Research Article

URF PUBLISHERS
connect with research world

Journal of Petroleum & Chemical Engineering

https://urfpublishers.com/journal/petrochemical-engineering

Vol: 1 & Iss: 1

Comparative Analysis of Machine Learning Algorithms in Predicting Rate of Penetration during Drilling

Olaosebikan Abidoye Olafadehan*, Ikenna David Ahaotu

Department of Chemical and Petroleum Engineering, University of Lagos, Akoka-Yaba, Lagos 101017, Nigeria

Citation: Olafadehan OA, Ahaotu ID, Comparative Analysis of Machine Learning Algorithms in Predicting Rate of Penetration during Drilling. *J Petro Chem Eng* 2023;1(1): 32-47.

Received: 14 October, 2023; **Accepted:** 31 October, 2023; **Published:** 07 November, 2023

***Corresponding author:** Olafadehan OA, Department of Chemical and Petroleum Engineering, University of Lagos, Akoka-Yaba, Lagos 101017 Nigeria, Phone: +234802-912-9559, Email: oolafadehan@unilag.edu.ng

ABSTRACT

Drilling for potential oil and gas reserves is one of the foremost practices in the petroleum industry. The drilling process, however, is quite expensive and can take quite some time to accomplish. Hence, there has been a rise in the need to reduce cost and time by optimizing the rate of penetration during drilling, which has led to the development of mathematical models to describe and evaluate this process. However, the accuracy of these models has varied owing to variation of the drilling parameters accounted for in each model. This event has led to the usage of alternative approaches such as Data driven models. In this study, the predictive capacities of the rate of penetration (ROP) during drilling using machine learning (ML) algorithms of support vector machine regression (SVR), Random Forest regression (RF), Linear regression (LR), KNearest neighbors (KNN), Stacking technique, Voting technique and Convolution neural network (CNN), were compared. Data from an oil well in Nigeria was used in this investigation. The data for the well was split into train–test sets in the ratio of 60:40. The train data was used to train and select the best model before making predictions on the test sets. The Stacking technique was found to have the best performance across both training and test data sets with respective accuracies of 99.8% and 97.5% in terms of the –score. The Voting technique also performed well, with respective accuracies of 93.6% and 92.6% in terms of the –score across both sets of data. The CNN model equally performed well on the training and test data sets, with respective accuracies of 92.4% and 92.8% in terms of the –Score. Generally, the machine learning models were able to detect patterns and gain valuable insights into the data. They can be employed for real time prediction of the rate of penetration during oil well drilling.

Keywords: Rate of penetration; Drilling; Artificial intelligence; Machine learning algorithms; train–test data.

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DDR	Daily Drilling Report
KNN	KNearest Neighbors
MAE	Mean Absolute Error
ML	Machine Learning
R2	Coefficient of determination
RMSE	Root mean squared Error.
ROP	Rate of penetration.
RPM	Rotary speed,
SVR	Support Vector Regressor
WOB	Weight on bit, kblf

1. Introduction

Drilling is a key aspect of the petroleum industry. It is the process of boring a hole deep into the subsurface section of the earth in order to reach formations with hydrocarbon reserves, for the aim of hydrocarbon recovery. The importance of this process cannot be understated and as a result, a lot of different drilling technologies were implemented to maximize drilling operations. The popular drilling method used today known as the rotary drilling, which is applied in drilling the majority of onshore and offshore wells and makes use of an applied axial force on the rotating drill bit to achieve penetration. It is impossible to overstate the significance of this procedure, which is why numerous drilling methods have been used to maximize drilling operations. The bulk of onshore and offshore wells are drilled using the widely used technique known as the rotary drilling, which applies an axial force to the revolving drill bit to accomplish penetration. In a rotary drilling process, key parameters need to be considered to ensure optimal operations, and a key parameter among these is the rate of penetration, ROP. It is the depth of penetration accomplished per unit time, and is usually measured as a factor of how many feet the bit can drill in an hour (i.e., ft/h). However, evaluation of ROP is difficult due to the complex relationship between other drilling parameters affecting the ROP. The rate of penetration (ROP) prediction is a key task in drilling economical assessments[1]. Not always is the lowest cost per foot provided by the fastest drilling pace. A rise in the project's overall cost may be caused by other factors. The characteristics of drilling fluid (such as mud viscosity, mud density, filtration loss), mechanical characteristics (such as bit type and weight), and formation properties (such as porosity, rock abrasivity, formation elasticity, formation stress, permeability) are a few examples of the properties that affect penetration rate[2]. Hence, it is important to maximize the rate of penetration in order to mitigate some of the general cost associated with drilling for extended periods. Therefore, it is necessary to understand the relationship between the ROP and other operational parameters.

Mathematical models have been used to model the relationship between some operational parameters and ROP e.g., Bourgoyne and Young[3] model and the Bingham[4]. The accuracy of these models has varied due to variation in the drilling parameters considered in each model. This has led to the usage of alternative approaches such as a data driven model e.g., artificial intelligence (AI). Artificial intelligence methods have developed rapidly over the past decades and has led to it been implemented in various sectors, including the oil and gas industry. Colossal amount of data is been generated on the oil field during operating hours. These data include drilling data, production data, seismic data and mud log data, amongst others. These data sets can be trained using artificial intelligence methods to make future predictions and generate hidden insights into the data. The AI methods have been used extensively in applications to the petroleum industry where they can provide solutions to drilling problems such as prediction of drill bit wear from drilling parameters, real-time predictions of alterations in drilling fluid rheology[5], and the estimation of oil recovery factor for water drive sandy reservoirs[6].

1.1 Artificial Intelligence

Machine Learning (ML) and Deep Learning (DL) are branches of artificial intelligence that deals with computerized systems and algorithms learning from previous data generated[7]. By utilizing various algorithmic strategies, they enable the systems to perform computational tasks without requiring explicit programming and learn from the data. Finding patterns in numerical data by applying computer algorithms to convert data into numerical form is known as machine learning. Amongst other formats, the data may be in the form of pictures, music, numbers, or alphabetical data. The algorithms used to find the patterns within these data are called machine learning models. These models, which include linear regression, logistic regression, decision Trees, random forest, K-Means, K-Nearest Neighbors, are used for prediction, data sub-grouping and sound-detection, amongst others. They have been applied to aid in the prediction of ROP values with better accuracy and generalization. ML operations are divided into supervised and unsupervised learning. Supervised learning is a paradigm in machine learning here input objects and a desired output value train a model. The training data is processed, and builds a function that maps new data on expected output values (e.g., regression and classification). In unsupervised learning, the data has no target label, the machine learning model aims at finding hidden patterns in the data using algorithms to make critical judgments in the future (e.g., clustering and recommendation).

Deep learning is a branch of the machine learning and artificial intelligence that mimics the operation of how the human brain receives, process and transmit information, as depicted in Figure 1.

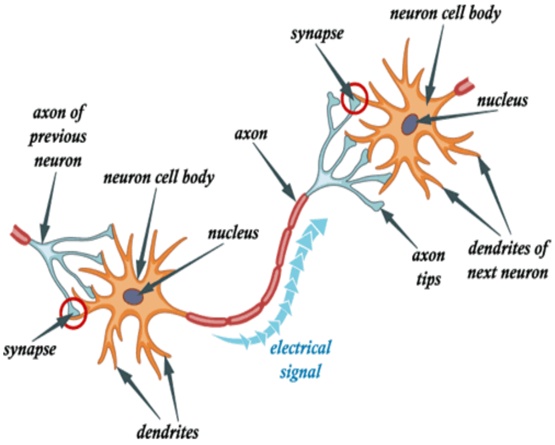


Figure 1: Human neuron model.

Deep learning (DL) is essentially a neural network with one or more layers. The components of the human neural network are modelled similar to the neural network operation[8]. The dendrites act as input nodes, cell body represents activation function, synapse is the weightage of each input, and the axon terminal is the output node as shown in Figure 2.

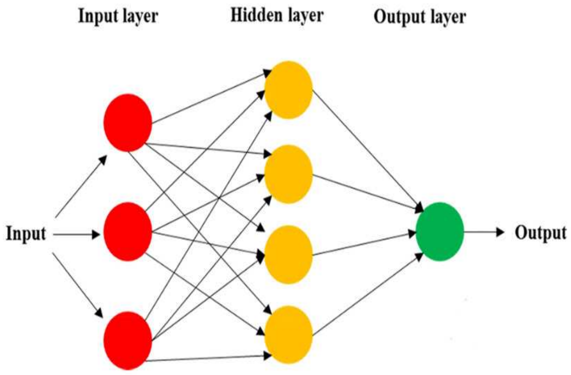


Figure 2: A typical feed forward neural network architecture[9].

Neural networks (or deep learning) are massively parallel distributed processor that store and make use of experiential knowledge. It is classified into 3 parts: artificial neural network (ANN), convolutional neural network (CNN), and recurrent neural network (RNN), which are used to carry out different operations. The ANN is mostly used to carry out regression and classification problems. The CNN is mostly used to carry out image processing and prediction while the RNN is mostly used for forecasting operations. A convolutional neural network and a few machine learning strategies are used in this work. Convolutional layers, feature extractors (filters), pooling layers, hidden layers, and one or more output layers are the components of a convolutional neural network. Weights are used to connect the layers in the hidden layer of the CNN structure. These weights facilitate information flow between layers and aid in neural network training. An activation function is present in every hidden layer, which helps to save computational time and cost by converting the data into a more computer-interactive format. To extract important features from the data, convolutional layers assist in performing convolutional operations on the data.

Before the data is sent to the filter, which extracts the features and patterns in the dataset, the convolutional layer typically receives the input in the form of length, breadth, height, and color channels. CNNs have two feature extraction layers: one that makes use of pooling layers and the other that makes use of filters. To extract even more important insights from the dataset, a pooling layer made up of a pooling approach is employed to perform pooling on the features that the filter helped extract. To conduct out-pooling, different sorts of pooling techniques are employed, such as MaxPooling, Average Pooling, and Global Pooling.

Bilgesu et al[10]. used an artificial neural network to develop an ROP model, which was dependent on several operating parameters. A data of 500 points was used, with nine features, which were tooth wear, rotary speed, torque, weight on bit, pump flow rate, rotating time, bearing wear, formation drillability, and formation abrasiveness. A train-test ratio of 9:1, which implies 90% of the data was used for training and 10% for validating the model. A coefficient of determination (R^2) between 0.902 and 0.982 was achieved after cross-validation across the data. In the work of Arabjamaloei and Shadizadeh[11], an artificial neural network with a single hidden layer of 10 neurons was developed and combined with genetic algorithm (GA) to create a model to predict ROP values. There were seven features and 300 points (rows) in the data. The bit type, formation properties, bit operating condition (rotary speed and bit weight), bit tooth wear, bit hydraulics, hydrostatic head, and equivalent circulating density were the input features. A total of 224 points were used for model training, 56 points for validation, and 20 points for testing. The generic algorithm was employed to find where the maximum rate of penetration occurred. With a low mean-square error for both training and test set, it was concluded that the neural network is valid for other data sets that fall within the range of data set used for training the model[12].performed a comparative evaluation of models for estimating the rate of penetration (ROP) by utilizing field data from a well located in Iran**.** The model used for this study were the Bingham[4], Warren[13] and, Bourgoyne and Young[3] models. They carried out ROP predictions on wells that were drilled with roller cone and PDC bits, and comparison was carried out on three separate drilling sections. However, there was a short coming of this study, in that threshold $W_{dB}$was neglected due to lack of drill-off test been carried out. The findings of this study demonstrated that among the models examined, the Bourgoyne and Young model exhibited the highest level of predictive performance. Mahasneh[14] developed a mathematical model to predict the rate of penetration (ROP) in gas wells, considering the factors of weight on bit (WOB), bit rotation speed (RPM), flow rate (FR), formation strength, depth, and formation compaction. He then used his model to optimize the drilling parameters for a gas well in Jordan, increasing the ROP by 15% and reducing the cost of drilling by 10%. Mahasneh[14]'s study demonstrated the importance of drilling optimization in improving the efficiency and cost-effectiveness of drilling operations. Amar and Ibrahim[15] worked on the comparative analysis of physics-based equations with artificial neural networks (ANN). They developed two neural network models to evaluate the ROP values. The input parameters into the neural networks were formation depth, ECD, weight on bit, DSR, pore pressure gradient, drill bit tooth wear, and Reynolds number function. The physics-based equations used for the comparative analysis were the Bingham[4] model and Bourgoyne and Young[3] model. A comparison of the predictive accuracy of the developed ANN-based models with the available empirical equations showed that both ANN-based models were highly accurate for estimating the ROP as compared with the empirical equations. Shi et al.[16]predicted the rate of penetration (ROP) using the Extreme Learning Machine (ELM) and Upper-layer solution-ware (USA) techniques. To construct the predictive models, various input parameters such as formation properties, rig hydraulics, bit specifications, weight on bit, rotary speed, and mud properties were utilized. These input features were selected based on reservoir data from Bohai Bay, China. The performance of the developed models using ELM and USA techniques was compared with an artificial neural network model. The accuracy of these models was evaluated using metrics such as regression coefficient (R^2), mean absolute error (*MAE*), and root mean square error (*RMSE*). The findings indicated that the ROP model developed with the USA technique exhibited the highest predictive performance compared to the other models. Additionally, it was observed that the development of the ROP model using the extreme learning technique required the most time investment. Ahmed et al.[17] investigated the application of a support vector machine model to estimate the rate of penetration in a formation containing shale materials. The input features used in the model were hinged on drilling parameters and mud properties such as weight on bit, rotary speed, pump flow rate, standpipe pressure, drilling torque, mud density, plastic viscosity, funnel viscosity, yield point and solid content (%). The support vector machine model and the Bourgoyne and Youngs model were trained on more than 400 real data in shale formation using these 10 features as inputs. The two models were both compared on their predictive performance on the test data. The Bourgoyne and Young (1974) model produced a coefficient of determination (R^2) of 0.0692 and an absolute percentage error of 23.41%. By applying the support vector machine (SVM) model, a coefficient of determination (R^2) of 0.995 and an absolute percentage error of 2.82% were obtained. It was concluded that SVM can be used to predict ROP with higher accuracy and also generate ROP values faster than the Bourgoyne and Young[3] model. Elkatany[5] developed an artificial neural network (ANN) model to predict the rate of penetration (ROP) using data collected from three vertical wells in an offshore oilfield. The ANN-ROP model was obtained based on drilling parameters and drilling fluid properties. Two wells were utilized for training the model, and the third well was used to evaluate the accuracy of the model.

The performance of the ANN-ROP model was compared to other ROP models of Bingham (1965), Bourgoyne and Young[3], and Maurer[18]. Elkatany[5] concluded that the proposed ANN-ROP model exhibited superior performance over others considered in his work. The training data consisted of 3333 data points and yielded a coefficient of determination (R^2) of 0.99, with an average absolute percentage error (*AAPE*) of 5%. The test set, consisting of 2700 unseen data points from the third well, resulted in the ANN-ROP model predicting the rate of penetration with $R^2 = 0.9$ and *AAPE* = 4%. Zhang et al.[19] proposed a deep convolutional neural network (CNN) model for predicting the rate of penetration (ROP) during drilling operations. The authors argued that existing models for predicting ROP are often inaccurate and unreliable, and that deep learning methods could provide a more accurate and practical solution. They collected data from drilling operations in two different fields and used it to train and test the proposed deep CNN model in their work. The model consists of six convolutional layers and is trained using a mean absolute percentage error (*MAPE*) loss function. The authors compared the performance of their deep CNN model to other machine learning models and found that it outperformed these models in terms of accuracy and reliability. They also conducted sensitivity analyses to determine the most important features for predicting ROP. They found that the weight on bit, the rotary speed, and the mud flow rate were the most important features for predicting ROP. Zhao et al.[20] focused on developing multiple artificial neural network (ANN) models for predicting the rate of penetration (ROP) using data collected from a gas well located in the southern region of Iran. A dataset comprising 3180 data points was obtained from various drilling sections, involving one run of a roller-cone bit and three runs of PDC bits. To construct the ANN-ROP models, several input variables were considered, including depth, rotary speed of the bit, weight on bit (WOB), shut-in pipe pressure, fluid rate, mud weight, the ratio of yield point to plastic viscosity, and the ratio of 10-minute gel strength to 10-second gel strength. Three different training functions, namely Levenberg-Marquardt (LM), Scaled Conjugate Gradient (SCG), and One-Step Secant (OSS), were employed in combination with the neural networks to estimate the penetration rates. It was concluded that the ANN-ROP model utilizing the Levenberg-Marquardt (LM) function demonstrated the best prediction performance, achieving a regression coefficient (R^2) of 0.91 in training and 0.89 in testing. Furthermore, they also applied the Artificial Bee Colony (ABC) algorithm to optimize the ROP. The optimization process resulted in an approximate improvement of 20–30% in the rate of penetration. Abdulmalek et al.[21] carried out a comparative analysis between artificial intelligence techniques and some traditional models for ROP prediction in shaley formations. An artificial neural network was developed for the ROP prediction in the shale formation. The parameters considered for the prediction of the rate of penetration (ROP) included torque, standpipe pressure, pump rate, mud weight, funnel and plastic viscosities, solid content, and yield point. The traditional ROP models such as those proposed by Bingham[4], Warren[13], Bourgoyne and Young[3], Maurer[18] and Hareland and Hoberock[22], were selected for comparison. Both the artificial neural network–ROP (ANN-ROP) model and the traditional models underwent training and testing using a dataset consisting of 347 data points from a deep shale formation in an onshore oilfield. Additionally, 200 new data points from an upper shale formation were utilized to validate the models. The results indicated that the ANN-ROP model outperformed the other models in comprehending the intricate relationships within the data and making accurate predictions. The ANN-ROP model achieved a rate of penetration prediction with an average absolute percentage error (*AAPE*) of 5.776% and a regression coefficient (R^2) of 0.996. Ashrafi et al.[23] explored the prediction of rate of penetration (ROP) using various optimization algorithms and neural network architectures. The optimization algorithms employed included Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Biogeography-based Optimizer (BBO), and Imperialist Competitive Algorithm (ICA). These algorithms were combined with different neural network architectures to develop hybrid ROP models. To evaluate the performance of the hybrid models, the results were compared with two other models: Non-linear Multiple Regression (NLMR) and Linear Multiple Regression (LMR) techniques. For the hybrid models, two popular neural network architectures, namely Multi-Layer Perception (MLP) and Radial-Based Function (RBF), were utilized. These architectures consisted of two hidden layers with 4 and 6 neurons, respectively. The activation function used in the hidden layers and output layer was tan-sigmoid. The input features were weight on bit, rotational speed of the drill bit, pump inlet flow rate, pore pressure pump pressure, gamma ray, density log, and shear wave velocity. The dataset used for the study consisted of 1000 data points, collected from the Marun oilfield in Iran. It was concluded in their study that the hybrid models utilizing PSO-MLP and PSO-RBF neural networks exhibited the best predictive accuracy for ROP. The root mean square error (*RMSE*) values for these models were 1.12 and 1.4, respectively, indicating their superior performance compared to the other developed models. Iqbal[24] developed a mathematical model to predict the rate of penetration (ROP) in drilling operations, considering the factors of weight on bit (WOB), bit rotation speed (RPM), flow rate (FR), formation strength, depth, and formation compaction. He then used his model to optimize the drilling parameters for a real-time drilling dataset from a Middle Eastern oil field, increasing the ROP by 10% and reducing the cost of drilling by 5%. Iqbal's study demonstrates the importance of using real-time drilling parameters to optimize drilling operations and provides a valuable contribution to the field of drilling engineering. Burgos et. al.[25] developed a convolutional neural network (CNN) model to predict the rate of penetration (ROP) during rotary drilling operations. The model takes in 10 drilling parameters as inputs, such as weight on bit, rotary speed, flow rate, and hook load. The inputs are normalized between 0 and 1. The CNN architecture consists of 3 convolutional layers followed by 2 fully connected layers. The output layer has a single node with a linear activation to predict the ROP value. The model was trained on data from over 600 wells. It achieved a mean absolute percentage error (*MAPE*) of 9.3% on the test set, outperforming traditional machine learning models like linear regression, random forests, and support vector regression. An ablation study showed that the CNN's ability to learn complex non-linear relationships between the drilling parameters allowed it to accurately predict ROP, whereas simply averaging the inputs did not work as well. The model was able to generalize the data from 50 additional wells, with the *MAPE* only increasing slightly to 10.2%. This shows the model has good generalization performance. In conclusion, the CNN approach effectively modelled the complexity between drilling parameters and ROP, outperformed traditional models, and generalized well to new data. This could enable more efficient drilling operations through accurate ROP predictions. Monazami et al.[26] used an artificial neural network (ANN) to predict the rate of penetration (ROP) in drilling operations. The ANN model took cognizance of formation strength, depth, formation compaction, pressure differential, bit diameter, weight on bit (WOB), bit rotation (RPM), and bit hydraulics. The authors evaluated the performance of their ANN model on a test dataset of ROP data. They found that the model was able to predict ROP with high accuracy. The average error between the predicted and actual ROP values was less than 5%. The model was able to predict ROP with high accuracy, suggesting that ANN is a promising tool for optimizing drilling parameters and improving the efficiency and cost-effectiveness of drilling operations. Abbas et al.[27] employed an artificial neural network (ANN) approach to develop a computational-based method for predicting the rate of penetration (ROP). Through a thorough analysis of feature selection, it was determined that out of the 25 input variables examined, 19 variables had the greatest influence on the ROP. A dataset consisting of 13,125 data points from 14 deviated wells in a formation located in southern Iraq was collected for the study. The data specifically pertained to the 8 ½" production casing section, which was drilled using a drag bit and a conventional bottom hole assembly (BHA) with a water-based mud circulating system. It was concluded that the ROP model based on the artificial neural network, utilizing three hidden layers and employing the tan–sigmoid activation function, exhibited the highest efficiency in predicting ROP. The model achieved a regression coefficient of 0.92 during training and 0.97 during testing, with mean absolute percentage errors (*MAPE*) of 9.1% and 8.8% in training and testing, respectively. Furthermore, the model demonstrated good performance on unseen data and did not exhibit overfitting issues. Miyora[28] studied the factors that affect the rate of penetration (ROP) in geothermal drilling and developed a mathematical model to predict ROP based on these factors. The model includes formation strength, depth, formation compaction, pressure differential, bit diameter, weight on bit (WOB), bit rotation (RPM), and bit hydraulics. Miyora () found that all these factors have a significant impact on ROP and used his model to optimize the drilling parameters for Well MW-17 in Menengai, Kenya, increasing the ROP by up to 20%. Al-AbdulJabbar et al.[29] utilized an artificial neural network (ANN) in combination with self-adaptive differential evolution (SADE) to predict the rate of penetration (ROP) specifically in horizontal carbonate reservoirs. The model incorporated six input variables, including rotary speed, torque, weight on bit, as well as formation petrophysical properties such as gamma ray, resistivity, and bulk density data. The developed model demonstrated strong performance, achieving a regression coefficient (R^2) of 0.96 and a mean absolute percentage error (*MAPE*) of 5.12%. To further evaluate the accuracy of the model, an unseen well was used as test data. The resulting regression coefficient (R^2) and *MAPE* values were 0.95 and 5.8%, respectively. Furthermore, their study aimed to enhance the interpretability of the ROP model by extracting the weights and biases in a matrix form, effectively transforming it from a black box model to a white box model. Wang et al.[30] proposed a hybrid ensemble learning approach for predicting the rate of penetration (ROP) during oil and gas drilling operations. They argued that existing models for predicting ROP are often inaccurate and unreliable, and that ensemble learning methods can provide a more accurate and practical solution. They collected data from drilling operations in the Gulf of Mexico and used it to train and test their hybrid ensemble learning model. The model consisted of several machine learning algorithms, including support vector regression (SVR), random forest regression (RFR), and gradient boosting regression (GBR), which were combined using a weighted average ensemble method. The authors compared the performance of their hybrid ensemble learning model to other machine learning models and found that it outperformed these models in terms of accuracy and reliability. The authors also conducted sensitivity analyses to determine the most important features for predicting ROP. They found that the weight on bit, the rotary speed, and the mud flow rate were the most important features for predicting ROP. Liu et al.[31] proposed a stacked generalization ensemble model for predicting the rate of penetration (ROP) in gas well drilling. The model is trained on a dataset of historical ROP data and drilling parameters from a shale gas survey well in Xinjiang, China. The model combined the predictions of six machine learning models: support vector regression (SVR), extremely randomized trees (XRT), random forest (RF), gradient boosting machine (GBM), light gradient boosting machine (LightGBM), and extreme gradient boosting (XGB). They first used Pearson correlation analysis to identify the most important features from the dataset. Then, they used a Savitzky-Golay smoothing filter to reduce noise in the dataset. Finally, they trained the stacked generalization ensemble model using the leave-one-out cross-validation method. The results showed that the stacked generalization ensemble model can significantly improve the accuracy of ROP prediction. The root mean square error (*RMSE*) of the model on the testing dataset is 0.4853 m/h, which is lower than the *RMSE* of any of the individual models. The model also has a high R^2 value of 0.9568. They also used the model to optimize the ROP parameters. They use particle swarm optimization (PSO) to search for the optimal combination of ROP parameters. The results show that the optimized ROP parameters can significantly improve the ROP. It was thus concluded that the stacked generalization ensemble model is a promising approach for predicting ROP in gas well drilling. The model is accurate and can be used to optimize the ROP parameters. Moraveji and Naderi[32]investigated the simultaneous effect of six variables on penetration rate using real field drilling data via response surface methodology (RSM). The important variables included well depth (*D*), weight on bit (WOB), bit rotation speed (*N*), bit jet impact force (IF), yield point, Y_p, to plastic viscosity ratio, PVR , (Y_p/PVR), 10 min to 10 s gel strength ratio (10MGS/10SGS). Equally, bat algorithm (BA) was used to identify optimal range of factors in order to maximize drilling rate of penetration. Their results indicated that the derived statistical model provides an efficient tool for estimation of ROP and determining optimum drilling conditions.

