Estimation of TBM Penetration rate using Gradient Boosting-based Algorithms

Ebrahim Ghorbani  
Nazarbayev University

Amoussou Coffi Adoko  
Nazarbayev University

Saffet Yagiz (✉️ saffet.yagiz@nu.edu.kz)  
Nazarbayev University

Research Article

**Keywords:** Tunneling, TBM performance, Rate of Penetration, Machine Learning, Gradient Boosting, Artificial Intelligence

**Posted Date:** August 8th, 2023

**DOI:** https://doi.org/10.21203/rs.3.rs-3228718/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License

**Additional Declarations:** No competing interests reported.
Estimation of TBM Penetration rate using Gradient Boosting-based Algorithms

Ebrahim Ghorbani¹, Amoussou Coffï Adoko², Saffet Yagiz³*

¹ School of Mining and Geosciences, Department of Mining Engineering, Nazarbayev University, KZ

Abstract

TBM performance prediction from the rate of penetration (ROP) point of view has yet to draw a lot of attention since it is one of the main challenges for mechanized excavation with tunnel boring machines (TBMs). In this study, five algorithms, Gradient Boosting (GB), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), AdaBoost (AB), and CatBoost (CB) have been conducted to predict the ROP based on the Gradient Boosting theory. Six tunnel cases from different projects were examined to obtain the aim of the research. Dataset developed using those tunnel datasets includes Uniaxial compressive strength (UCS), Rock Type, Distance between Plane of Weakness (DPW), and TBM-related parameter of thrust force (TF). Mentioned Gradient Boosting algorithms were performed to obtain the most accurate results for the study. The developed models showed that XGBoost outperformed the other models, followed by the CatBoost model according to seven different evaluation metrics used to rank the models. After parameter tuning, the GB model outperformed others while those were not improved very much. By using the overall ranking according to the metrics and considering the parameter tuning time, XGBoost and CatBoost presented the first two best performances. Through SHAP values and dependency plots, the features and importance of the inputs showed the TF has the highest impact on the ROP, followed by UCS, Rock Type, and

* Corresponding author.

Corresponding author: Ebrahim Ghorbani, amoussou.adoko@nu.edu.kz

Email Addresses: e.ghorbani1991@gmail.com; https://orcid.org/0000-0003-0735-4824 (E. Ghorbani), amoussou.adoko@nu.edu.kz; https://orcid.org/0000-0003-1396-7811 (A. C. Adoko), saffet.yagiz@nu.edu.kz; https://orcid.org/0000-0002-7271-3136 (S. Yagiz).
DPW. It is concluded that the XGBoost and CatBoost algorithms could be used for modeling to obtain the TBM penetration for similar rock types.

**Keywords:** Tunneling; TBM performance; Rate of Penetration; Machine Learning; Gradient Boosting; Artificial Intelligence

**Article Highlights**

- Gradient Boosting-based algorithms may show high accuracy due to their modeling process involving previous weak predictions.
- The extreme Gradient Boosting technique provides high performance in the TBM penetration rate estimation among the selected ones.
- Among the parameters involved in the model development for TBM penetration rate estimation, TBM thrust force has the highest impact on the prediction.

**1 Introduction**

Tunnel boring machines (TBMs) have been one of the most promising techniques to excavate tunnels and install the supporting system simultaneously with a very fast construction rate. While the TBM has displayed huge advantages, estimation of its performance from different points of view, such as the estimation of its rate of penetration (ROP), the advance rate (AR), and cutter tools wear prediction is crucial. This is due to the fact that their performance may be varied fast since it is a colossal project which involves sundry parameters, get involve many different parties and stages, and needs huge investments, so it requires to be evaluated consistently during the construction (Oreste, 2006; Vergara & Saroglou, 2017). Some of the earliest studies to address these issues can be traced back to classic TBM performance modeling from empirical and analytical to statistical ones, such as the NTNU model developed in 1976 to estimate the net penetration.
rate (Bruland, 1999), the Colorado School of Mines (CSM) model (Rostami & Ozdemir, 1993) which then modified by Yagiz (Yagiz, 2002; Yagiz et al., 2009), using the brittleness index (BI), the distance between weakness planes (DPW) and the angle between the weakness planes (α) in the model. Barton also developed a model, named \( