energies

MDPI

Article

Applying Machine Learning to Predict the Rate of Penetration for Geothermal Drilling Located in the Utah FORGE Site

Mohamed Arbi Ben Aoun [1,2,*] and Tamás Madarász [2]

[1] Department of Civil, Geological and Mining Engineering, Polytechnique Montréal, 2500 Chemin de Polytechnique, Montréal, QC H3T 1J4, Canada
[2] Institute of Environmental Management, University of Miskolc, 3515 Miskolc-Egyetemváros, Hungary; tamas.madarasz@uni-miskolc.hu
* Correspondence: mohamed-arbi.ben-aoun@polymtl.ca

Abstract: Well planning for every drilling project includes cost estimation. Maximizing the rate of penetration (ROP) reduces the time required for drilling, resulting in reducing the expenses required for the drilling budget. The empirical formulas developed to predict ROP have limited field applications. Since real-time drilling data acquisition and computing technologies have improved over the years, we implemented the data-driven approach for this purpose. We investigated the potential of machine learning and deep learning algorithms to predict the nonlinear behavior of the ROP. The well was drilled to confirm the geothermal reservoir characteristics for the FORGE site. After cleaning and preprocessing the data, we selected two models and optimized their hyperparameters. According to our findings, the random forest regressor and the artificial neural network predicted the behavior of our field ROP with a maximum absolute mean error of 3.98, corresponding to 19% of the ROP's standard deviation. A tool was created to assist engineers in selecting the best drilling parameters that increase the ROP for future drilling tasks. The tool can be validated with an existing well from the same field to demonstrate its capability as an ROP predictive model.

Keywords: rate of penetration (ROP); predictive modeling; geothermal energy; machine learning; deep learning; random forests; artificial neural network; python programming

Citation: Ben Aoun, M.A.; Madarász, T. Applying Machine Learning to Predict the Rate of Penetration for Geothermal Drilling Located in the Utah FORGE Site. *Energies* **2022**, *15*, 4288. https://doi.org/10.3390/en15124288

Academic Editors: János Szanyi, Ladislaus Rybach and Renato Somma

Received: 6 May 2022
Accepted: 6 June 2022
Published: 11 June 2022

1. Introduction

Geothermal reservoirs are made of fractures of hard volcanic formations in fields with a gradient between 7 °C and 16 °C per 100 m. These are high-temperature (HT) drilling environments. Thus, the well-planning for geothermal conditions requires additional material selection and considerations. In addition, deep geothermal fluids are sub-hydrostatic. Their hydrostatic pressure is lower than the hydrostatic pressure of the drilling mud. As a result, drilling in geothermal reservoirs is considered challenging [1].

Deep geothermal drilling accounts for more than 30% of geothermal project costs [2]. Thus, cost reduction is needed for this task. One solution is increasing the speed of drilling, also called rate of penetration (ROP). The latter is a performance metric for drilling operation and is dependent on several factors. Weight on bit (WOB), rotating speed (RPM), and the flow rate are three operational drilling factors that can be modified at the surface to affect ROP. Formation proprieties such as rock strength, abrasiveness, heterogeneity, pore pressure, and permeability also affect ROP. However, current technology does not permit the control of rock parameters [3].

A study by [4] established that the ROP of a roller cone bit, under perfect hole cleaning conditions, is proportional to the rotary speed and the bit weight squared, and inversely proportional to the bit diameter squared. However, these conditions are not necessarily met [5]. Thus, poor hole cleaning scenarios cause an increase in ROP due to the increasing WOB of the hook load adjustment. Bourgoyne et al. [6] illustrated a relationship between the ROP and drilling parameters similar to Maurer's, but added that a minimum WOB is

needed to start drilling. Dupriest and Koederitz [7] elaborated the concept of mechanical specific energy (MSE). The latter is the amount of mechanical work needed to excavate a unit volume of a rock. It quantifies the relationship between input energy and ROP. The relationship between the ROP and drilling parameters is linear when the MSE is constant. An increase in MSE means that the system is foundering, and a disproportionate amount of energy is used for the given ROP. Young [8] demonstrated that ROP is influenced by the pressure gradient ahead of the bit. The latter cannot be measured in the field. Bourgoyne et al. [9] developed a comprehensive ROP analytical model that includes variables such as the compressive strength of the formation. The confined compressive strength is a crucial parameter for drilling optimization. However, it is only measured in the lab [10]. Nevertheless, drilling optimization requires models quantifying the correlation between all operational factors and the ROP.

Machine learning is a major subfield in computational intelligence; it extracts information from raw data. Machine learning has a variety of applications in the information technology sector, including speech recognition, object recognition in computer vision, robotics, computer games, etc. Machine learning models have been extensively used for ROP prediction. Models including multilayer perceptron neural network (MLPNN), radial basis function neural network (RBFNN), and support vector regression (SVR) have been implemented successfully for this task [11]. A recent study by [12] demonstrated an implementation of ensemble learning methods (e.g., random forest regressor) for ROP prediction of a deep well crossing multiple lithologies. The authors pointed out that the random forest outperformed the ANN, with an average absolute error percentage of 7.8%.

Comprehensive ROP modeling requires additional formation variables, such as pressure gradient compressive strength of the formation. The latter is obtained from well log data or from lab measurements of retrieved core samples. These procedures are expensive and time-consuming. Consequently, it is necessary to build a new approach for ROP prediction. Previous studies established by [13] on drilling optimization of the same drilling dataset of the Utah FORGE site. The authors applied an optimization method called the differential evolution algorithm (DEA). It predicts the unconfined compressive strength (UCS) for the drilled feet to simulate the ROP for the next drilling feet, based on the previous UCS value.

The successful application of machine learning is the main motivation to choose these methods. The latter require continuous data collection during drilling. It is feasible, thanks to advancements in drilling data acquisition and computing technologies. In addition, no previous research has applied a data driven approach (e.g., machine learning and deep learning) for predicting the ROP of the 58–32 well. This research outlines the procedure of transforming the abundance of raw data into useful information, also called data mining [14]. Our contribution is building a predictive model explaining the behavior of the ROP based on the analysis of the patterns and extracting correlations from our drilling data. The ROP predictive model takes as input the drilling parameters and other factors related to drilling. We share the developed code for public access as follows: https://github.com/Arbi-ben-aoun/Drilling-rate-of-penetration-prediction (accessed on 1 May 2022).

In the following section, we will present the theory underlying the best-performing machine learning algorithm used.

2. Related Work

2.1. Decision Trees

The random forest method is based on a popular method developed by Leo Breiman in 1984 [15]. The regression and classification tree (CART) divides the predictor space into several regions [16]. Regression trees delineate the regions of predictions from a given training dataset. They assign the mean of the corresponding sample to each specified region. The term decision tree originates from the rules of splitting, which are summarized into a tree. Even though decision trees have high interpretability compared to other supervised

learning algorithms, they produce less accurate results, due to the major problem of overfitting. The solution to this problem is ensemble learning, which is the central idea behind the random forest algorithm.

For simplicity, consider a two-dimensional feature space and a binary splitting in the following example. First, the feature space is split into two regions, and the mean of the samples for each region is calculated. The choice of a split variable and the split point is dependent on the best fit. Next, the two regions are split further. This procedure will continue until a certain stopping rule is applied.

Figure 1 shows that the first split is at $X_1 = t_1$. Next, the region of $X_1 \leq t_1$ is split at $X_2 = t_2$. Then, the region $X_1 > t_1$ is split at $X_1 = t_3$. Finally, the region $X_1 > t_3$ is split at $X_2 = t_4$.

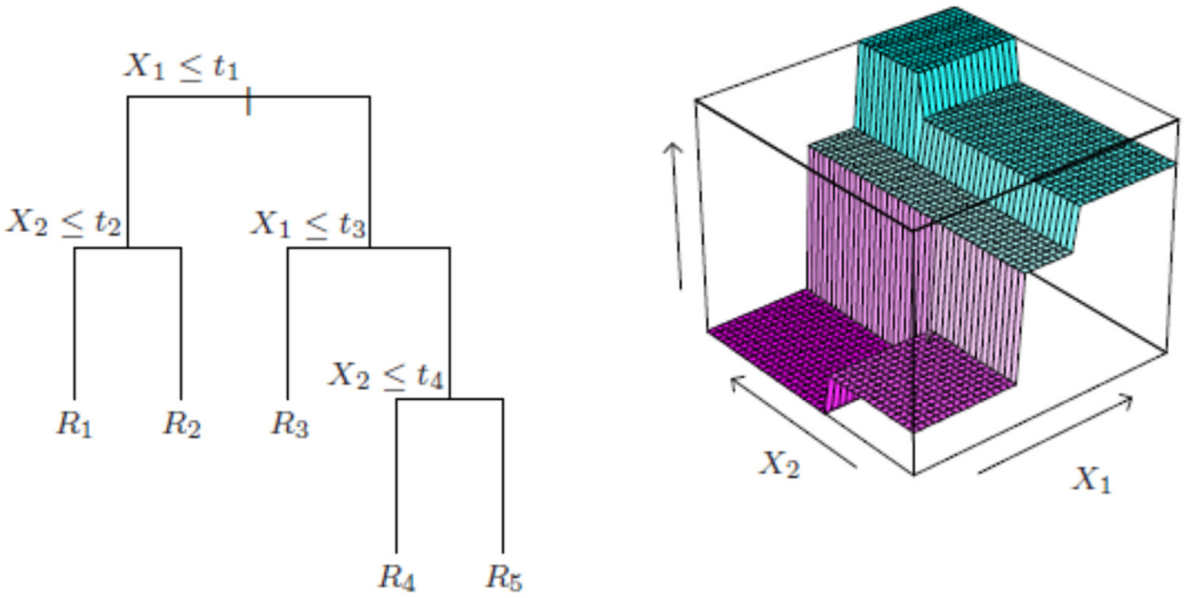

Figure 1. Partition of feature space in a regression tree [17].

We obtain five regions in the end. The following regression tree model predicts an output $\hat{f}(X)$, with a constant C_m representing the mean for each region R_m, considering ***I()*** as an indicator function that returns 1 if its argument is true and 0 if the argument is false [15]. Equation (1) is the prediction output after splitting.

$$\hat{f}(X) = \sum_{m=1}^{5} C_m * I * \{(X_1, X_2) \in R_m\} \quad (1)$$

Equations (2) and (3) consider a binary split variable j with a split point s to find the best split point, resulting in two half-planes $R_1(j,s)$ and $R_2(j,s)$.

$$R_1(j, s) = \{X \mid X_j < s\} \quad (2)$$

$$R_2(j, s) = \{X \mid X_j > s\} \quad (3)$$

Equation (4) defines $\hat{C}_m$ as tthe average of the samples of the corresponding regions after the split:

$$\hat{C}_m = \text{average}\ (y_i \mid x_i \in R_m) \quad (4)$$

This leads to an optimization problem that searches the splitting variable j and the split point s for solving the following minimization problem [18].

$$Min_{j,s}\ [\ Min\ C_1 \sum_{x_i \in R_{1(j,s)}} (y_i - c_1)^2 + Min\ C_2 \sum_{x_i \in R_{2(j,s)}} (y_i - c_2)^2] \quad (5)$$

The objective is to minimize the mean squared error of each tree node. When it comes to stopping the tree growth, deep splits will produce good results on the train set, but will be inaccurate on the unseen test set. This problem is called overfitting and must be avoided in supervised learning tasks. A strategy to prevent deep trees is first building a large tree T_0. Then, this large tree is pruned (by cutting away its branches) for the flowing cost-complexity pruning [15].

$$C\alpha\ (T)=\sum_{m=1}^{|T|}\sum_{x_i\in R_i}(y_i-c_m)^2+\alpha\,|T| \tag{6}$$

In Equation (6), $|T|$ is the number of the terminal nodes of the subtree T for the subtree $T \subset T_0$.

This implies another optimization problem to find the variable $\boldsymbol{\alpha}$ that minimizes $C\alpha\ (T)$.

2.2. Bagging

The overfitting problem is due to the high variance. The concept of bagging, or aggregation, is introduced for a given training set with size m [19], considering a random sample of size $k < m$ for n training sets with replacement, known as bootstrap sampling. The outcome of the latter is n independent samples. Equation (7) explains the variance of the bootstrap samples, with variance σ^2 as the variance for each sample.

$$\sigma_n^2=\frac{\sigma^2}{n} \tag{7}$$

When bagging is applied in decision trees, the average of n set of bootstrap samples reduces the variance and predicts more accurate results. n numbers of regression trees are built on the n separated bootstrapped samples [20]. The n averaged predictions are explained by Equation (8).

$$\hat{f}_{\mathbf{avg}}(X)=\frac{1}{N}\sum_{i}^{n}\hat{f}_b(X)\ \#(8) \tag{8}$$

where $\hat{f}_b$ is the prediction of a single decision tree built on a single bootstrap sample. Figure 2 summarizes the bagging procedure.

2.3. Random Forest Regressor

The random forest method is an improvement of bagged decision trees by adding a tweak that decreases the correlation of our decision trees. m number of predictors (independent input variables) are chosen from a total of p predictors for each tree. Equation (9) is an estimation of m for regression tasks [17,19].

$$m\ \approx \sqrt{p} \tag{9}$$

Additionally, the random forest regression will have other advantages in feature selection, such as variable importance. As mentioned previously, the random forest method selects a subset of predictors during bagging and records the loss for each tree. Intuitively, the predictors that decrease the loss are more important to our predictions, and the predictors that do not affect the accuracy are considered the least important [20]. However, domain knowledge is also essential for selecting the predictors. Analytical ROP models also highlight the parameters required for prediction.

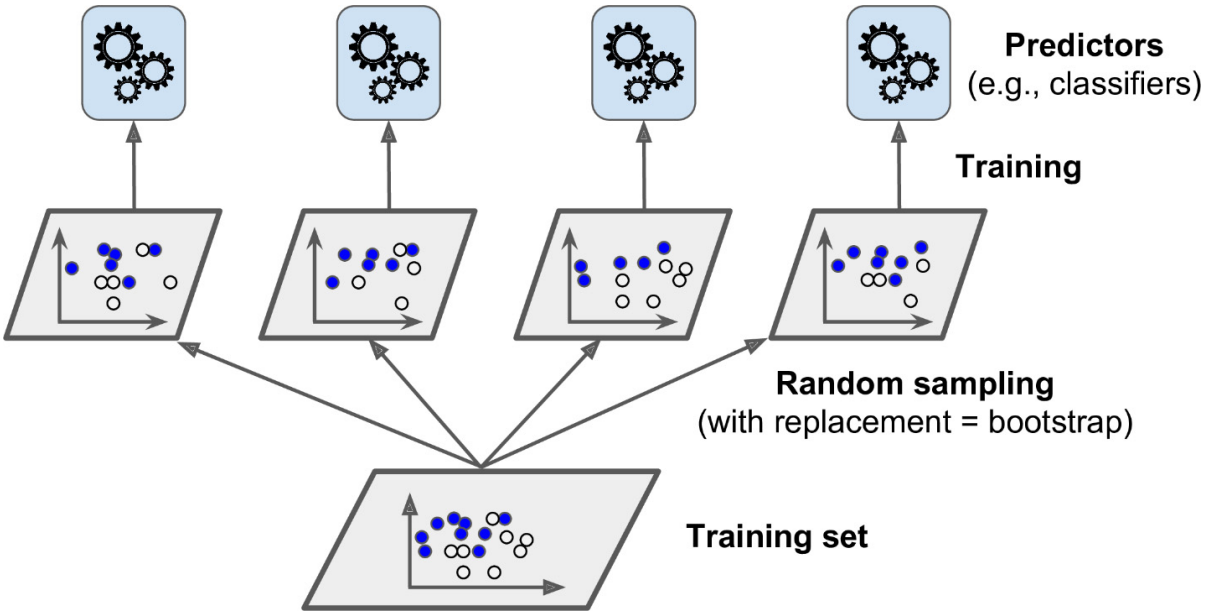

Figure 2. Bagging of the training set [21].

3. Materials

3.1. Study Area

The U.S. Department of Energy's Frontier Observatory for Research in Geothermal Energy (FORGE) is a field laboratory that offers opportunities to research, develop, and test new technologies for enhanced geothermal systems. In 2018, the U.S. Department of Energy selected the location of south-central Utah for establishing their site. Since the 1970s, several geological and geophysical studies had been directed in this region to develop geothermal resources at Roosevelt Hot Springs. The FORGE project has been realized in three major phases. Phase one involved desk studies of existing data from five sites within the United States. Phase 2 involved drilling the well 58–32, with a total depth of 2290 m GL, and a bottom-hole temperature of 199 °C. The latter reached low permeability crystalline rocks at 961 m GL. The site is located in south-central Utah, U.S.A. The Utah Frontier Observatory for Research in Geothermal Energy (FORGE) site is located 350 km south of Salt Lake City and 16 km north-northeast of Milford, Utah [22]. The site area covers 5 km^2 and is located on the west sloping part of an alluvial fan in the Milford Valley (Figure 3).

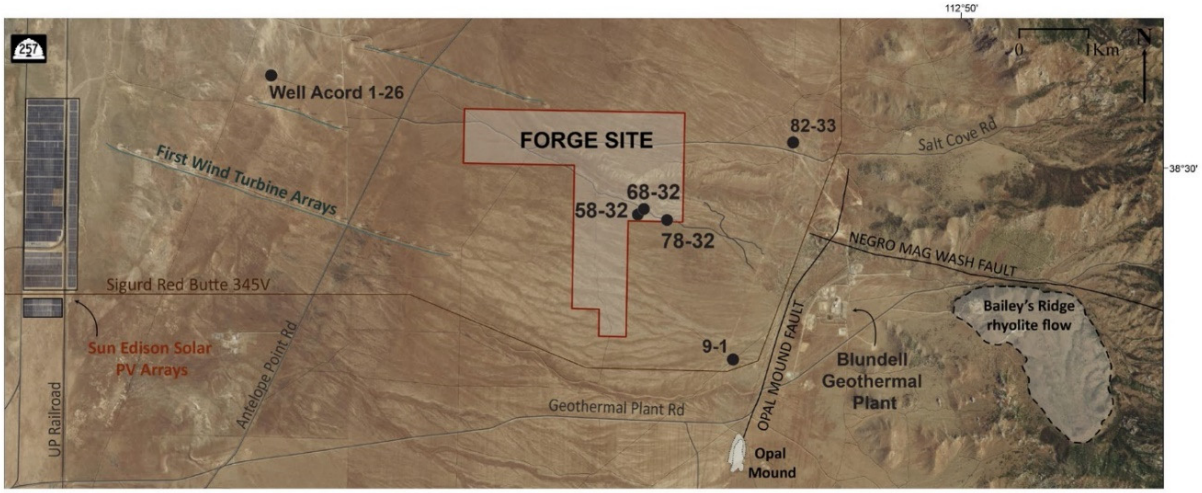

Figure 3. Forge site location [23].

The main activity of phase 2B of this project is drilling a deep vertical well 58–32 with a depth of 2298 m. The well determines whether the geothermal reservoir characteristics meet the FORGE site's requirements for enhanced geothermal system development. Field and laboratory measurements confirmed that the reservoir is hosted by a hot crystalline low permeability granitic rock with a temperature greater than 175 °C. The success in proving this result was due to a synthesis of a large amount of geoscientific data collected over a 40-year period [24,25]. Figure 4 shows the simplified geology of the FORGE site.