The aim of this study is to analyze the performance of machine learning and deep learning techniques in predicting the rate of penetration during drilling, which is crucial in optimizing drilling operations. The results of this study can contribute to drilling planning and optimization of future wells. Exact prediction of the rate of penetration during drilling will save the oil and gas industry a large amount of expenses during drilling operation and reduce the amount of non-productive time (NPT) encountered during drilling operation.

1.2 Approaches to Rate of Penetration Modelling

Over the past few years, a large amount of research has gone into ways in which ROP can be modelled with its dependent drilling parameters (controllable and uncontrollable). A key drive that leads to further research regarding this field is the non-comprehensiveness of previous models developed. This is because not all of the known ROP-affecting factors have been accounted for in a single model, which has led to poor accuracy and generalizability of the estimated models[33]. The seemingly large number of factors affecting the ROP and essential requirement for a model with high accuracy and reliable generalization has led to development of various ROP estimation models. An approach to carry out this modelling is hinged on two patterns, which are physics-based approach, and data-driven approach. The physics-based approach involves the use of mathematical modelling techniques to evaluate relationships between dependent parameter (ROP) and the independent parameters, so as to estimate accurate ROP values. These mathematical relationships are developed based on the physics of the borehole. There are various models used for ROP estimation that are created using the physics-based approach e.g., Cunningham model, Bingham[4] model, Maurer[19] model, Motahhari et al. model[33] and Hareland and Rampersad model[34]. The Cunningham model is given by:

$$R = KW_0N \qquad (1)$$

where R is the rate of penetration (ft/h), K the constant of proportionality, W_0 the threshold weight on bit (lb$_f$) and N rotary speed (rpm).

Bingham[4] model: $R = aN(W_{oB}/D_B)^b$ (2)

where W_{oB} is the weight-on bit (klb), D_B is the bit diameter (in), a and b are the dimensionless constants for each rock formation.

Maurer[18] model: $\frac{dF}{dt} = K\left[\frac{N(W - W_0)^2}{D^2 s^2}\right]$ (3)

where $\frac{dF}{dt}$ is the rate of penetration (ft/h), W the weight (Ib$_f$), s the confined rock strength (psi) and D the depth (ft).

Motahhari et al. model[33]: $R = w_f\left(\frac{GN^{\gamma}W_{oB}^{\alpha}}{D_B s}\right)$ (4)

where w_f is the dimensionless wear function, G is a model coefficient related to bit-rock interactions and bit geometry, α and γ are ROP model exponents. The bit coefficient, G, is determined by the bit design, cutter size, cutter rock friction coefficient and the bit geometry. In this model, a decrease in the value of the wear function, while keeping other model parameters constant leads to a decrease in ROP. In the case of the bit size or compressive strength, when its value is decreased an inverse occurs. The relationship between N, W_{oB} and R is non-linear. Hence, the exponents can yield an optimum value for W_{oB} and N due to the exponential nature of the relationship.

Hareland and Rampersad Model[34]: $R = 14.44N_cNA_v/D_B$ (5)

where N_c is the number of cutters and A_v the area of rock compressed ahead of a cutter (in^2).

Other models used for ROP estimation are as follows:

Bourgoyne and Young[3] model: $R = aN(W_{oB}/D_B)^b$ (6)

where W_{oB} is the weight-on bit (klb), D_B is the bit diameter (in), a and $b$are the dimensionless constants for each rock formation.

Bourgoyne et al.[35] aimed at seeking to optimize the controllable parameters during drilling operation. They proposed the development of an ROP model based on the application of multiple linear regression technique. The controllable parameters used in developing this model were eight: strength of formation, normal compaction function, weight on bit, bit teeth wear, rotary speed function, bit hydraulic function, differential pressure function, and under compaction function. These parameters were treated as independent parameters on the ROP (the dependent parameter). The developed model was then applied to estimate ROP for wells drilled vertically using roller cone bits, and it was concluded that the application of the ROP model could help reduce drilling operational cost by 10%. On inception, the model was basically created for modelling ROP for roller cone bits, but overtime has also shown effectiveness in modelling ROP for PDC bits. The Bourgoyne et al.[35] model is given by:

$$R = \prod_{i=1}^{8} F_i \qquad (7)$$

where $F_1\,(=e^{a_1})$ is the formation strength function for Bourgyone and Young model, $F_2\left[=e^{a_2(10000-D)}\right]$the normal compaction function for Bourgyone and Young model, $F_3\left\{=\exp\left[a_3D^{0.69}(g_p - 9.0)\right]\right\}$ the under compaction function for Bourgyone and Young model, $F_4\left\{=\exp\left[a_4D(g_p - \rho_c)\right]\right\}$ the pressure differential function for Bourgyone and Young model, $F_5\left[=\left\{\exp a_5 \hbar \left[\frac{(w/d)-(w/d)_t}{4-(w/d)_t}\right]\right\}\right]$ the weight on bit function for Bourgyone and Young model, $F_6\left\{=\exp\left[a_6 In(N/100)\right]\right\}$ the rotary speed function for Bourgyone and Young model, $F_7\left\{=\exp\left[a_7(-h)\right]\right\}$ the bit tooth wear function for Bourgyone and Young model and $F_8\left\{=\exp\left[a_8(pq/350\mu d_n)\right]\right\}$ the bit hydraulic function for Bourgyone and Young model.

The physics-based approach has limitations due to the failure to consider all the parameters affecting the drilling operation and in the choice of an empirical constant for the ROP estimation with respect to the well/borehole in operation. This gave rise to the use of data-driven approaches, which make use of data generated during drilling (Logging While Drilling (LWD)) and artificial intelligence techniques for ROP estimation[17]. The application of AI models for ROP estimation was suggested by Bilgesu et al.[10], so as to get over the weakness of the physics-based approach and improve the accuracy of ROP predictability.

2. Methodology

2.1 Methods

Figure 3 shows the proposed methodology, adopted in this study.

2.2 Data Collection

The data utilized for this study was obtained from the Daily Drilling Report (DDR) for an oil well in Nigeria. It contains parameters that ROP depends on, which will help make a robust model. Such parameters are weight on bit, pump flow rate, mud weight, mud type, drill bit diameter and wellbore trajectory, amongst others. After data collection, the uncertainties within the dataset and the suitable parameters are defined. This leads to filtration of the dataset. The well contains data of 27 columns (the number of variables), 17280 rows, 0% missing cells, and 0% duplicate rows.

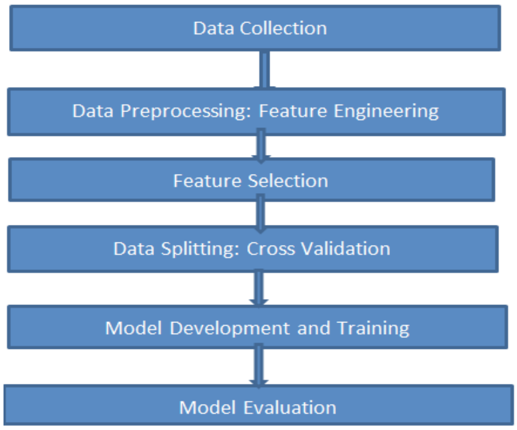

Data Collection
Data Preprocessing: Feature Engineering
Feature Selection
Data Splitting: Cross Validation
Model Development and Training
Model Evaluation

Figure 3: Methodology workflow.

2.3 Data Processing

The data preprocessing phase is also known as Feature Engineering Phase. The data set used for the study is subjected to various statistical manipulations and transformations in order to extract relationships and insights between parameters in the data and, process the data into forms that are more understandable by the algorithms, hence, producing better model performances. Such techniques include exploratory data analysis, missing data imputation, outlier handling, feature scaling, variable transformation, and discretization, amongst others. These processes help the model match key relationships between the input parameters and the target variable. In this study, the data preprocessing techniques used were outlier handling, variable transformation and feature scaling.

Outlier handling: This refers to the process involved in dealing with outliers found in a dataset. Outliers are simply data point that vary significantly to majority of the dataset. Outliers must be dealt with since they can significantly affect the outcomes and precision of statistical models. Outlier treatment can be done in a number of ways, such as by removing outliers, capping, or imputing more representative values. The method utilized in this study was the capping technique, which involved imputing the interquartile range of the variable with the outlier where the outliers are in the variable. Figures 4 and 5 show the box plot of ROP data before and after outlier.

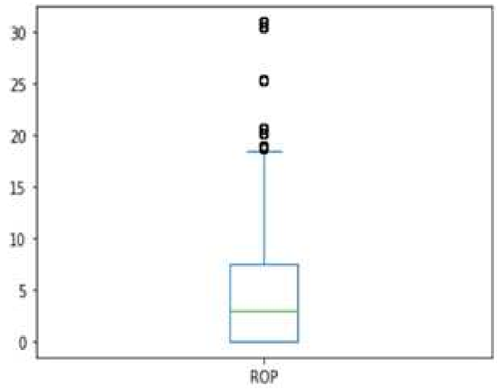


Figure 4: Box plot of ROP data before outlier.

Variable Transformation: This technique was employed in this study to treat the variables that were skewed either to the left or to the right. It involves changing a variable's scale or distribution to satisfy requirements or enhance the performance of statistical models. This preprocessing technique was performed on variables that were skewed either to the left or right, so as to equalize variances and establish linearization among the variables, which makes it easier to interpret and model. The variable transformation technique use in this study were LogTransformer and BoxCoxTransformer.

