however the opposite is not true). BERT is able to pick out those observations, and in conjunction with numerical drilling records, DPDBN can give accurate forecast about possible future problems.

Another observation is that by adding outputs from BERT, the prediction quality will improve in most cases. This shows that DPDBN is capable of processing very large inputs, and only pick out the most relevant features while ignoring the rest, which is a vital property when processing data like the inputs with BERT which are composed of more than one thousand features.

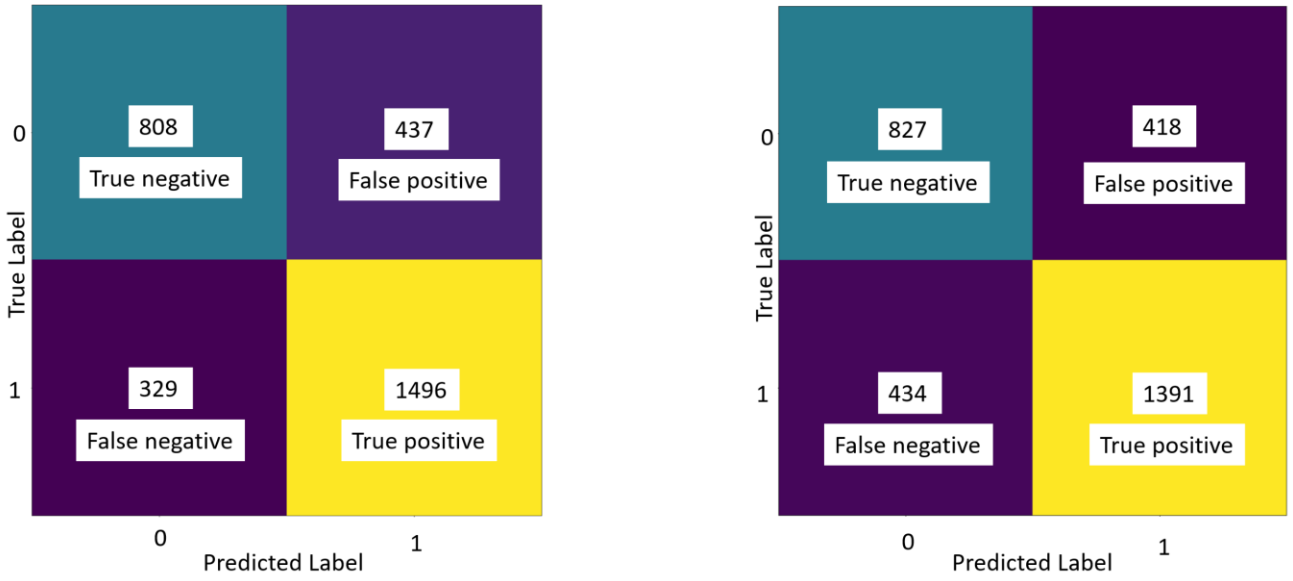


**Figure 22: Confusion matrix for tripping predictions: left uses textual information, right do not. One indicates tripping did happen, zero indicates tripping did not happen.**