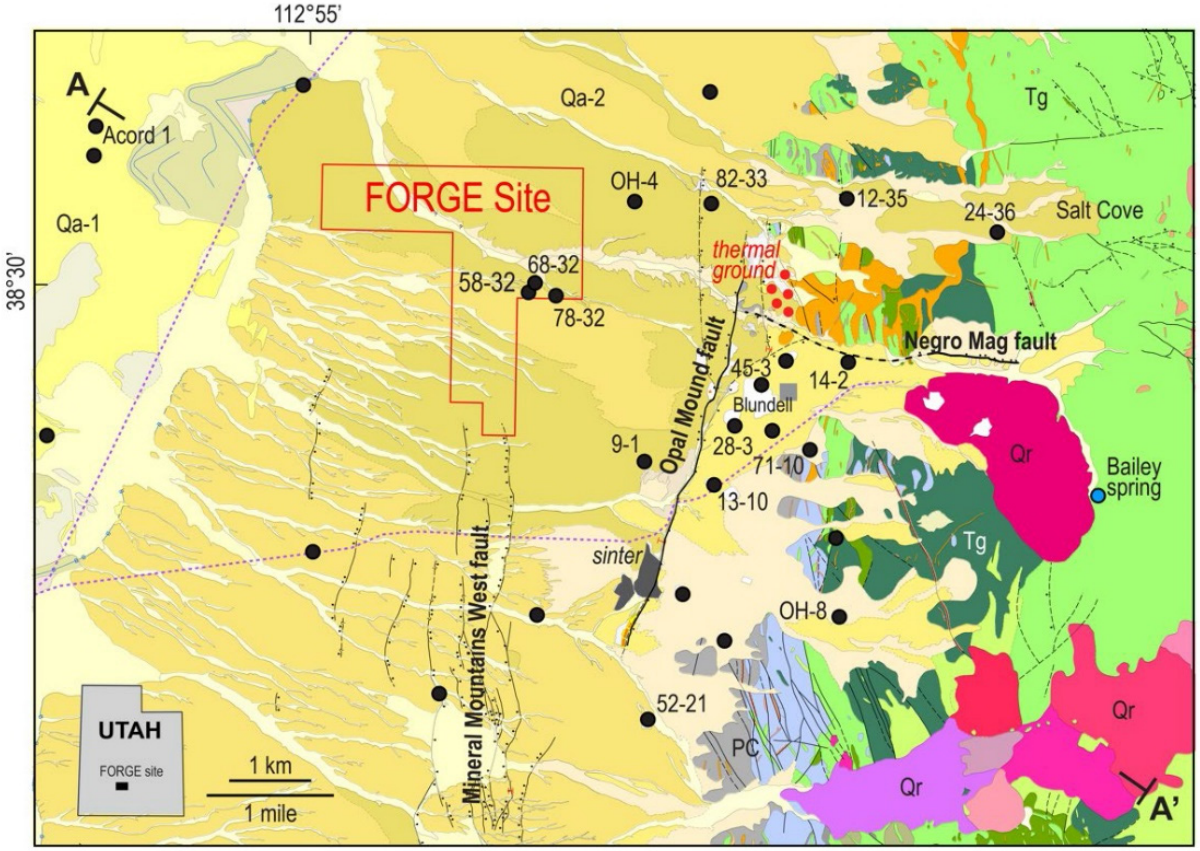

Figure 4. Geological map of the FORGE Utah site [25] based on the compilation from new field observations, well data, and previous work [26,27] Abbreviations: Qa-1 = Lake Bonneville silts and sands; Qa-2 = alluvial fan deposits; Qr = quaternary rhyolite lava and pyroclastic deposits; Tg = tertiary granitoid; PC = Precambrian gneiss; black filled circles = wells.

The Utah forge reservoir has a very low porosity (<0.1) and a low permeability (0.1 to 80 micro-darcies μD). Due to these factors, there is no evidence of an existing hydrothermal flow. The only natural hydrothermal system is hosted in the shallow aquifer of the basin fill alluvium overlying the granitoid layer. The latter is the outflow from the Roosevelt Hot Springs system that lies more than 3 km to the east [25]. This geothermal water is abundant, but proved to be non-potable. However, it still fulfills the water requirement for future injection-production testing at the Utah FORGE site. Additionally, the Opal Mound Fault, which serves as a no-flow lateral barrier, separates the natural hydrothermal system from the EGS reservoir. Figure 5 shows the hydrogeological setting of the site.

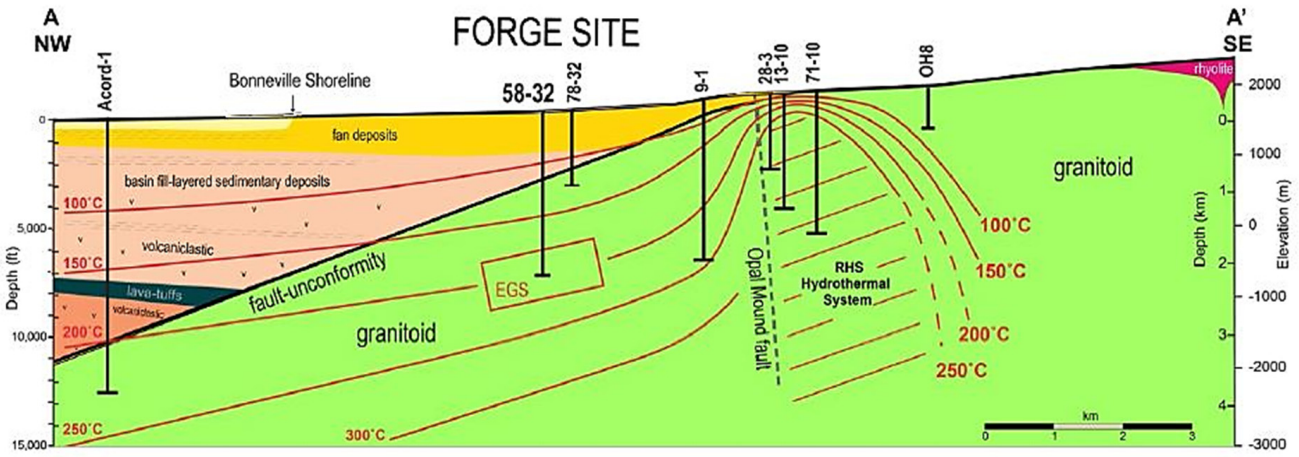

Figure 5. Cross section of the FORGE Utah site [25] based on the compilation from new field observations, well data, and previous work [26,27].

3.2. Drilling Dataset

Diagnostic drilling data (Pason log files) from Well 58–32 are collected using drill bits integrated with sensors, computers and an entire networking infrastructure to record the drilling data in real-time. The collected data contained useful information about the drilling rig's performance, such as the rate of penetration (ROP), and drilling parameters, such as weight on bit, temperature, pump pressure, etc. The drilling dataset of **Well 58–32** [28] was obtained and summarized in Table 1.

Table 1. Well 58–32 drilling dataset statistics.

	Count	Mean	std	Min	25%	50%	75%	Max
Depth (m)	7311	1168.864	654.5272	25.96	600.545	1173.99	1734.71	2296.94
ROP (m/h)	7311	12.80416	23.13962	0	3.47	5.48	13.5	907.62
Weight on Bit (kg)	7311	10,483.76	4135.825	0	8303.85	10,807.26	13,460.32	21,337.87
Temp Out (°C)	7311	52.2553	6.811023	28.93	46.74	51.59	58.05	66.5
Temp In (°C)	7311	47.95309	6.629486	29.44	42.695	47.34	52.7	63.51
Pit Total (m^3)	7311	37.6687	2.9034	27.17	35.7	37.86	39.68	44.5
Pump Press (KPa)	7311	8733.445	3382.374	137.49	4589.24	9877.5	11,512.44	1,5171.96
Hook Load (kg)	7311	36,864.21	12019.88	12,367.35	24,816.33	36,344.67	47,904.76	67,541.95
Surface Torque (KPa)	7311	903.1323	335.8324	0	806.715	967.44	1084.45	1887.23
Rotary Speed (rpm)	7311	54.94729	25.94765	0	38.09	50.38	75.965	271.58
Flow In (liters/min)	7311	2711.315	536.7113	0	2347.94	2650.58	3121.485	12,558.14
Flow Out %	7311	79.69283	11.9094	0.69	72.65	80.71	88.845	111.21
WH Pressure (KPa)	7311	−246.571	1535.307	−8493.47	20.13	40.96	56.95	120.04
H2S Floor	7311	−0.02737	0.042453	−0.1	−0.07	−0.01	0	0.78
H2S Cellar	7311	0.004303	0.025282	−0.08	−0.01	0	0.02	0.07
H2S Pits	7311	0.148833	0.11529	−0.06	0.06	0.14	0.22	0.72

This data is made publicly accessible under the license Attribution 4.0 International (Creative Commons BY 4.0). The file format of the drilling data is a comma-separated value (CSV). The dataset is clean from missing values, thanks to Utah FORGE's precise real-time drilling measurement.

4. Methods

Supervised learning is a type of machine learning algorithm that requires both features (input variables) and the desired output. The algorithm tries to train and adjust its parameters to produce the desired output from the training set. Then, the trained model is tested on unseen data, also known as the test set, for which we already know the output. Finally, the model predictions are compared with the known output values. These models are referred to as supervised learning algorithms because they have a "teacher" that provides supervision to the algorithms in the form of the known outputs for each example learned from the training data. Concerning the workflow for performing our data analysis and predictive modeling, the first step consists of the preprocessing of our data to prepare it for predictive modeling. The second step consists of the predictive modelling step. First, we train our model using the training set. Second, we validate it on an existing validation set to tune and optimize our model parameters. Lastly, the model is tested on unseen data [29]. The best model is selected according to the error. Figure 6 summarizes the workflow of our predictive modeling.