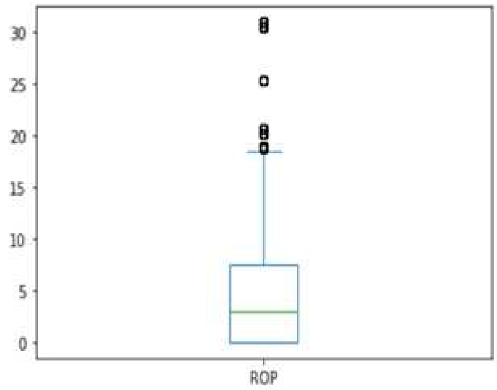


Figure 5: Box plot of ROP data after outlier removal. removal.

Feature Scaling: This involves changing the scale of numerical features in a dataset. To make the features similar and prevent some from predominating others based only on their initial scale, the range or distribution of the features must be changed. This preprocessing technique is so important as it helps the machine learning models to better understand the features as they will usually be within the range of 0 to 1, which the models usually prefer. For some particular models, it is a necessary requirement to perform feature scaling on the dataset before passing it into them e.g., ANN and CNN, while some models are not influenced when the dataset is scaled or not e.g., Random Forest and Extra Trees. There are many types of Feature scaling techniques e.g., Standard Scaler, MinMax Scaler and Robust scaling. Each of these techniques has their rules of engagement, so as to get better model performance. These rules depend on dataset and model to be used. In this study, the standard scaler was utilized so as to scale features to have a mean of 0 and a standard deviation of 1. Table 1 shows the features definition with the data types used.

Table 1: Features definition, unit, and data types.

Feature	Definition	Units	Data type
Depth	The actual depth at which the drilling is taking place.	m	Numerical
Lag Depth	Time delay or lag between the measured depth and the corresponding ROP value.	m	Numerical
WHO	Weight on String.	klb	Numerical
ROP	Rate of Penetration.	m/h	Numerical
RPM TURBIN	Turbine Speed.	rev/min	Numerical
Torque	Rotational force of drill string.	klb.ft	Numerical
SPP	Standpipe Pressure.	psi	Numerical
Flow In	flow rate of drilling. fluid pumped into the wellbore during drilling.	gpm	Numerical
Mw In	total volume of drilling mud pumped into the wellbore during a specific period of time.	pcf	Numerical
Mw out	total volume of drilling mud pumped out of a wellbore during a specific period of time.	pcf	Numerical
PIT#1	mud pit volume in the first mud pit or mud tank.	Bbl	Numerical
PIT#2	mud pit volume in the second mud pit or mud tank.	Bbl	Numerical

PIT#3	mud pit volume in the third mud pit or mud tank.	Bbl	Numerical
PIT#4	mud pit volume in the fourth mud pit or mud tank.	Bbl	Numerical
PIT#5	mud pit volume in the fifth mud pit or mud tank.	Bbl	Numerical
PIT#6	mud pit volume in the Sixth mud pit or mud tank.	Bbl	Numerical
TOT ACT	Total Actual Time.	Bbl	Numerical
Steel Volume	The volume of steel that is used or consumed during the drilling process.	Bbl	Numerical
Over pull	Additional force applied to the drilling assembly in order to increase the drilling efficiency.	klb	Numerical
Flow Paddle	Percentage of drilling fluid that circulates through the wellbore during drilling.	%	Numerical
Bit Position	It refers to the vertical depth at which the drilling bit is located within the wellbore.	m	Numerical
Hook Position	Vertical position of the drilling hook or traveling block.	m	Numerical
String Weight	Total weight of the drill string, including the drill pipe, bottom hole assembly (BHA), and any other components attached to it.	klb	Numerical
Drag	Resistance encountered by the drill string and drill bit as they are advanced through the formation.	klb	Numerical

Table 2 gives information on the statistic of ROP variable.

Table 2: Descriptive Statistics of ROP variable.

Statistic	Mean	Standard deviation	Minimum	25%	50%	75%	Maximum
ROP	3.964	4.317	0.000	0.000	2.880	7.410	18.525

Table 3 gives the Pearson correlation of the oil well features and their values.

Table 3: Pearson correlation of features with rate of penetration.

Features	Well data
Depth	-0.37
Lag depth	0.98
WHO	0.15
RPM TURBIN	0.57
Torque	0.70
SPP	0.64
Flow in	0.57
Mw in	-0.08
Mw out	0.20
PIT#1	0.24
PIT#2	-0.14
PIT#3	-0.10
PIT#4	-0.26
PIT#5	0.05
PIT#6	0.23
TOT ACT	-0.25
Steel volume	0.20
Flow paddle	0.75
Bit position	0.20
Hook position	0.26
String eight	-0.01
Drag	-0.53

The correlation heat map of the well data is depicted in Figure 6.

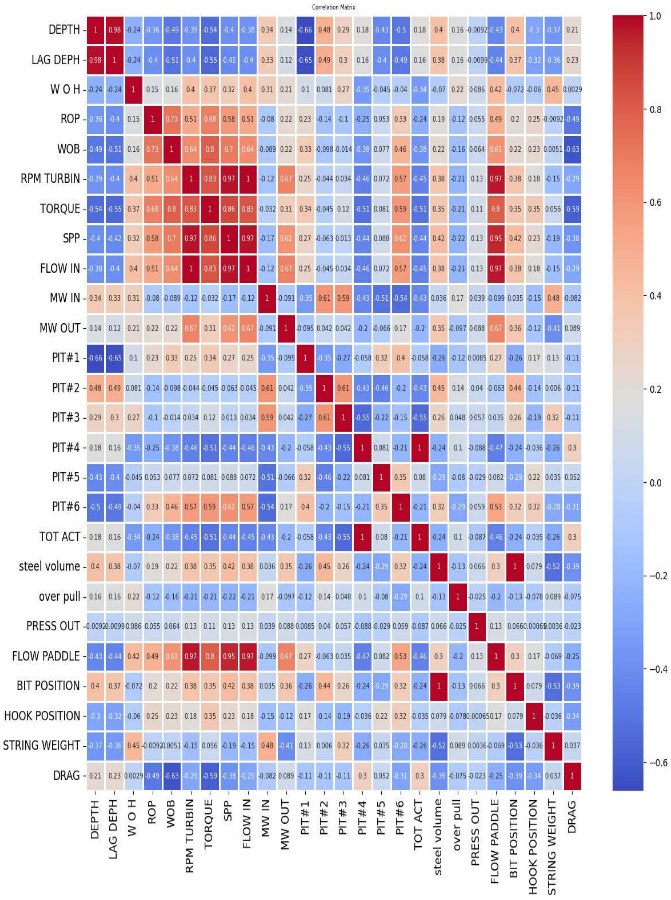


Figure 6: Correlation heat map of the well data.

2.4 Feature Selection

The defined input parameters from the dataset must pass through the feature selection phase. The feature selection is the process of selecting a subset of relevant features (variable, predictors) for usage in building machine learning algorithms. It involves selecting the pool of features that has significant impact on making prediction with the machine learning algorithm. It is a crucial phase, in the bid that a good machine learning model is developed. The feature selection algorithms are divided into three main categories: filter, wrapper, and embedded methods. The feature selection helps a user to better interpret the model e.g., a model of 10 input parameters is much easier to interpret than that of 100 parameters. It also shortens training time for the machine learning algorithm and enhances generalization by reducing overfitting. In this study, a filter method known as mutual information was used to select optimal features for model building. Mutual information is a statistical measure of the mutual dependence of 2 variables. In other words, mutual information quantifies the amount of information gained about one random variable through observing another random variable. The mutual information algorithm is given by:

$$I(X;Y)=\sum\sum p(x,y)\times\log\{p(x,\ y)[\ p(x)\times p(y]\} \quad (8)$$

where I is the ranking score, X and Y the respective input and output nodes, x and y are the dependent and target variables respectively.

The algorithm selects the highest-ranking features that best describes the target variable and separates them into percentiles e.g., 10th, 20th, 30th etc., depending on the highest ranking. In this study, the features in the top 50th (50 percentile), which translated to 13 features out of the possible 27, were selected to be used to the build the machine learning models. The features selected after future selection are depth, lag depth, WOB, SPP, MW IN, PIT#2, PIT#3, PIT#4, PIT#6, TOT ACT, steel volume, bit position and hunk position

2.5 Data Splitting

Data splitting (otherwise known as cross validation) is a process utilized in the building of artificial intelligence models. Here, data is partitioned into two or more ways to enable the model identify the patterns within the data set and predict its performance on unseen (real world) data. Two sets of the dataset are created: a training set and a testing set. The training set is used to train the artificial intelligence model on the data while the testing set is used to assess the model's performance in real world scenarios. This is because there is a probability that the built model may not be robust enough to perform successfully on unknown (real world) data. There are various methods used for cross validation operation viz. holdout method, K-fold method, Stratified K-fold method, Leave One-Out method, amongst others. The K-fold cross validation technique was implemented in this study using the Python Sklearn package.

A 60:40 split of the oil well data was made into train and test sets. There are 10368 rows and 13 columns in the training set and 6912 rows and 13 columns in the test set. The training results were obtained by training the model on the train data and using the resulting model to predict the training set. The test results were obtained by training the model on the train set before predicting the test set.

3. Model Development and Training

Seven machine learning techniques were analyzed in this work, to be trained to make predictions of the rate of penetration for the oil well. The machine learning models that were employed for this analysis are outlined as follows and their written codes can be found in the Appendix.

3.1 Random Forest Regression

Random forest can be applied to both classification and regression problems. It is an ensemble learning technique that creates a large number of decision trees during training period and utilizes averaging to improve the prediction accuracy and control over-fitting. Random forests are widely used for applications (such as credit scoring and spam filtering) because they can handle both categorical and continuous data. During training, random forests create a lot of decision trees[36]. Each tree is constructed using a random subset of the features and a sample of the training data. The individual decision trees predictions are combined by the random forest algorithm to provide a prediction. For a wide range of applications, random forests are a potent and useful machine learning technique. They are often good performers and are quite simple to teach and tune.

A schematic of the decision tree regression is depicted in Figure. 7.

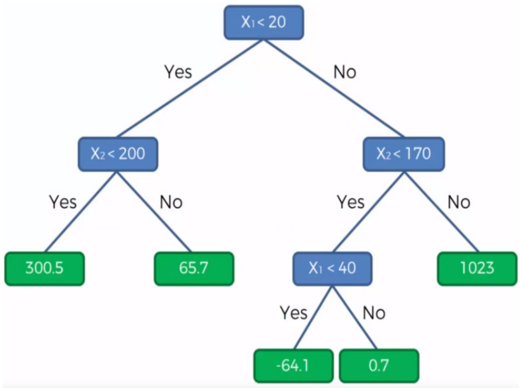


Figure 7: Decision tree regression schematic[37].

A random forest model works by training multiple decision trees in parallel and uses a bagging technique to obtain a robust model. Usually, machine learning models have hyperparameters, that is, parameters in the algorithm that are constant throughout training that help the algorithm better understand the data patterns. Hyperparameters in random forest algorithm include max_depth, max_features, min_samples_leaf, min_samples_split, n_estimators. To obtain optimal performance of the random forest algorithm, optimal values must be selected for these hyperparameters. To get the optimal values of the random forest algorithm, a hyperparameter search algorithm (known as the randomized search algorithm) must be used. This algorithm helps to generate the optimal hyperparameter value for the hyperparameter to be utilized in the model. After implementing randomized search algorithm on the well data, the optimal value of the hyperparameters were max_depth = 31, max_features = sqrt, min_samples_leaf = 3, min_samples_split = 13, n_estimators = 666.

3.2 Linear Regression

Linear regression is a supervised method of machine learning that uses one or more input features to predict a continuous target variable. It is assumed that there is a linear relationship between the input variables and the goal variables. Linear regression is intended to establish the optimal line according to the data, minimizing the difference in predicted and real values. The algorithm operates by generating the coefficients of the line's linear equation. Some hyperparameters in linear regression are copy_X and fit_Intercept. After implementing randomized search algorithm on the well data using linear regression as the base model, the optimal value of the hyperparameters were copy_X = True, and fit_Intercept = True.