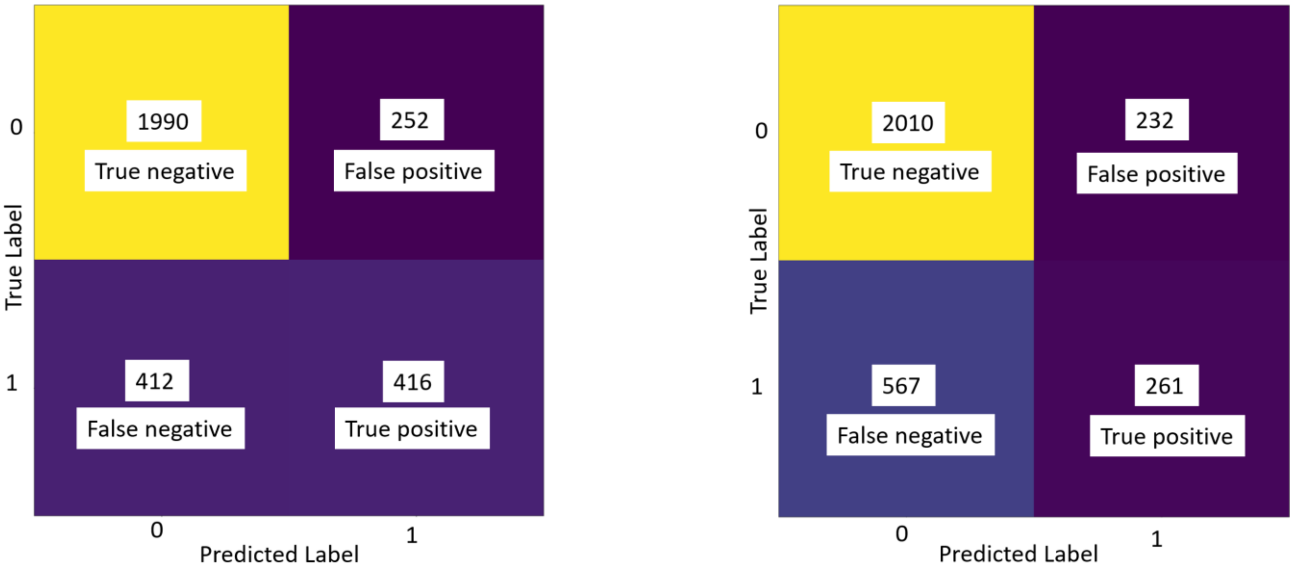


**Figure 23: Confusion matrix for problem predictions: left uses textual information, right do not. One indicates problem did happen, zero indicates problem did not happen.**

## 5. CONCLUSION

Although this study failed to develop an accurate ROP predicting deep neural network model, it outlines in detail the procedures on how to process, develop, and train a deep neural network model on geothermal drilling records.

Due to the fact that the dataset used in this study is daily-averaged, it is hard to pinpoint the reason for the low performance of ROP prediction. As there many different drilling operations throughout a single drilling day, daily-averaged records cannot capture all of these operations. If higher resolution data were available, then better ROP predictions may be achievable.

This study also shows that using Bidirectional Encoder Representations from Transformers (BERT) can effectively process and encode textual information into a form that can be incorporated with normal numerical records. This enables the use of textual data in drilling optimization without costly manual preprocessing. If a sufficiently powerful neural network is used in modeling, then the encoded textual information will provide benefits when used with conventional numerical data.

## ACKNOWLEDGEMENT

This research was part of the EDGE project "Toward Drilling the Perfect Geothermal Well: An International Research Coordination Network for Geothermal Drilling Optimization Supported by Deep Machine Learning and Cloud Based Data Aggregation", supported by the US Department of Energy under contract DE-EE0008793. The authors are grateful for the financial support of the work, as well as the many fruitful discussions with other members of the EDGE team.

## REFERENCES

Carbonari, R., Ton, D., Bonnneville, A., Daniel, B., Claudouhos, T., Gearrison, G., Horne, R., Petty, S., Rallo, R., Schultz, A., Sorlie, C.F., Thorbjornsson, I.O., Uddenberg, M., and Weydt, L.: First Year Report of EDGE Project: an International Research Coordination Network for Geothermal Drilling Optimization Supported by Deep Machine Learning and Cloud Based Data Aggregation, 46th Workshop on Geothermal Reservoir Engineering, Stanford University, 2021

Maaten, L., and Hinton, G.: Visualizing data using t-SNE, Journal of Machine Learning Research, 9(86):2579–2605, 2008

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., et al.: Language models are few-shot learners, arXiv, cs.CL/2005.14165, 2020

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach, arXiv, cs.CL/1907.11692, 2019

Devlin, J., Chang, M., Lee, K., and Toutanov, K.: BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv, cs.CL/1810.04805, 2019