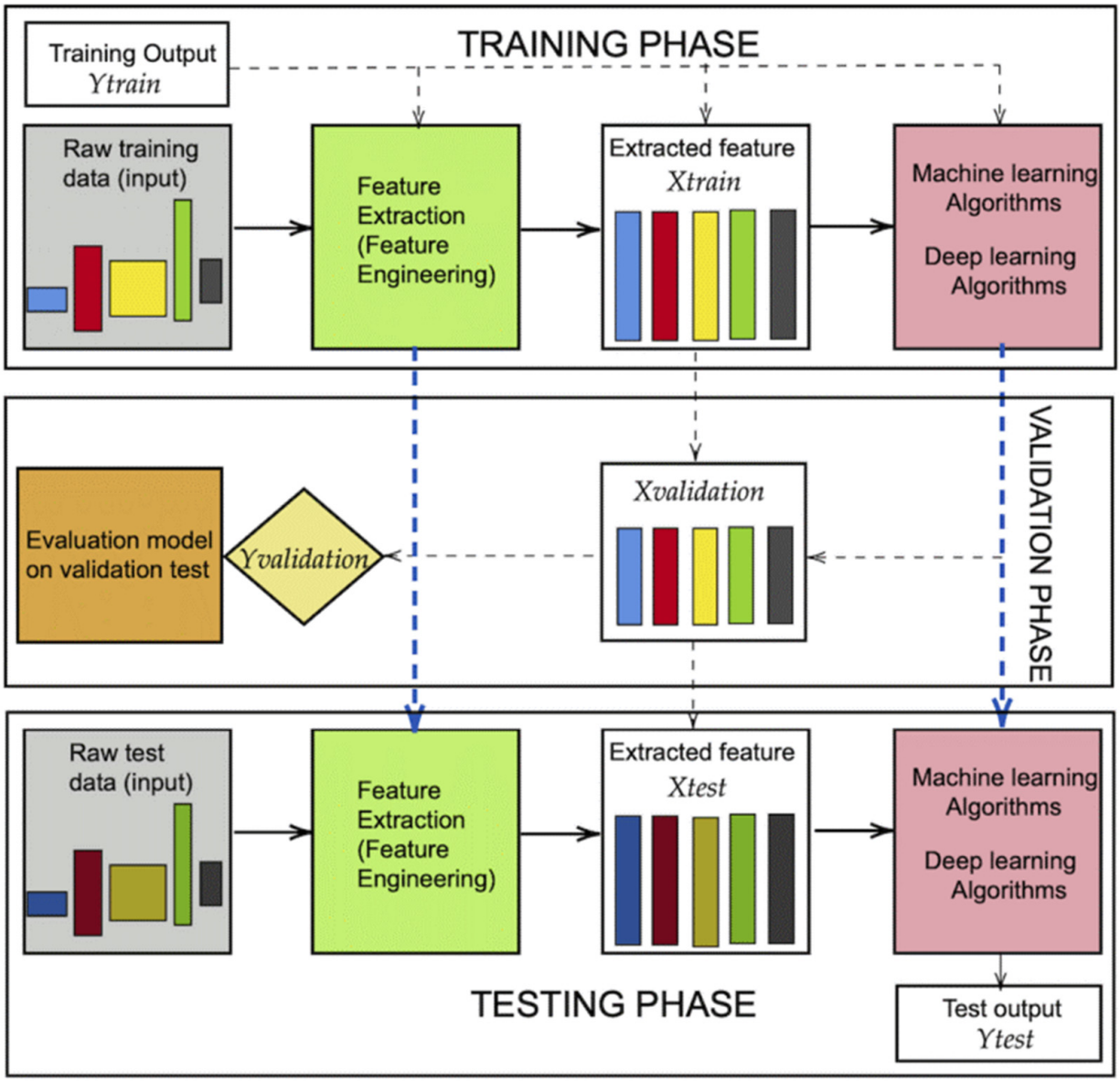

Figure 6. Workflow required for machine algorithms [30].

The main tool used was the open-source Jupyter computational notebook. It has an integrated development environment for Python version 3.9.0, which comes with the essential libraries needed for data analysis and machine learning included. The random forest regressor and the artificial neural network are the predictive models we applied for our dataset, and both are included in the Python libraries.

4.1. Data Preprocessing

Data preprocessing techniques are transformations applied to the training set to improve the performance of predictive models. Prediction is improved by transformations such as reducing data skewness [31] and removing outliers [32]. Feature selection is a simpler strategy that involves removing predictors based on their lack of information and is another effective technique for improving the performance of machine learning algorithms [33].

4.1.1. Feature Selection Based on Domain Knowledge

We chose the following features or predictors based on domain knowledge for drilling engineering:

Target variable Y: ROP (m/h).

Predictors X_i: Depth(m), weight on bit (kg), rotary speed (rpm), pump press (KPa), temp in (°C), flow in (L/min), and flow out %.

4.1.2. Correlation Measurement

Correlation is obtained using the Pearson correlation coefficient, which measures the linear relationship between two predictors. Table 2 summarizes the correlation results.

Table 2. Pearson correlation results.

Predictors	Pearson Correlation between ROP (m/h) and Predictors
Depth (m)	−0.508247
Weight on Bit (kg)	−0.523441
Rotary Speed (rpm)	0.28907
Pump Press (KPa)	−0.49319
Temp In (degC)	−0.221713
Flow In (liters/min)	0.481607
Flow Out %	−0.116068

4.1.3. Outlier Removal

A dataset can contain extreme values that are outside of the expected range and are unlike the other data. These outliers reduce the machine learning algorithm's ability to generalize. Removing these outliers improves the model's performance. There is no universal solution for outlier removal, but it is common is to rely on data visualization. We used both pair plot and boxplot for this purpose. Figure 7 shows the pair plot results

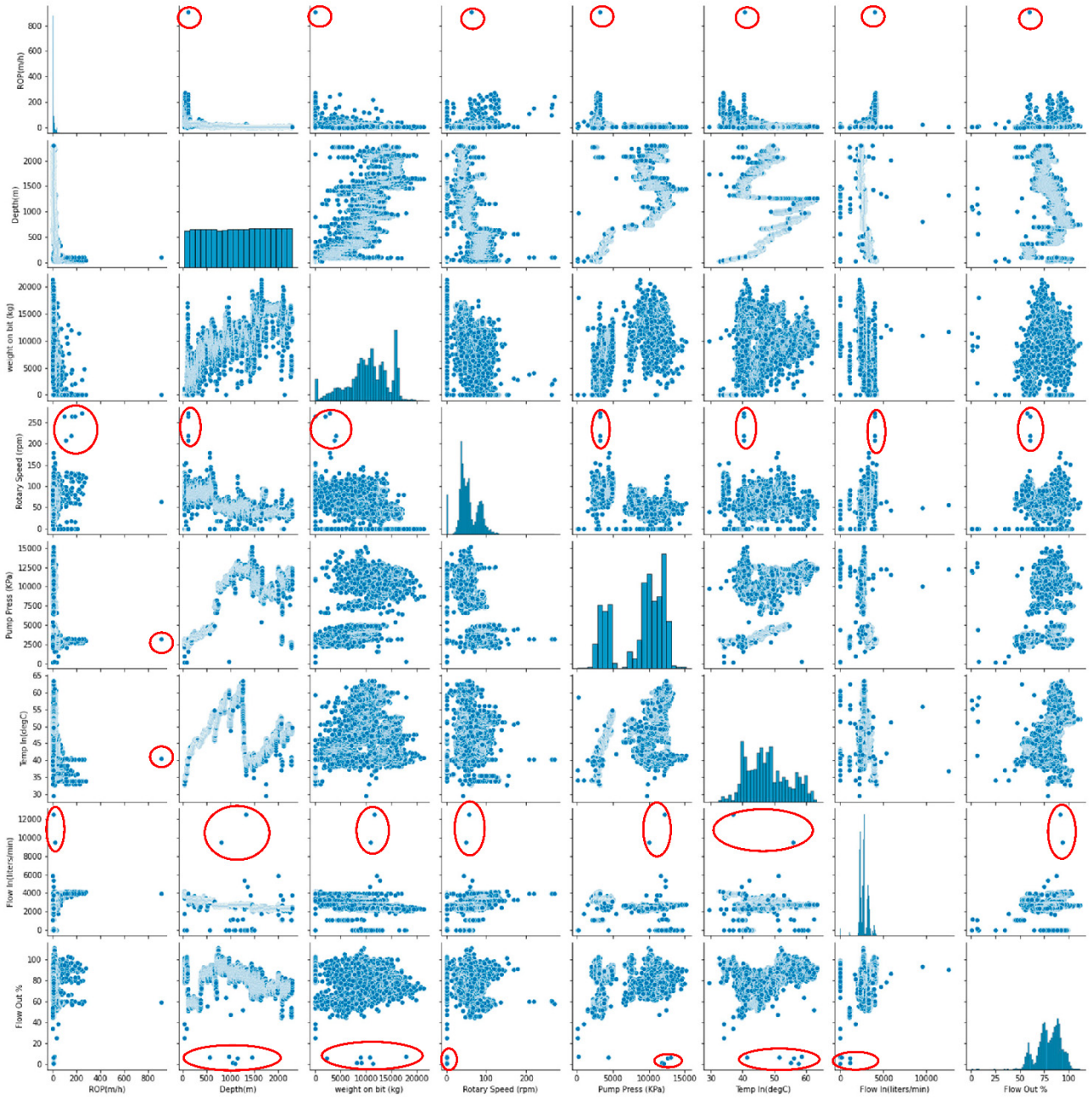

Figure 7. Pair plot of our drilling dataset with the encircled outliers.

The pair plot shows the distribution plot and the scatter plot associated with each pair of features. We found outliers in the ROP, rotary speed, flow in, and flow out features.

The previous features were examined in-depth using the boxplot. The ROP boxplot below clearly demonstrates that ROP above 800 (m/h) is an outlier (Figure 8).

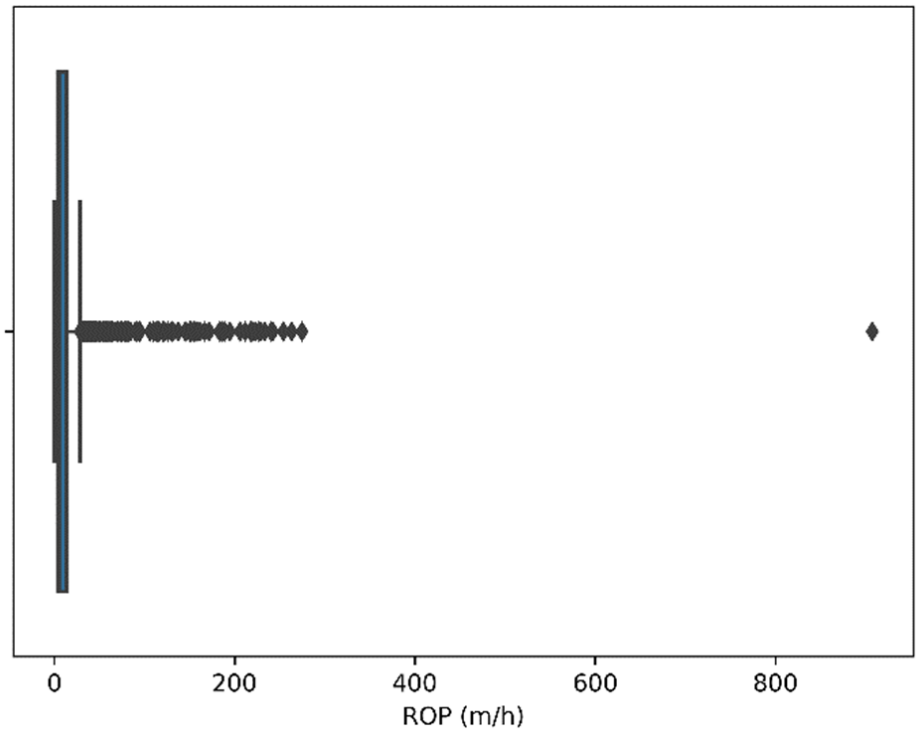

Figure 8. ROP boxplot.