3.3 KNearest Neighbor

KNearest Neighbor (KNN), as shown in Figure 8, is a supervised model-based machine learning technique that can be applied to both classification and regression models. KNN is not a parametric algorithm, meaning that it does not make any assumptions about the distribution of data. The KNN method is based on the hypothesis that similar occurrences will share similar labels. The KNN technique identifies the K closest neighbors to a given data point by reference to a distance metric, typically Euclidean, and assigns the label to the majority of these K neighbors for a given data point. When the algorithm is doing a regression, it takes the weighted average of all the target values from the K neighbor and uses it to predict the new value for the given data point. The number of neighbors is a hyperparameter that can be changed. Some hyperparameters in KNN algorithms are algorithm, leaf_size, p, weights, and n_neighbours. After implementing randomized search algorithm on the well data using KNearest neighbor as the base model, the optimal value of the hyperparameters were algorithm= auto, leaf_size = 10, p = 1, weights = distance and n_neighbours = 3.

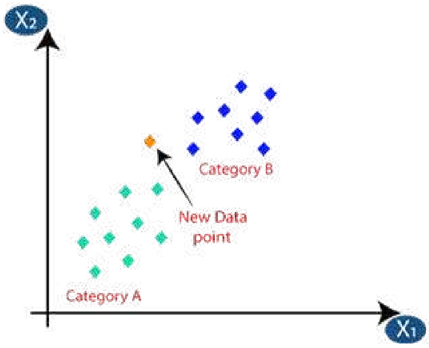


Figure 8: KNearest Neighbor[38].

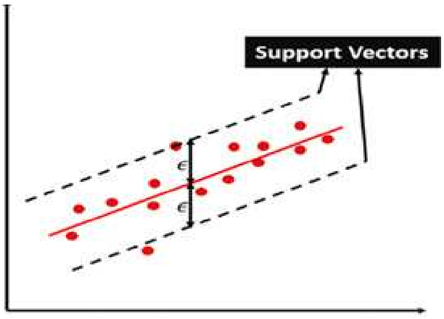


Figure 9: SVM schematic[39].

3.4 Support Vector Machine Regression

Support vector machine (SVM) regression is a supervised learning algorithm that is primarily used in classification tasks. It is derived from the concepts of support vector machines (SVM), as shown in Figure 9. The goal of SVM regression is to identify a function that best matches the relationship between the input and target variables. The SVM regression generates a high-dimensional hyperplane with each data point as a feature vector in the hyperplane space. The objective of the algorithm is to find the hyperplane with the greatest margin, i.e., the distance from the hyperplane to the nearest data point in each class. In the regression case, SVM chooses the hyperplane that contains the most data points within the given range. The range is the margin of tolerance, which allows some data points to fall outside of the range. The support vectors are the data points that fall within or cross the range. Some hyperparameters in support vector regression algorithm are *C*, epsilon, and kernel. After the implementation of the randomized search algorithm on the well data using support vector regression algorithm as the base model, the optimal values of the hyperparameters were *C* = 10, epsilon = 1 and kernel = rbf.

3.5 Stacking Technique

Stacking is a type of machine learning technique, whose algorithm is shown in Figure 10, that uses the predictive power of different machine learning algorithms to make better predictions on datasets. The stacking technique typically involves the use of base models and a meta model. The base models are usually common machine learning algorithms such as decision trees, random forests, and support vector machines. These base models are trained on a dataset and are used to make predictions; these predictions are then combined in a meta model, which can be linear regression or a neural network to make final predictions. It is a powerful machine learning technique since it utilizes the diverse knowledge of the base models. The base models used for this study are random forests, or support vector machines, linear regression, and nearest neighbors, while the meta model used is the linear regression model.

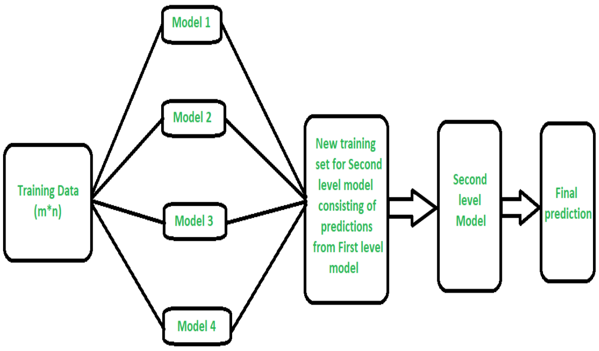

Model 1
Model 2
Model 3
Model 4
Training Data (m*n)
New training set for Second level model consisting of predictions from First level model
Second level Model
Final prediction

Figure 10: Stacking algorithm [40].

3.6 Voting Technique

Voting is a machine learning technique that involves the integration of predictions from multiple independent models to form a final prediction, as shown in Figure 11.

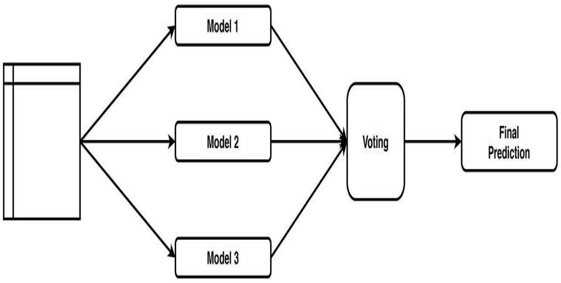


Figure 11: Voting Algorithm (LevelUpCoding).

Voting technique is commonly referred to as ensemble voting, or majority voting, and is based on the principle that the integration of the opinions of multiple models can often lead to greater prediction accuracy than the use of a single model. Under the Voting algorithm, each base model is trained on the same data set, but with different algorithms or settings. During the prediction phase, each base model makes its own prediction based on the data it has been trained on. Finally, the final prediction is calculated by adding up all the predictions using a voting system. The base models used for this study are random forests, or support vector machines, linear regression, and nearest neighbors, while the meta model used is the linear regression model.

3.7 Convolutional Neural Network

Convolution Neural Networks (CNNs) are a type of deep learning algorithm that is commonly employed in the analysis and interpretation of visual data, including images and videos. CNNs are widely used for image classification, object recognition and image segmentation. However, not only can CNNs be used for image classification, but they can also be used in regression-based projects, where it is purposed to predict continuous variables. A convolution neural network (CNN) usually consists of four components: convolutional layers, pooling layers, fully connected layers, and output layers, as shown in Figure 12. These four components usually make for the architecture of CNNs. The main difference between a CNN and a regression-based CNN is the output layer (output layer) and loss function (loss function). The output layer in a CNN based on regression is distinct from that of a Softmax-based CNN. Instead of predicting class probabilities using a function of a Softmax, an output layer is typically composed of an individual neuron with a function of a linear activation. This allows the network to produce a continuous value immediately as a regression prediction. For regression tasks, a loss function is often used to measure the difference between predicted and actual target values. Examples of loss functions that are commonly used include MSE (mean squared error) and MAE (mean absolute error).

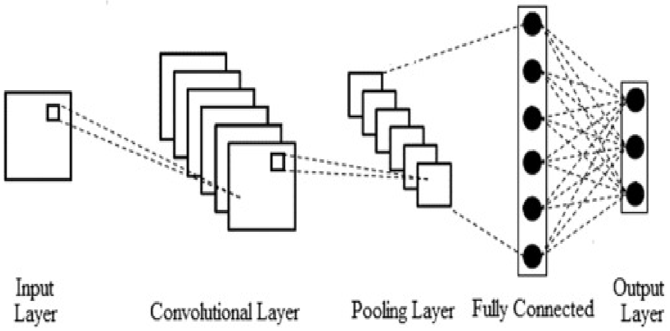


Figure 12: Convolutional neural network[39].

3.8 Model Evaluation

There are metrics usually used to reflect how well the model has learnt patterns in the data and the performance of the model on the unseen (test) data set. There are metrics used for evaluating the performance of machine learning models. These metrics show how far a model's prediction is from the true values. In this study, four error metrics are used to estimate a model performance on the learning patterns in the dataset and unseen data (test data). They are the mean absolute error (*MAE*), root mean squared error (*RMSE*), mean squared error (*MSE*) and coefficient of determination, R^2–score, given by Equations (9) to (12) respectively.

$$MAE = \sum_{i=1}^{n} \frac{|\hat{y} - y_i|}{n} \qquad (9)$$

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y} - y_i)^2}{n}} \qquad (10)$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\hat{y} - y_i}{y_i} \right| \qquad (11)$$

$$R^2 = \frac{\sum_{i=1}^{n} (\hat{y} - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \qquad (12)$$

where $\hat{y}$, y_i and $\bar{y}$ are the respective predicted, actual and mean values and n the number of observations.

4. Results and Discussion

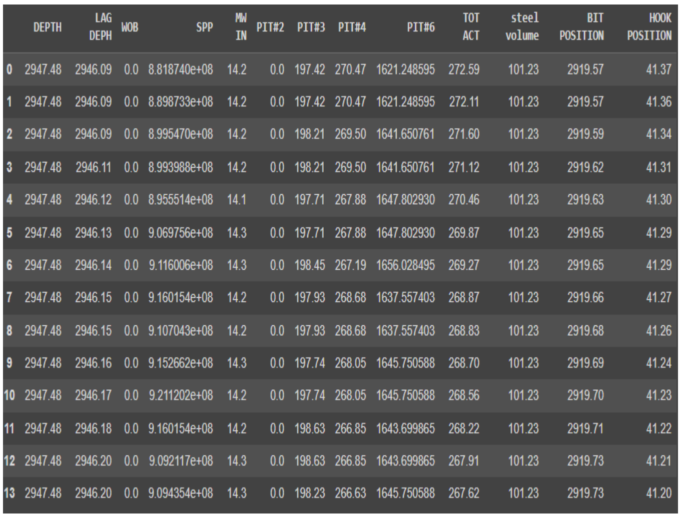
The well data after carrying out various statistical analyses, the features were reduced from the previous 27 columns to 13 columns, , as displayed in Table 4, which is an excerpt of the well data used for both training and testing. It shows a sample of the data utilized after feature selection has been carried out, leaving 17280 rows and 13 columns. These data were then separated using cross validation to train and test data respectively. The train data contained 10368 rows and 13 columns, while the test data contained 6912 rows and 13 columns. The linear regression model was applied to the training data and test data after the optimal hyperparameters had been generated. The train and test data were standardized such that data has a mean of 0 and standard deviation of 1. The results obtained using the linear regression model are presented in Table 5.

The random forest regression, KNearest neighbor, and support vector regression (SVR) model were applied to the training data and test data after the optimal hyperparameters had been generated using the randomized search cv algorithm. The train and test data were standardized such that data has a mean of 0 and standard deviation of 1. The random forest, KNearest neighbor, and support vector regression (SVR) models' results are presented in Tables 6–8 respectively.

Equally, the stacking and voting techniques were applied to the training data and test data after the optimal hyperparameters had been generated using the randomized search cv algorithm for the base model used in the technique. The train and test data were standardized such that data has a mean of 0 and standard deviation of 1. The results obtained using the stacking and voting techniques are presented in Tables 9 and 10 respectively.

The convolutional neural network (CNN) model was applied to the training data and test data using an epoch of 120 and a batch size of 32 together with an output layer of 1. The architecture of the CNN model created is as follows: two 1–D (one dimensional convolutional layers), filters (32 and 64), kernel size of two, one Global MaxPooling Layer, 5 hidden layers and 1 output layer. The train and test data were standardized such that data has a mean of 0 and standard deviation of 1. The results obtained using the CNN model are presented in Table 11.

Table 4: Sample taken from well data used to build ML models.