Rotary speed above 200 (rpm) is considered an outlier (Figure 9).

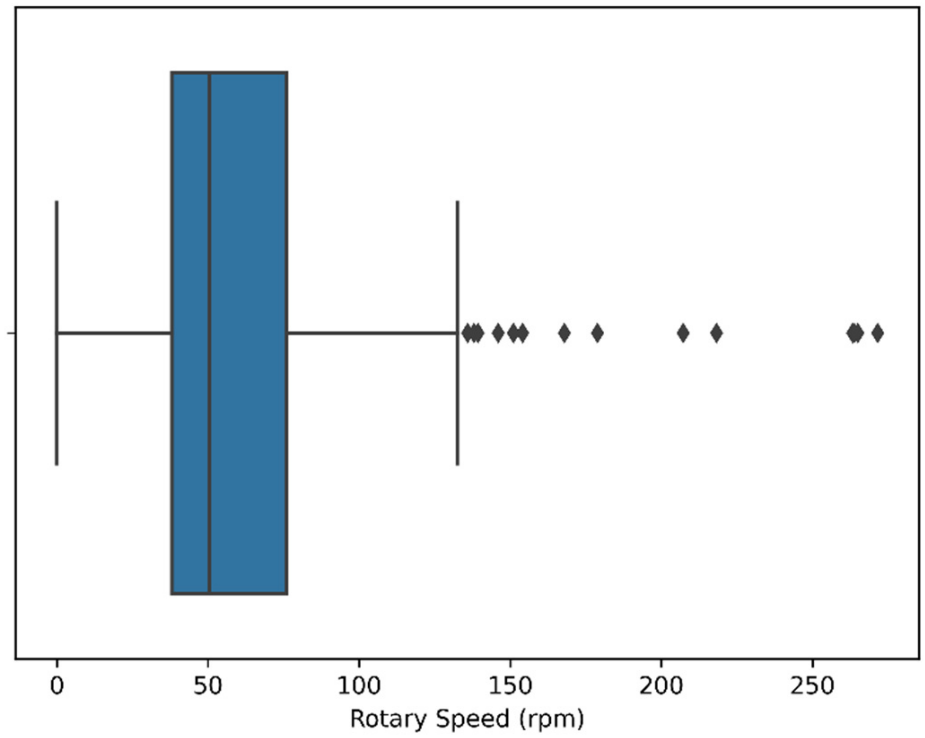

Figure 9. Rotary speed boxplot.

Flow in above 7000 (L/min) is considered an outlier (Figure 10).

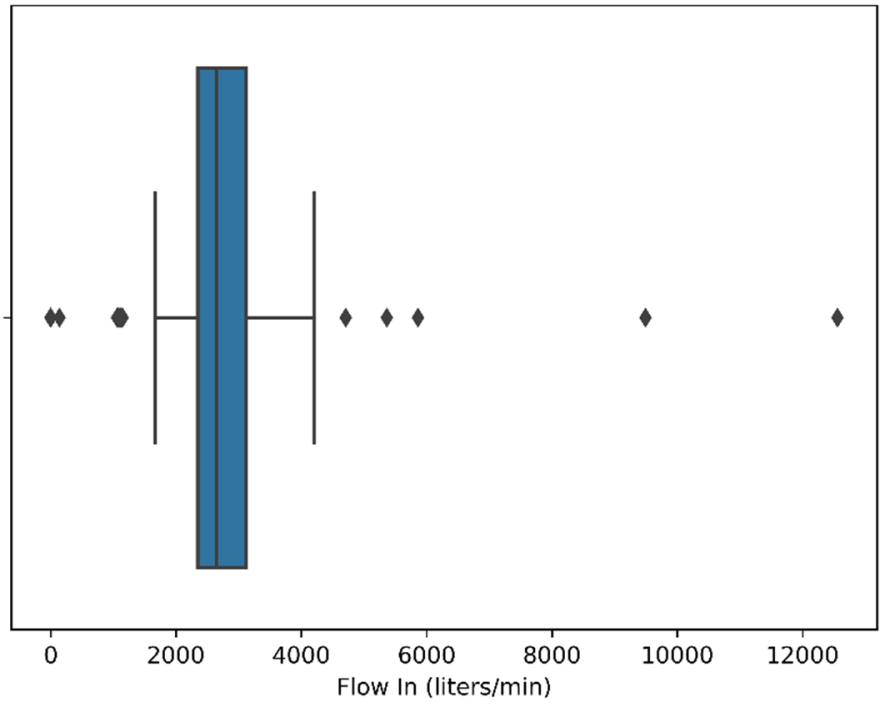

Figure 10. Flow In boxplot.

Finally, the flow out of less than 10% is considered an outlier (Figure 11).

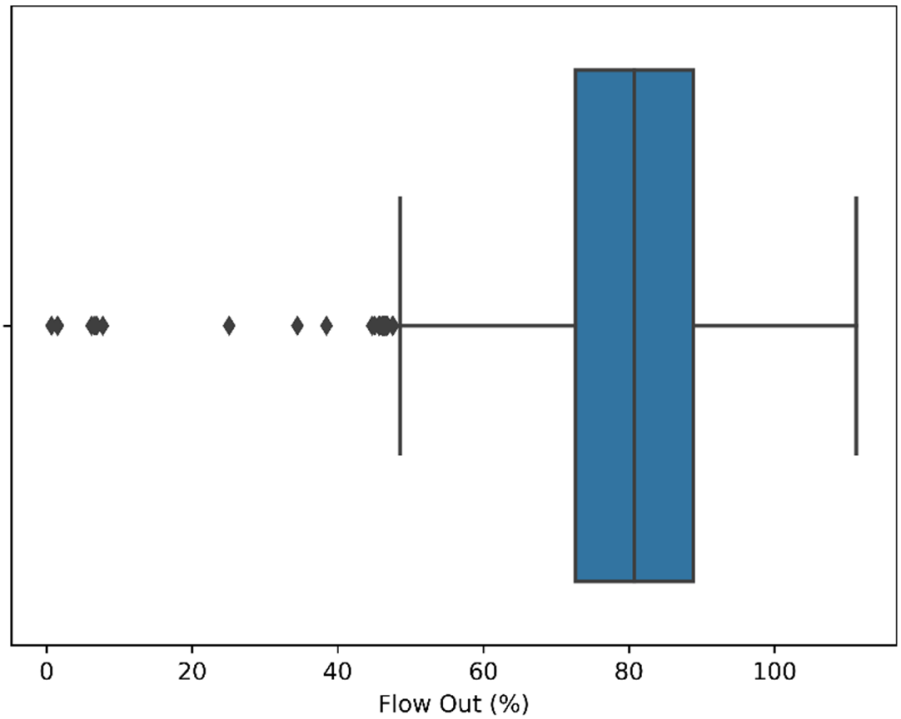

Figure 11. Flow Out boxplot.

4.1.4. Tree Based Feature Selection

As we recall from tree-based models, they indicate the importance of features [34]. Therefore, we train a random forest model, without any optimization, to identify the most important predictors in the random forest split. The feature importance between the target variable and the predictors is shown in Figure 12.

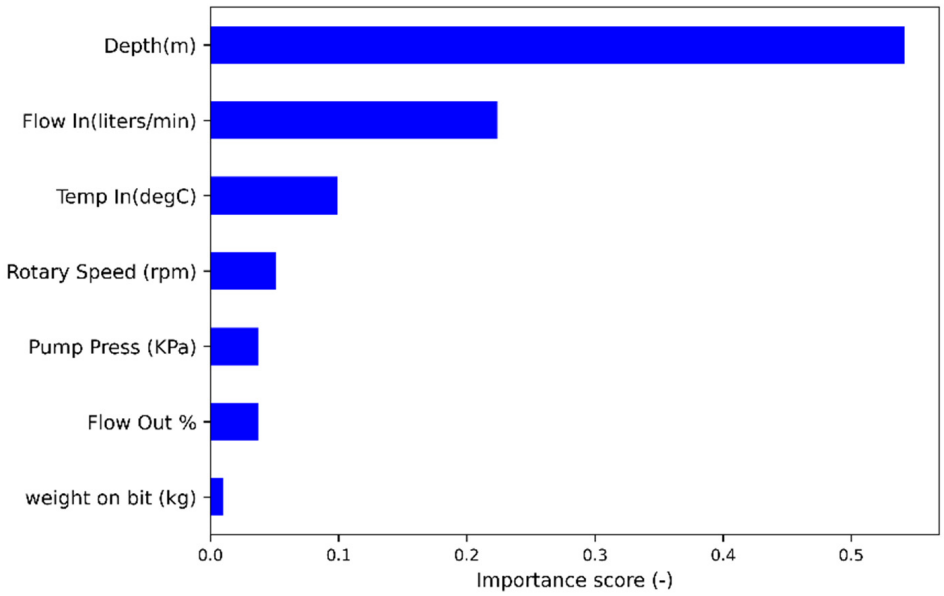

Figure 12. Feature importance.

To summarize, the flow out feature has the least correlation, and the flow out and weight on bit have the least importance based on the tree model (Figure 13).

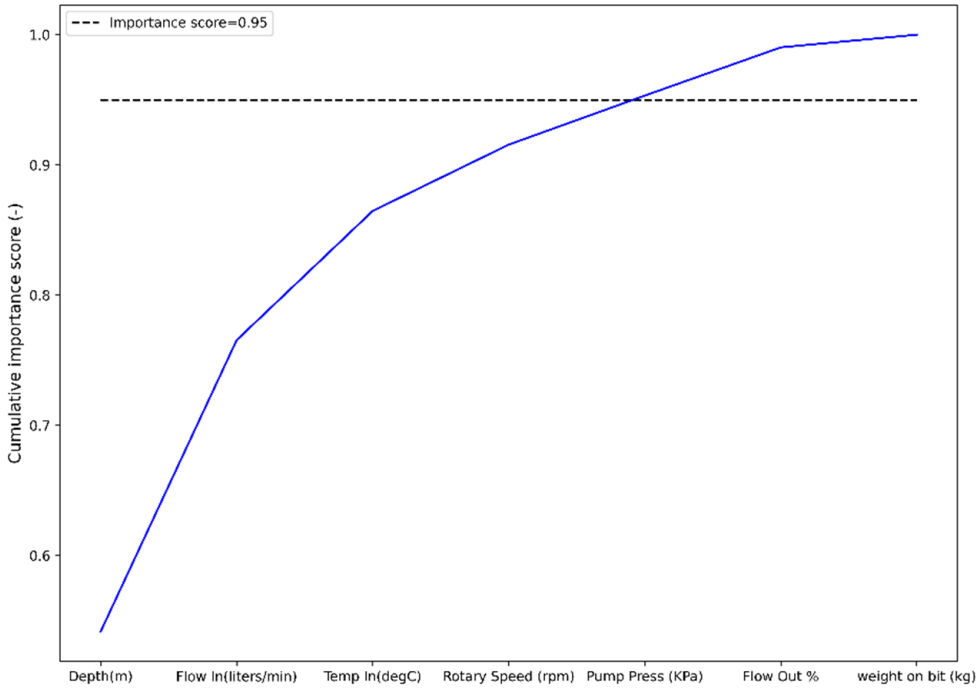

Figure 13. Cumulative feature importance with a 0.95 threshold line.

After examining Figure 14 further, we discovered that the weight on bit (WOB) behavior along the ROP is nearly constant over the high ROP values. The latter explains the WOB's low importance score for this dataset.

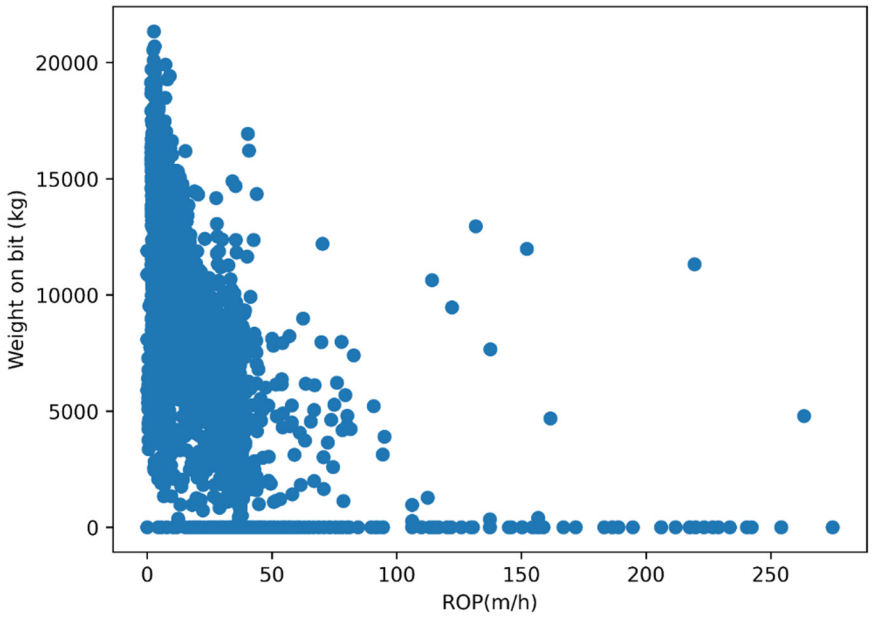

Figure 14. Weight on bit vs ROP scatter plot.

The WOB feature cannot be removed due to the engineering importance of this predictor. Additionally, it showed a high linear correlation in Table 2. After combining domain knowledge and statistical scores, we decided to remove the flow out predictor.

4.1.5. Data Scaling

Machine learning algorithms rely on minimizing the cost function. Thus, data normalization is employed to speed up the calculation of gradient descent. This involves using a common scale for the values of the predictors [35]. Based on the probability distributions in Figure 7, standardization cannot be applied because our predictors do not follow a Gaussian distribution. Min-max normalization is used instead and described in Equation (10).

$$X_{MinMax} = \frac{X - Min\ X}{Max\ \ X - Min\ X} \tag{10}$$

where X represents a predictor value, $Min\ X$ is the minimum value of the considered predictor, and $Max\ X$ is the maximum value of the predictor.

4.2. Accuracy Assessment of Regression Models

A commonly used metric to evaluate the performance of a predictive model on a given data set is the loss function defined by the mean squared error (MSE) [36] and is given by Equation (11)

$$\text{MSE} = \sqrt{\sum_i \left(y_i(x_i) - \hat{y}_i(x_i)\right)^2} \tag{11}$$

If the model fits perfectly the output values, the MSE will be close to 0. However, MSE alone may not be sufficient to clearly see the perfect fit of our model. R^2 score [37] is an additional metric for evaluating model fit. The latter measures the proportion of variability in the target variable y explained by the features X, assuming a linear relationship existing between the predicted variable and the predictors. R^2 is close to 1 if the predictors X can explain the target variable y, and close to 0 if the variability is poorly explained.

$$R^2 = \frac{Total\ squared\ Sum - Residual\ squares\ Sum}{Total\ squared\ Sum} = 1 - \frac{\sum_i \left(y(xi) - \hat{y}\ (xi)\right)^2}{\sum_i \left(y(xi) - \overline{y}(xi)\right)^2} \tag{12}$$

It is also worth mentioning that R^2 is identical to r^2, where r = Cor (X, Y) represents the Pearson correlation coefficient.

4.3. Model Selection

A performance factor to consider for the machine learning model is the generalization ability for unseen test data. It will give a more realistic measure of our model's reliability. To further understand this concept, the bias-variance decomposition [38] is defined as follows:

$$\boldsymbol{Y=f(X)+\varepsilon} \quad \text{where } \boldsymbol{E(\varepsilon)=0} \tag{13}$$

The decomposition of the mean squared error for a given point X_0 with a prediction value of $\hat{f}(x_0)$ is as follows:

$$\boldsymbol{E(y_0-\hat{f}(x_0))^2=Var(\hat{f}(x_0))+[Bias(\hat{f}(x_0))]^2+Var(\varepsilon)} \tag{14}$$

Equation (14) showed that the expected MSE can be decomposed into the sum of three fundamental quantities: the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$, and the variance of the error term $Var(\varepsilon)$. Thus, in order to minimize the expected value of the mean squared error, a statistical model that minimizes both variance and bias in our predictions is required.

When predicting nonlinear data, a simple model (e.g., linear regression) has a high variance and high bias. The model is underfitting, in this case. On the contrary, a complex predictive model has a low bias and high variance for that data. The model is overfitting, in this case, because it has lost its generalization ability. The optimal model neither overfits nor underfits. To assess the generalization ability of a selected model, a traditional method called cross-validation is applied [39,40]. Another popular method is called *k*-fold cross-validation [41].

k fold is calculated by averaging the k MSE values, which is expressed by Equation (15)

$$\boldsymbol{CV_k=\frac{1}{k}\sum_{i}^{k}MSE\ (k)} \tag{15}$$

where *k* represents the number of equally sized parts, called folds.

The model that has the lowest cross-validation error is the best predictive model.

5. Results

5.1. Training and Cross-Validation of Random Forest Regressor

After preprocessing and feature engineering, we trained the random forest with the 6 scaled predictors: depth (m), weight on bit (kg), rotary speed (rpm), pump press (KPa), temp in (°C), and flow in (liters/min). Table 3 shows our new pre-processed dataset, which contains 7293 data samples following outlier removal.

First, we split our data into a 70% training set and a 30% test set. Then, the model's hyperparameters are optimized. The hyperparameter for the random forest method is the number of decision trees. Finally, the model evaluation is done by using a validation set. A total of 20% of the training set is used as the evaluation set for each fold, which corresponds to 14% of the total dataset. Additionally, we used the 5 folds cross-validation. These previous tasks are realized simultaneously using GridSerachCV implemented in python.

5.2. Training and Cross-Validation Artificial Neural Network

For comparison, we trained an artificial neural network (ANN) model, known as an efficient predictive model, with the chosen features. We split our data into an 80% training set and a 20% test set. Then, the model's hyperparameters are optimized. The hyperparameters for the ANN are the number of layers and the number of neurons per layer. The activation function is also a hyperparameter [42]. Previous research in the deep

learning field pointed out that ReLU proved to be an effective activation function for most deep learning applications [43,44]. Thus, we chose ReLU as our activation function due to its popularity in research and industry.

In addition, we need to evaluate the model by using a validation set. The 5 folds validation was also used for cross-validation.

Table 3. Drilling dataset after pre-processing.

	ROP (m/h)	Depth (m)	Weight on Bit (kg)	Rotary Speed (rpm)	Pump Press (KPa)	Temp In (°C)	Flow In (L/min)
count	7293	7293	7293	7293	7293	7293	7293
mean	12.56974	1170.125	10,492.41894	54.855718	8737.605204	47.953857	2710.542394
std	20.19483	654.3972	4130.250795	25.296998	3378.177407	6.626395	511.248043
min	0	25.96	0	0	137.49	29.44	0
25%	3.47	601.94	8308.39	38.12	4593.17	42.72	2347.94
50%	5.47	1176.13	10,807.26	50.38	9877.5	47.34	2650.58
75%	13.46	1736.1	13460.32	75.95	11,510.1	52.7	3120.96
max	274.75	2296.94	21,337.87	178.86	15,171.96	63.51	5864.13

5.3. Model Comparison

The number of decision trees, 200, 300, and 400, were chosen as hyperparameters for the random forest. The cross-validation results revealed that the optimal hyperparameter for the random forest was 300 decision trees. The evaluation metric scores are a mean absolute error of 2.46 and an R^2 score of 0.84. When compared to the standard deviation of the ROP, the mean absolute error represents only 12%. Based on the R^2 score, our optimized random forest model explained 84% of the ROP variance. These are considered acceptable results. Moreover, there may be room for improvement to reduce the error further. Figure 15 gives a more visual understanding of the prediction accuracy.