	DEPTH	LAG DEPH	WOB	SPP	MW IN	PIT#2	PIT#3	PIT#4	PIT#6	TOT ACT	steel volume	BIT POSITION	HOOK POSITION
0	2947.48	2946.09	0.0	8.818740e+08	14.2	0.0	197.42	270.47	1621.248595	272.59	101.23	2919.57	41.37
1	2947.48	2946.09	0.0	8.898733e+08	14.2	0.0	197.42	270.47	1621.248595	272.11	101.23	2919.57	41.36
2	2947.48	2946.09	0.0	8.995470e+08	14.2	0.0	198.21	269.50	1641.650761	271.60	101.23	2919.59	41.34
3	2947.48	2946.11	0.0	8.993988e+08	14.2	0.0	198.21	269.50	1641.650761	271.12	101.23	2919.62	41.31
4	2947.48	2946.12	0.0	8.955514e+08	14.1	0.0	197.71	267.88	1647.802930	270.46	101.23	2919.63	41.30
5	2947.48	2946.13	0.0	9.069756e+08	14.3	0.0	197.71	267.88	1647.802930	269.87	101.23	2919.65	41.29
6	2947.48	2946.14	0.0	9.116006e+08	14.3	0.0	198.45	267.19	1656.028495	269.27	101.23	2919.65	41.29
7	2947.48	2946.15	0.0	9.160154e+08	14.2	0.0	197.93	268.68	1637.557403	268.87	101.23	2919.66	41.27
8	2947.48	2946.15	0.0	9.107043e+08	14.2	0.0	197.93	268.68	1637.557403	268.83	101.23	2919.68	41.26
9	2947.48	2946.16	0.0	9.152662e+08	14.3	0.0	197.74	268.05	1645.750588	268.70	101.23	2919.69	41.24
10	2947.48	2946.17	0.0	9.211202e+08	14.2	0.0	197.74	268.05	1645.750588	268.56	101.23	2919.70	41.23
11	2947.48	2946.18	0.0	9.160154e+08	14.2	0.0	198.63	266.85	1643.699865	268.22	101.23	2919.71	41.22
12	2947.48	2946.20	0.0	9.092117e+08	14.3	0.0	198.63	266.85	1643.699865	267.91	101.23	2919.73	41.21
13	2947.48	2946.20	0.0	9.094354e+08	14.3	0.0	198.23	266.63	1645.750588	267.62	101.23	2919.73	41.20

Table 5: Linear regression model results.

Error metric	Training data	Test data
RMSE	2.611	2.565
MSE	6.819	6.582
MAE	1.773	1.744
R^2 Score	0.639	0.639

Table 6: Random Forest model results.

Error Metric	Training Data	Test Data
RMSE	0.469	0.676
MSE	0.220	0.458
MAE	0.207	0.300
Score R^2	0.988	0.975

Table 7: KNearest neighbor model results.

Error Metric	Training Data	Test Data
RMSE	0.523	0.724
MSE	0.309	0.524
MAE	0.201	0.239
Score R^2	0.984	0.971

Table 8: SVM model results.

Error Metric	Training Data	Test Data
RMSE	1.724	1.669
MSE	2.972	2.784
MAE	0.841	0.832
Score	0.843	0.847

Table 9: Stacking technique results.

Error Metric	Training Data	Test Data
RMSE	0.306	0.548
MSE	0.033	0.423
MAE	0.094	0.300
Score R^2	0.98	0.976

Table 10: Voting Technique results.

Error Metric	Training Data	Test Data
RMSE	0.803	0.826
MSE	1.167	1.331
MAE	0.646	0.681
Score R^2	0.938	0.926

Table 11: CNN results.

Error Metric	Training Data	Test Data
RMSE	0.797	0.751
MSE	1.167	1.331
MAE	0.636	0.564
Score R^2	0.924	0.928

From the results displayed in Tables 5–11, the stacking technique performed better than all the models and techniques employed in this study for the training data. Hence, the decreasing order of performance of the models for the training data is as follows: stacking technique > random forest model > KNearest neighbor model > CNN model > Voting technique > SVR model > linear regression model. In terms of the *RMSE*, the stacking technique was 35% better than the random forest model, 41% better than the KNearest neighbor model, 62% better than the CNN model and Voting technique, 82% better than the SVR model and 88% better than the linear regression model. In terms of the *MAE*, the stacking technique was 55% better than the Random Forest model, 53% better than the KNearest Neighbour model, 85% better than the CNN model and Voting technique, 89% better than the SVR model and 95% better than the linear regression model.

For the testing data, generalizing across the four metrics, the stacking technique yet again out-performed other models. It was only in terms of the *MAE* that the KNearest neighbor model outperformed the stacking technique by 20%, but in terms of the *RMSE*, *MSE*, R^2 Score, the stacking technique outperformed the KNearest Neighbour model. SVR model and linear regression model performed better on the test data compared to their performances on the train data, indicating generalization of the models and lack of overfitting on the training data.

In terms of the test (unseen) data, the stacking technique performed better than all the traditional ML models employed in this study. The next to it on the ranking of the model that best performed on the test data was the Random Forest model, followed by the KNearest Neighbour model, then the CNN model, then the Voting technique, then the SVR model and lastly the linear regression model.

In terms of the *RMSE*, the stacking technique was 19% better than the Random Forest model, 24% better than the KNearest Neighbour model, 27% better than the CNN model, 34% better than the Voting technique, 67% better than the SVR model and 79% better than the linear regression model.

In terms of the *MAE*, the stacking technique and the Random Forest model had the same performance score of 0.30. The stacking technique was still 47% better than the CNN model, 56% better than the Voting technique, 64% better than the SVR model and 83% better than the linear regression model.

Our findings in this investigation that the complex ML models of Stacking, Voting and CNN have the capacity to perform better than the traditional ML model was buttressed in the work of Burgos et al, which was equally corroborated in the study of Zhang et. al.[19], where the CNN model developed outperformed all the traditional ML models in terms of accuracy and reliability. It can equally be deduced from this study that irrespective of the architecture and predictive capacity of the ML model, traditional ML models, with proper feature engineering and hyperparameter tuning, can perform better than more complex machine learning models.

5. Conclusions

A comparative analysis of machine learning algorithms in predicting rate of penetration during drilling was carried out in this study. Data was obtained from the Daily Drilling Report (DDR) for an oil well. The well contains data of 17280 rows and 27 columns. The data preprocessing techniques of outlier handling, variable transformation and feature scaling were employed. Each of the seven machine learning techniques employed to predict the rate of penetration during drilling was able to extract meaningful information and patterns from the oil well data. However, some models outperformed other models by a distance, which reflects the predictive power of the algorithms. The capacity of the stacking algorithm to combine the predictive power of each base model gave it an edge over the rest of the models. The voting technique performed well, but not measured up to the performance of the stacking technique. Hence, the stacking technique is a more powerful ensembling technique than the voting technique. Amongst the base models, the random forest and KNearest Neighbors models are robust since they performed well on both the train and test data, while the SVM and linear regression models gave the highest errors on both the train and test data but they also showed their generalization capability and lower tendency to overfit. The CNN model has the capacity to perform well on regression-based task like rate of penetration predictions since it performed well on the test and train data.

Statements and Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this article.

Funding The authors received no specific funding for this work. Hence, the corresponding author confirms that there are no financial and personal relationships with other people or organizations that could inappropriately influence this study.

6. References

1. Azar HF, Saksala T, Jalali SME. Artificial neural networks models for rate of penetration prediction in rock drilling. J Structural Mechanics 2017;50(3):252-255.
2. Rupert JP, Padro CW, Blattel SR. The effects of weight material type and mud formulation on penetration rate using invert oil systems. Paper presented at the Society of Petroleum Engineers (SPE) Annual Technical Conference and Exhibition 1981.
3. Bourgoyne Jr AT, Young Jr FS. A multiple regression approach to optimal drilling and abnormal pressure detection. SPE J 1974;14(04):371-384.
4. Bingham MG. A new approach to interpreting rock drillability. Technical Manual Reprint Oil & Gas Journal 1965: 1-93.
5. Elkatatny S. Real time prediction of rheological parameters of KCl water-based drilling fluid using artificial neural networks. Arabian Journal for Science and Engineering 2017;42:1655-1665.
6. Mahmoud AA, Elkatatny S, Chen W, Abdulraheem A. Estimation of oil recovery factor for water drive sandy reservoirs through applications of artificial intelligence. Energies 2019;12(9):3671.
7. Connor Shorten “Machine Learning vs. Deep Learning” Towards Data Science. 2018.
8. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–444.
9. Otchere DA, Ganat TOA, Gholami R, Ridha S. Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models. Journal of Petroleum Science and Engineering 2021;200:108182.
10. Bilgesu HI, Tetrick LT, Altmis U, Mohaghegh S, Ameri S. A new approach for the prediction of rate of penetration (ROP) values. Paper presented at the Society of Petroleum Engineers (SPE) Eastern Regional Meeting 1997; SPE–39231–MS.
11. Arabjamaloei R, Shadizadeh S. Modeling and optimizing rate of penetration using intelligent systems in an Iranian southern oil field (Ahwaz oil field). Petroleum Science and Technology 2011;29(16):1637–1648.
12. Bataee M, Mohseni S. Application of artificial intelligent systems in ROP optimization: a case study in Shadegan oil field. Paper presented at the Society of Petroleum Engineers (SPE) Middle East Unconventional Gas Conference and Exhibition 2011;SPE-140029-MS.
13. Warren TM. Penetration-rate performance of roller-cone bits. SPE Drill Eng 1987;2(01):9–18.
14. AL-Mahasneh MA. Optimization Drilling Parameters Performance during Drilling in Gas Wells. International Journal of Oil, Gas and Coal Engineering 2017;5:19-26.
15. Amar K, Ibrahim, A. Rate of penetration prediction and optimization using advances in artificial neural networks, a comparative study. In Proceedings of the 4th International Joint Conference on Computational Intelligence 2012;1:647-652.
16. Shi X, Liu G, Gong X, Zhang J, Wang J, Zhang H. An efficient approach for real-time prediction of rate of penetration in offshore drilling. Mathematical Problems in Engineering 2016;(Article ID 3575380):1–13.
17. Ahmed A, Elkatatny S, Abdulraheem A, Mohammed M, Ali A, Mohamed I. Prediction of rate of penetration of deep and tight formation using support vector machine. In Proceedings of the SPE Kingdom of Saudi Arabia Annual Technical Symposium and Exhibition, Dammam, Saudi Arabia. 2018; SPE–192316–MS.
18. Maurer WC. The, “perfect-cleaning” theory of rotary drilling. J Pet Technol 1962;14(11):1270-1274.
19. Zhang Y, Zhang X, Chen Y. Deep neural networks for predicting rate of penetration in drilling. Journal of Petroleum Science and Engineering 2018;165:734-743.
20. Zhao Y, Noorbakhsh A, Koopialipoor M, Azizi A, Tahir MM. A new methodology for optimization and prediction of rate of penetration during drilling operations. Engineering with Computers 2020;36:587-595.
21. Abdulmalek A, Abdulwahab A, Salaheldin E, Abdulazeez A. New artificial neural networks model for predicting rate of penetration in deep shale formation. Sustainability 2019;11(22): 6527.
22. Hareland G, Hoberock LL. Use of drilling parameters to predict in-situ stress bounds. Paperpresented at the SPE/IADC Drilling Conference. Netherlands 1993:SPE-25727-MS.
23. Ashrafi SB, Anemangely M, Sabah M, Ameri MJ. Application of hybrid artificial neural networks for predicting rate of penetration (ROP): a case study from Marun oil field. Journal of Petroleum Science and Engineering 2019;175:604-623.
24. Iqbal F. Drilling optimization technique using real time parameters. SPE Russian Oil & Gas Technical Conference and Exhibition, Moscow, Russia, 2008.
25. Burgos CE, Zhang T, Li J, Zhang C, Chen S. ROP prediction using convolutional neural networks for Paleozoic shale drilling. Journal of Petroleum Science and Engineering 2019;17:633-641.
26. Monazami M, Hashemi A, Shahbazian M. Drilling rate of penetration prediction using artificial neural network: A case study of one of Iranian Southern oil fields. Journal of Oil and Gas Business 2012.
27. Abbas AK, Rushdi S, Alsaba M, Al Dushaishi MF. Drilling rate of penetration prediction of high-angled wells using artificial neural networks. J. Energy Resour. Technol 2019;141(11):112904.
28. Miyora TO. 2014. Modeling and optimization of geothermal drilling parameters: A case study of well MW-17 in Menengai Kenya, MS Thesis. University of Iceland 2014.
29. Al-AbdulJabbar A, Elkatatny S, Mahmoud AA, et al. Prediction of the rate of penetration while drilling horizontal carbonate reservoirs using the self-adaptive artificial neural networks technique. Sustainability 2020;12(4):1376.
30. Wang K, Zhang Y, Zhang X, Wang Y. A hybrid ensemble learning approach for rate of penetration prediction in oil and gas drilling. Journal of Petroleum Science and Engineering 2020;194:107424.
31. Liu N, Gao H, Zhen Z, Hu Y, Duan L. A stacked generalization ensemble model for optimization and prediction of the gas well rate of penetration: a case study in Xinjiang. Journal of Petroleum Exploration and Production Technology 2021;6:1595-1608.
32. Moraveji MK, Naderi M. Drilling rate of penetration prediction and optimization using response surface methodology and bat algorithm. Journal of National Gas Science and Engineering 2016;31:829–841.
33. Motahhari HR, Hareland G, Nygaard R, Bond B. Method of optimizing motor and bit performance for maximum ROP. J Can Pet Technol 2009;48(06):44-49.
34. Hareland G, Rampersad PR. Drag - Bit Model Including Wear. America/Caribbean Petroleum Engineering Conference 1994: SPE-26957-MS.
35. Bourgoyne Jr AT, Millheim KK, Chenevert ME, Young Jr FS.