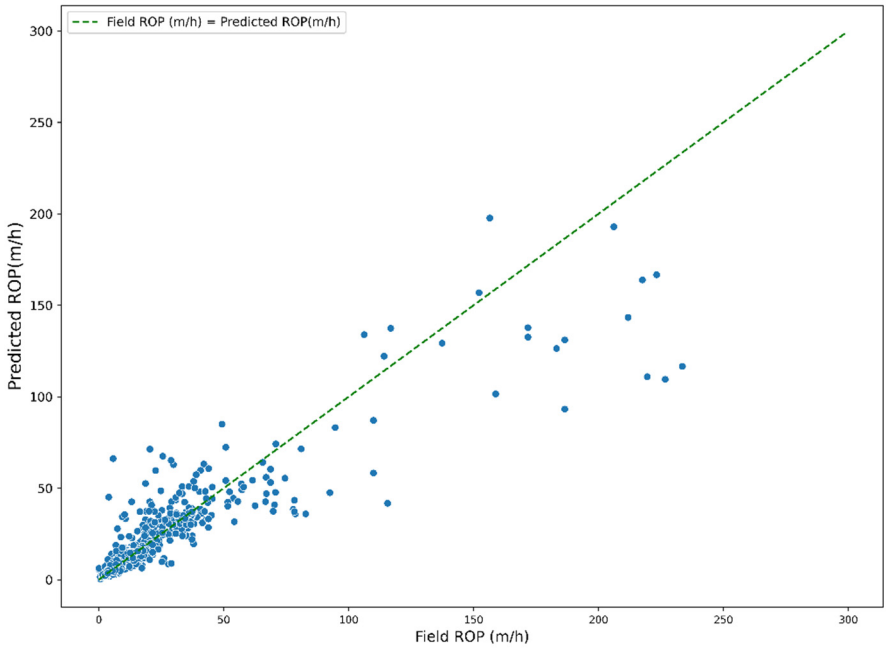

Figure 15. Measured vs predicted ROP by random forest regressor.

The predictions are more accurate in the lower range of the ROP. Typically, high ROP occurs at shallower depths, implying that our model predicts ROP better in deeper formations.

For hyperparameter optimization of the ANN, the number of hidden layers chosen were 1, 2, and 3. The number of neurons for each layer chosen were 2, 3, 6, 12, and 24. The cross-validation results showed that the optimal hyperparameter combination for the neural network proved to be 3 hidden layers with 12 neurons. Figure 16 shows the optimal architecture of the optimized neural network.

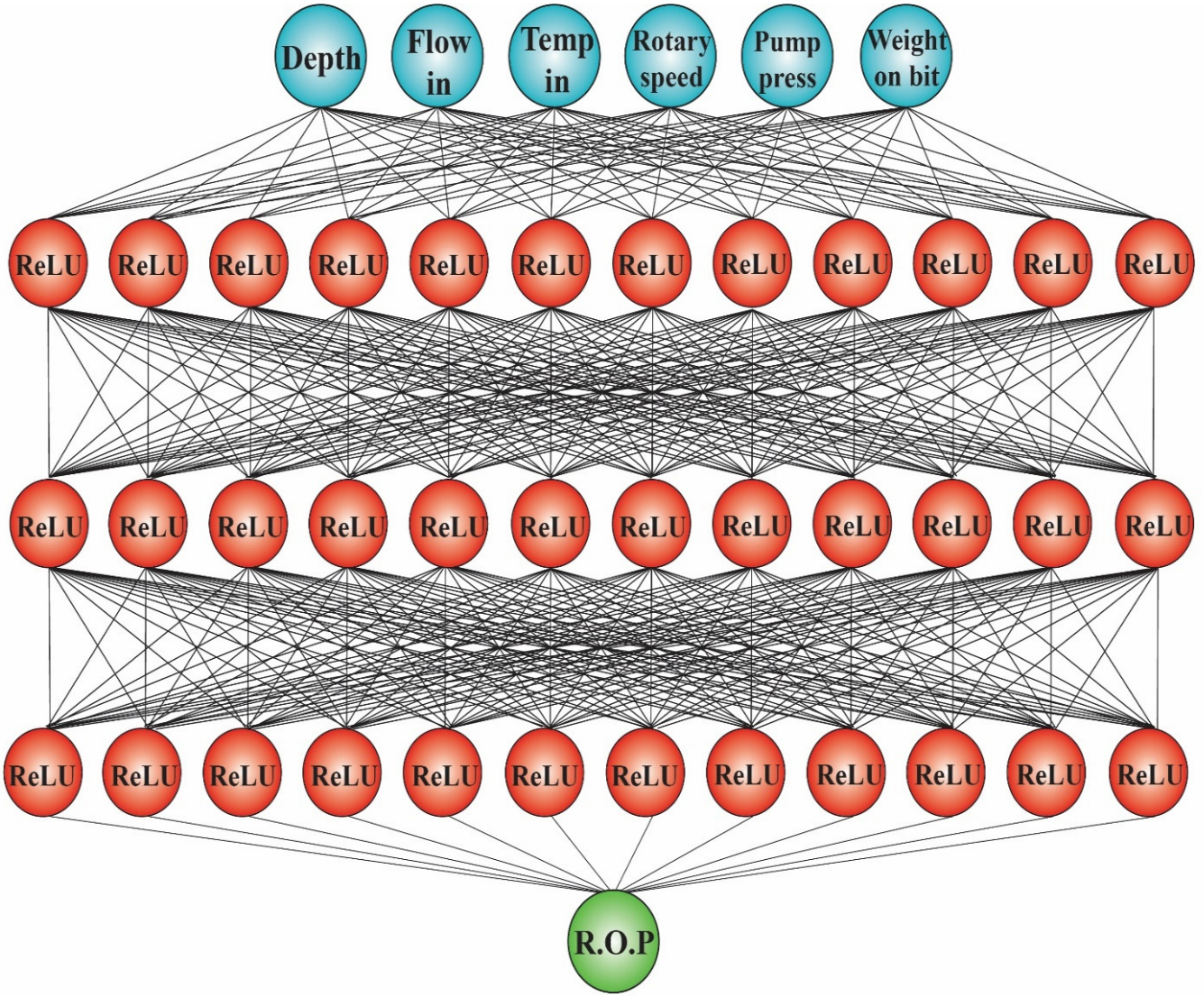

Figure 16. Architecture of the optimized neural network.

The optimized neural network has a mean absolute error of 3.98 and an R^2 score of 0.73. When compared to the standard deviation of the ROP, the mean absolute error represents 19%. Based on the R^2 score, our optimized neural network explained 73% of the ROP variance (Figure 17). It is still an acceptable result, but not as good as the results using the random forest regressor.

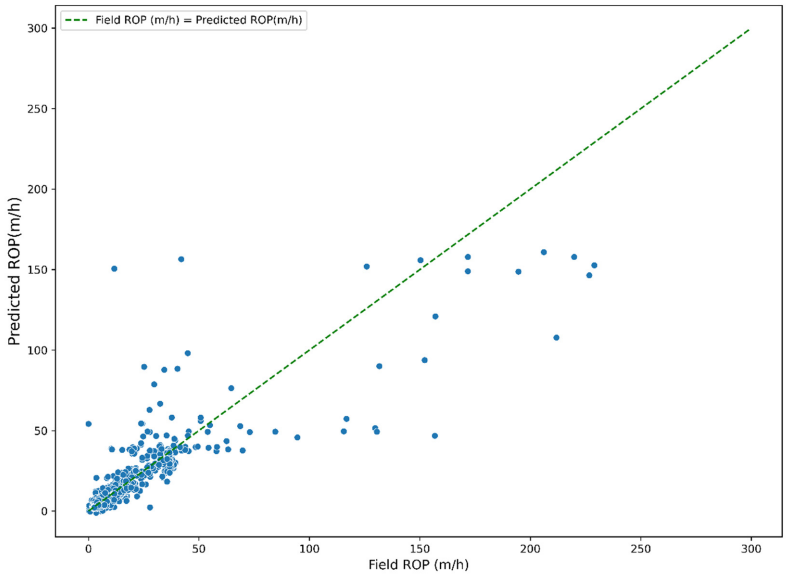

Figure 17. Measured vs predicted ROP by ANN.

Figure 18 illustrates the predicted and filed ROP values, along with the depth, for both the random forest and the artificial neural network.

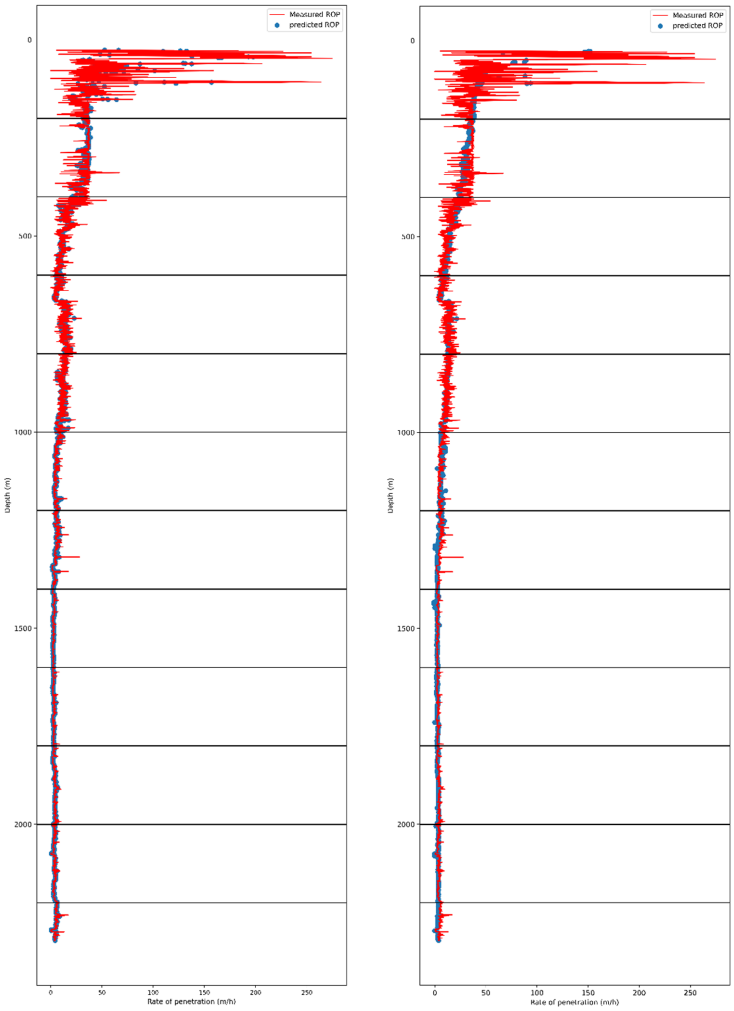

Figure 18. Depth vs ROP profile. (**Left**): random forest prediction. (**Right**): ANN prediction.