Applied drilling engineering. SPE Textbook Series 1991;2:ISBN: 978-1-55563-001-0.

36. Quinlan JR. Induction of decision trees. Machine Learning 1986;1(1):81-106.

37. SametGirgin, Decision Tree Regression in 6 Steps with Python, PursuitData (Medium). 2019.

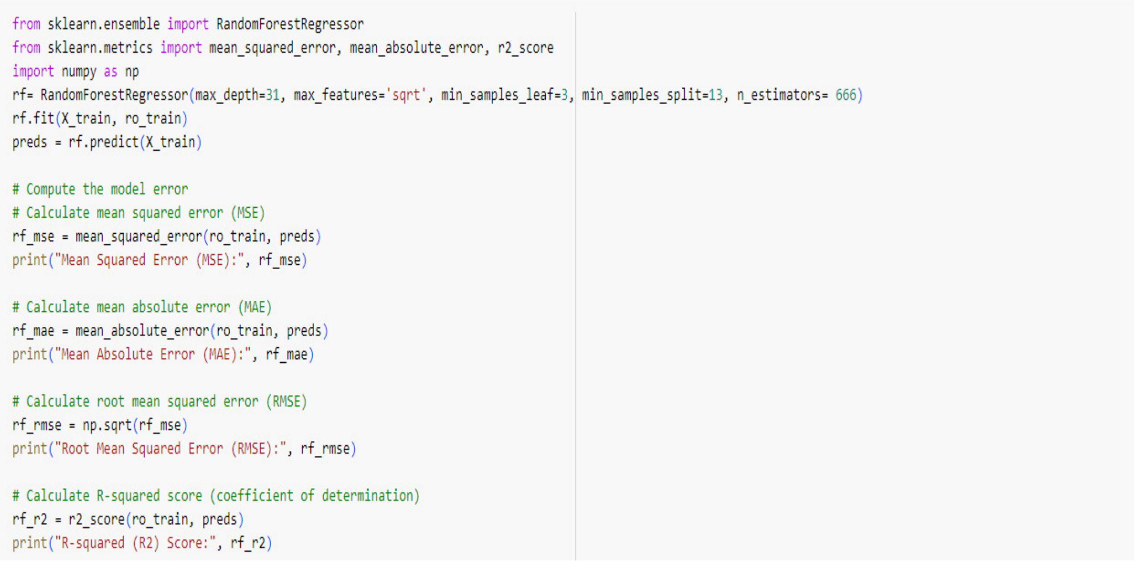
38. Javat (2022).

39. Pandey YN, Rastogi A, Kainkaryam S, Bhattacharya S, Saputelli L. Overview of Machine Learning and Deep Learning Concepts. Machine Learning in the Oil and Gas Industry 2020:75-152.

40. GeeksForGeeks (2022)

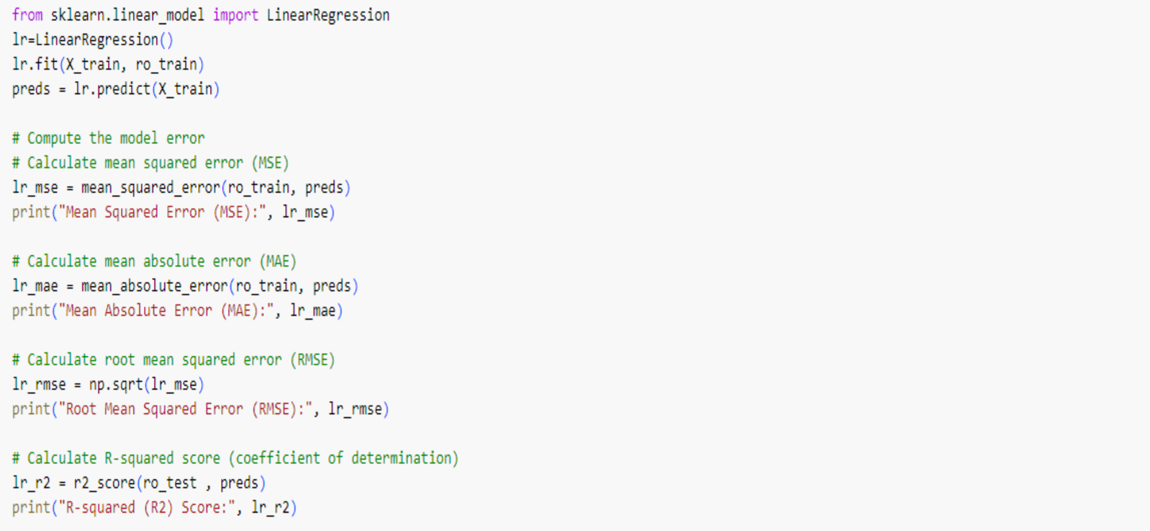
7 Appendix: Codes for the different algorithms employed in this work.

Random Forest Algorithm



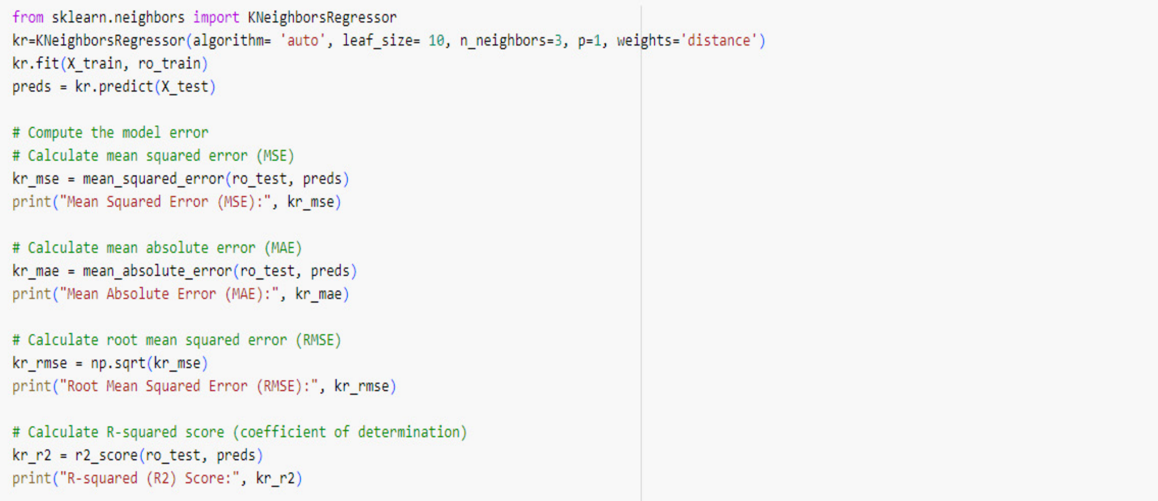
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np
rf= RandomForestRegressor(max_depth=31, max_features='sqrt', min_samples_leaf=3, min_samples_split=13, n_estimators= 666)
rf.fit(X_train, ro_train)
preds = rf.predict(X_train)

# Compute the model error
# Calculate mean squared error (MSE)
rf_mse = mean_squared_error(ro_train, preds)
print("Mean Squared Error (MSE):", rf_mse)

# Calculate mean absolute error (MAE)
rf_mae = mean_absolute_error(ro_train, preds)
print("Mean Absolute Error (MAE):", rf_mae)

# Calculate root mean squared error (RMSE)
rf_rmse = np.sqrt(rf_mse)
print("Root Mean Squared Error (RMSE):", rf_rmse)

# Calculate R-squared score (coefficient of determination)
rf_r2 = r2_score(ro_train, preds)
print("R-squared (R2) Score:", rf_r2)
```

Linear Regression Algorithm



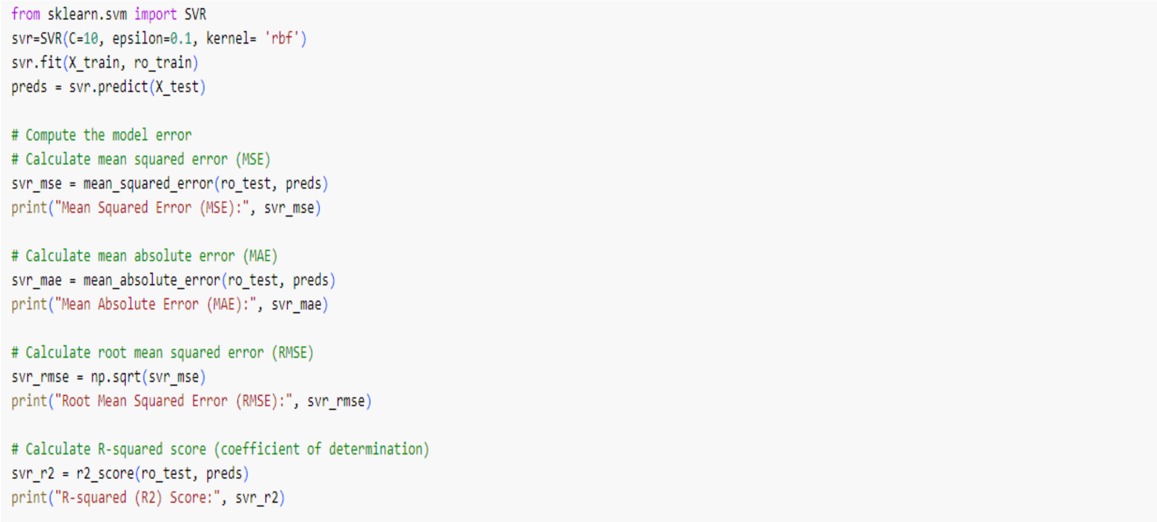
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(X_train, ro_train)
preds = lr.predict(X_train)

# Compute the model error
# Calculate mean squared error (MSE)
lr_mse = mean_squared_error(ro_train, preds)
print("Mean Squared Error (MSE):", lr_mse)

# Calculate mean absolute error (MAE)
lr_mae = mean_absolute_error(ro_train, preds)
print("Mean Absolute Error (MAE):", lr_mae)

# Calculate root mean squared error (RMSE)
lr_rmse = np.sqrt(lr_mse)
print("Root Mean Squared Error (RMSE):", lr_rmse)

# Calculate R-squared score (coefficient of determination)
lr_r2 = r2_score(ro_test , preds)
print("R-squared (R2) Score:", lr_r2)
```

K Nearest Neighbour Algorithm



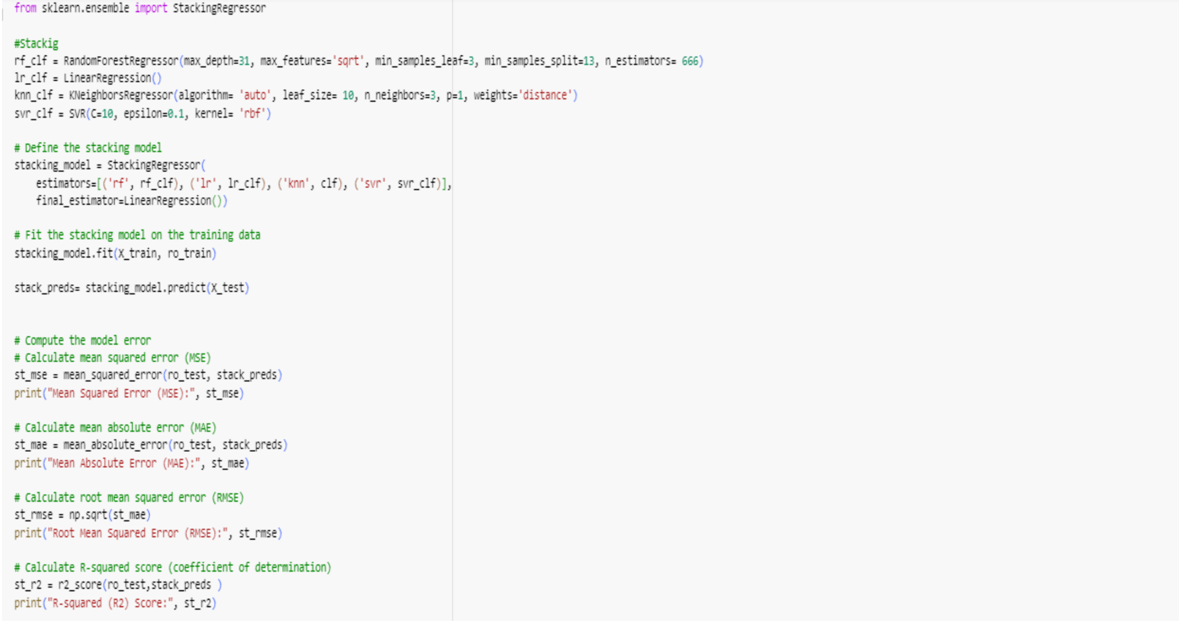
```
from sklearn.neighbors import KNeighborsRegressor
kr=KNeighborsRegressor(algorithm= 'auto', leaf_size= 10, n_neighbors=3, p=1, weights='distance')
kr.fit(X_train, ro_train)
preds = kr.predict(X_test)

# Compute the model error
# Calculate mean squared error (MSE)
kr_mse = mean_squared_error(ro_test, preds)
print("Mean Squared Error (MSE):", kr_mse)

# Calculate mean absolute error (MAE)
kr_mae = mean_absolute_error(ro_test, preds)
print("Mean Absolute Error (MAE):", kr_mae)

# Calculate root mean squared error (RMSE)
kr_rmse = np.sqrt(kr_mse)
print("Root Mean Squared Error (RMSE):", kr_rmse)

# Calculate R-squared score (coefficient of determination)
kr_r2 = r2_score(ro_test, preds)
print("R-squared (R2) Score:", kr_r2)
```

SVR Algorithm



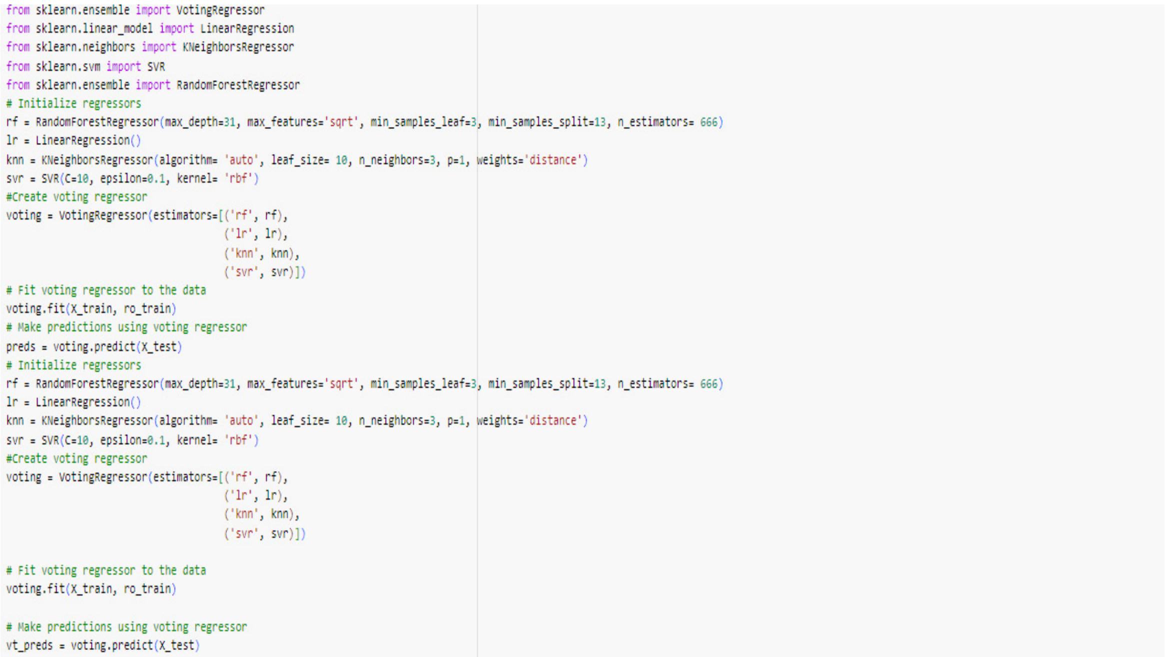
```
from sklearn.svm import SVR
svr=SVR(C=10, epsilon=0.1, kernel= 'rbf')
svr.fit(X_train, ro_train)
preds = svr.predict(X_test)

# Compute the model error
# Calculate mean squared error (MSE)
svr_mse = mean_squared_error(ro_test, preds)
print("Mean Squared Error (MSE):", svr_mse)

# Calculate mean absolute error (MAE)
svr_mae = mean_absolute_error(ro_test, preds)
print("Mean Absolute Error (MAE):", svr_mae)

# Calculate root mean squared error (RMSE)
svr_rmse = np.sqrt(svr_mse)
print("Root Mean Squared Error (RMSE):", svr_rmse)

# Calculate R-squared score (coefficient of determination)
svr_r2 = r2_score(ro_test, preds)
print("R-squared (R2) Score:", svr_r2)
```

Stacking Algorithm



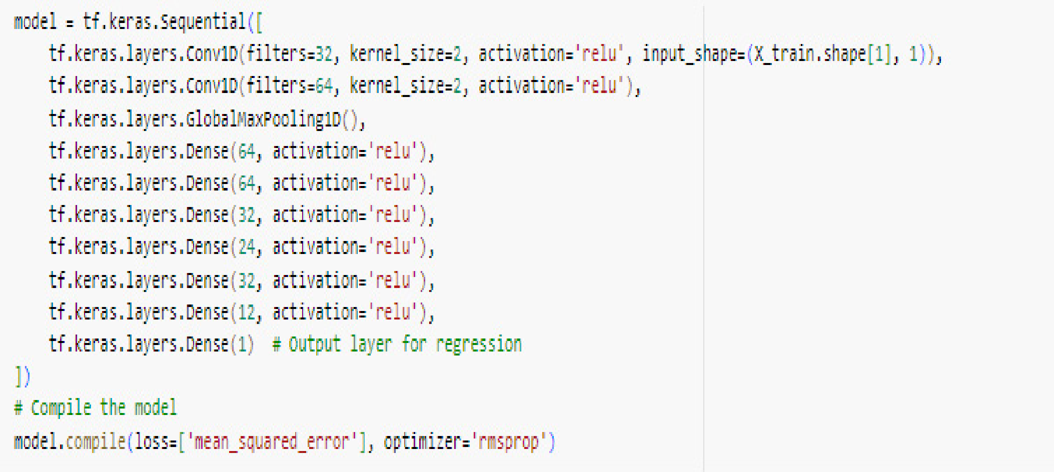
```
from sklearn.ensemble import StackingRegressor

#Stackig
rf_clf = RandomForestRegressor(max_depth=31, max_features='sqrt', min_samples_leaf=3, min_samples_split=13, n_estimators= 666)
lr_clf = LinearRegression()
knn_clf = KNeighborsRegressor(algorithm= 'auto', leaf_size= 10, n_neighbors=3, p=1, weights='distance')
svr_clf = SVR(C=10, epsilon=0.1, kernel= 'rbf')

# Define the stacking model
stacking_model = StackingRegressor(
    estimators=[('rf', rf_clf), ('lr', lr_clf), ('knn', clf), ('svr', svr_clf)],
    final_estimator=LinearRegression())

# Fit the stacking model on the training data
stacking_model.fit(X_train, ro_train)

stack_preds= stacking_model.predict(X_test)


# Compute the model error
# Calculate mean squared error (MSE)
st_mse = mean_squared_error(ro_test, stack_preds)
print("Mean Squared Error (MSE):", st_mse)

# Calculate mean absolute error (MAE)
st_mae = mean_absolute_error(ro_test, stack_preds)
print("Mean Absolute Error (MAE):", st_mae)

# Calculate root mean squared error (RMSE)
st_rmse = np.sqrt(st_mae)
print("Root Mean Squared Error (RMSE):", st_rmse)

# Calculate R-squared score (coefficient of determination)
st_r2 = r2_score(ro_test,stack_preds )
print("R-squared (R2) Score:", st_r2)
```

Voting Algorithm

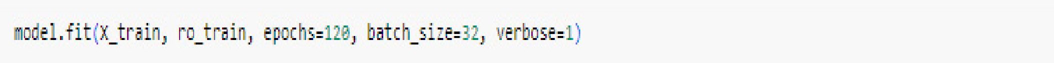


```
from sklearn.ensemble import VotingRegressor
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
# Initialize regressors
rf = RandomForestRegressor(max_depth=31, max_features='sqrt', min_samples_leaf=3, min_samples_split=13, n_estimators= 666)
lr = LinearRegression()
knn = KNeighborsRegressor(algorithm= 'auto', leaf_size= 10, n_neighbors=3, p=1, weights='distance')
svr = SVR(C=10, epsilon=0.1, kernel= 'rbf')
#Create voting regressor
voting = VotingRegressor(estimators=[('rf', rf),
                                     ('lr', lr),
                                     ('knn', knn),
                                     ('svr', svr)])
# Fit voting regressor to the data
voting.fit(X_train, ro_train)
# Make predictions using voting regressor
preds = voting.predict(X_test)
# Initialize regressors
rf = RandomForestRegressor(max_depth=31, max_features='sqrt', min_samples_leaf=3, min_samples_split=13, n_estimators= 666)
lr = LinearRegression()
knn = KNeighborsRegressor(algorithm= 'auto', leaf_size= 10, n_neighbors=3, p=1, weights='distance')
svr = SVR(C=10, epsilon=0.1, kernel= 'rbf')
#Create voting regressor
voting = VotingRegressor(estimators=[('rf', rf),
                                     ('lr', lr),
                                     ('knn', knn),
                                     ('svr', svr)])

# Fit voting regressor to the data
voting.fit(X_train, ro_train)

# Make predictions using voting regressor
vt_preds = voting.predict(X_test)
```

CNN Algorithm Code



```
model = tf.keras.Sequential([
    tf.keras.layers.Conv1D(filters=32, kernel_size=2, activation='relu', input_shape=(X_train.shape[1], 1)),
    tf.keras.layers.Conv1D(filters=64, kernel_size=2, activation='relu'),
    tf.keras.layers.GlobalMaxPooling1D(),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(24, activation='relu'),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(12, activation='relu'),
    tf.keras.layers.Dense(1)  # Output layer for regression
])
# Compile the model
model.compile(loss=['mean_squared_error'], optimizer='rmsprop')
```



```
model.fit(X_train, ro_train, epochs=120, batch_size=32, verbose=1)
```