6. Discussion

It was indeed interesting that the random forest regressor outperformed the neural network, even though the ANN 's performance was still acceptable. These findings support the results found by [12]. Thus, we conclude that neural networks are not the solution to every problem. Neural networks tend to overfit the training data. In practice, to solve this overfitting issue, we usually refer to the regularization techniques. After several trials, it this did not considerably affect the performance. Additionally, 7293 data samples used for training were not enough. Deep learning models require a massive amount of data to perform better. Machine learning models, such as the random forest method, outperform neural networks when the dataset is smaller, not to mention the huge computational cost of ANN compared to the random forest method.

The Pearson correlation coefficient measures a linear relationship between variables. The feature importance of the random forest method extracts a more complex relationship between variables. This indicated that the depth variable is the most important contributor to the ROP value.

After building our predictive model, we must validate it with at least one other well from the same geothermal field. Formations inhibit similar compressive strength in the same field. Additionally, the previous models could be improved if we separated our predictions into different formations and built a predictive model for each formation, since, in drilling practice, we tend to evaluate the ROP on each formation, rather than the measured depth due to the variations of compressive strength for different formations. This will definitely reduce the error for predicting the behavior of ROP values in shallower formations.

7. Conclusions

After hyperparameter optimization, the random forest regressor method, containing 300 decision trees, was a better method, compared to the ANN, for the evaluation of the 58–32 well dataset. The mean absolute error = 2.46, which is a low margin of error compared to the overall standard deviation of our field ROP values. The R^2 score = 0.84, indicating that the model explained 84% of field ROP variability. Domain knowledge, the Pearson correlation coefficient, and the feature importance of decision trees supported the feature selection process. The data-driven models for ROP prediction are promising tools for future drilling projects on this FORGE site.

Author Contributions: Conceptualization and Supervision, T.M.; Writing—original draft, M.A.B.A.; Writing—review & editing, T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The research was conducted at the University of Miskolc as part of the project supported by the Ministry of Innovation and Technology from the National Research, Development and Innovation Fund according to the Grant Contract issued by the National Research, Development and Innovation Office (Grant Contract reg. nr.: TKP-17-1/PALY-2020).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Capuano, L.E. Geothermal well drilling. In *Geothermal Power Generation*; Elsevier: Amsterdam, The Netherlands, 2016; pp. 107–139. [CrossRef]
2. Thorhallsson, S.; Sveinbjornsson, B.M. Geothermal drilling cost and drilling effectiveness. In Proceedings of the Short Course on Geothermal Development and Geothermal Wells, Santa Tecla, El Salvador, 11–17 March 2012.

3. Soares, C.; Gray, K. Real-time predictive capabilities of analytical and machine learning rate of penetration (ROP) models. *J. Pet. Sci. Eng.* **2018**, *172*, 934–959. [CrossRef]
4. Maurer, W.C. The 'Perfect—Cleaning' Theory of Rotary Drilling. *J. Pet. Technol.* **1962**, *14*, 1270–1274. [CrossRef]
5. Alawami, M. A real-time indicator for the evaluation of hole cleaning efficiency. In Proceedings of the SSPE/IATMI Asia Pacific Oil & Gas Conference and Exhibition, Bali, Indonesia, 29–31 October 2019.
6. Bourgoyne, A.T.; Millheim, K.K.; Chenevert, M.E.; Young, F.S. *Applied Drilling Engineering*; Society of Petroleum Engineers: Richardson, TX, USA, 1986; Volume 2.
7. Dupriest, F.E.; Koederitz, W.L. Maximizing Drill Rates with Real-Time Surveillance of Mechanical Specific Energy. In Proceedings of the SPE/IADC Drilling Conference, Amsterdam, The Netherlands, 23–25 February 2005. [CrossRef]
8. Young, F.S., Jr. Dynamic Filtration During Microbit Drilling. *J. Pet. Technol.* **1967**, *19*, 1209–1224. [CrossRef]
9. Bourgoyne, A.T., Jr.; Young, F.S., Jr. A Multiple Regression Approach to Optimal Drilling and Abnormal Pressure Detection. *Soc. Pet. Eng. J.* **1974**, *14*, 371–384. [CrossRef]
10. Shi, X.; Meng, Y.; Li, G.; Li, J.; Tao, Z.; Wei, S. Confined compressive strength model of rock for drilling optimization. *Petroleum* **2015**, *1*, 40–45. [CrossRef]
11. Brenjkar, E.; Delijani, E.B. Computational prediction of the drilling rate of penetration (ROP): A comparison of various machine learning approaches and traditional models. *J. Pet. Sci. Eng.* **2022**, *210*, 110033. [CrossRef]
12. Alsaihati, A.; Elkatatny, S.; Gamal, H. Rate of penetration prediction while drilling vertical complex lithology using an ensemble learning model. *J. Pet. Sci. Eng.* **2021**, *208*, 109335. [CrossRef]
13. Atashnezhad, A.; Akhtarmanesh, S.; Hareland, G.; Al Dushaishi, M. Developing a Drilling Optimization System for Improved Overall Rate of Penetration in Geothermal Wells. In Proceedings of the 55th U.S. Rock Mechanics/Geomechanics Symposium, Virtual, 18–25 June 2021.
14. Mitchell, T.M. Machine learning and data mining. *Commun. ACM* **1999**, *42*, 30–36. [CrossRef]
15. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*, 1st ed.; Routledge: London, UK, 2017. [CrossRef]
16. Hastie, T.; Tibshirani, R.; Friedman, J.H. *The Elements of Statistical Learning—Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2001.
17. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning*; Springer: New York, NY, USA, 2013; Volume 103. [CrossRef]
18. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
19. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
20. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
21. Géron, A. *Hands-on Machine Learning with Scikit-Learn and Tensorflow: Concepts*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.
22. Moore, J.; McLennan, J.; Pankow, K.; Simmons, S.; Podgorney, R.; Wannamaker, P.; Xing, P. The Utah Frontier Observatory for Research in Geothermal Energy (FORGE): A Laboratory for Characterizing, Creating and Sustaining Enhanced Geothermal Systems. In Proceedings of the 45th Workshop on Geothermal Reservoir Engineering, Stanford, CA, USA, 10–12 February 2020; p. 10.
23. Frontier Observatory for Research in Geothermal Energy (FORGE). *Phase 2B Tropical Report*; University of Utah: Salt Lake City, UT, USA, 2018.
24. Allis, R.; Moore, J.; Davatzes, N.; Gwynn, M.; Hardwick, C.; Kirby, S.; Simmons, S. EGS Concept Testing and Development at the Milford, Utah FORGE Site. In Proceedings of the 41st Workshop on Geothermal Reservoir Engineering, Stanford, CA, USA, 22–24 February 2016; p. 13.
25. Simmons, S.F.; Kirby, S.; Bartley, J.; Allis, R.; Kleber, E.; Knudsen, T.; Moore, J. Update on the Geoscientific Understanding of the Utah FORGE Site. In Proceedings of the 44th Workshop on Geothermal Reservoir Engineering, Stanford, CA, USA, 11–13 February 2019; p. 10.
26. Kirby, S.M.; Knudsen, T.; Kleber, E.; Hiscock, A. Geologic Setting of the Utah FORGE Site, Based on New and Revised Geologic Mapping. *Trans. Geotherm. Resour. Counc.* **2018**, *42*, 1097–1114.
27. Nielson, D.L.; Evans, S.H., Jr.; Sibbett, B.S. Magmatic, structural, and hydrothermal evolution of the Mineral Mountains intrusive complex, Utah. *GSA Bull.* **1986**, *97*, 765–777. [CrossRef]
28. Podgorney, R. *Utah FORGE: Drilling Data for Student Competition*; Idaho National Laboratory: Idaho Falls, ID, USA, 2018. [CrossRef]
29. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013. [CrossRef]
30. Majdoub, A. Development of a Machine Learning Model Based on Feature Selection to Predict Volve Production Rate. *Discover-Volve*. 2020. Available online: https://www.discovervolve.com/2021/02/23/development-of-a-machine-learning-model-based-on-feature-selection-to-predict-volve-production-rate/ (accessed on 1 May 2022).
31. Box, G.E.P.; Tidwell, P.W. Transformation of the Independent Variables. *Technometrics* **1962**, *4*, 531–550. [CrossRef]
32. Geladi, P.; Manley, M.; Lestander, T. Scatter plotting in multivariate data analysis. *J. Chemom.* **2003**, *17*, 503–511. [CrossRef]
33. Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
34. Zhou, Z.; Hooker, G. Unbiased Measurement of Feature Importance in Tree-Based Methods. *ACM Trans. Knowl. Discov. Data* **2021**, *15*, 1–21. [CrossRef]

35. Patro, S.G.K.; Sahu, K.K. Normalization: A Preprocessing Stage. *arXiv* **2015**, arXiv:150306462. [CrossRef]
36. Wackerly, D.D.; Mendenhall, W.; Scheaffer, R.L. *Mathematical Statistics with Applications*, 7th ed.; International ed.; Thomson Higher Education: Belmont, CA, USA, 2008.
37. Barten, A.P. The coefficient of determination for regression without a constant term. In *The Practice of Econometrics*; Heijmans, R., Neudecker, H., Eds.; Springer: Dordrecht, The Netherlands, 1987; Volume 15, pp. 181–189. [CrossRef]
38. James, G.M. Variance and Bias for General Loss Functions. *Mach. Learn.* **2003**, *51*, 115–135. [CrossRef]
39. Browne, M.W. Cross-Validation Methods. *J. Math. Psychol.* **2000**, *44*, 108–132. [CrossRef]
40. Xu, Y.; Goodacre, R. On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *J. Anal. Test.* **2018**, *2*, 249–262. [CrossRef] [PubMed]
41. Bengio, Y.; Grandvalet, Y. No Unbiased Estimator of the Variance of K-Fold Cross-Validation. *J. Mach. Learn. Res.* **2004**, *5*, 1089–1105.
42. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
43. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *arXiv* **2018**, arXiv:181103378.
44. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; p. 8.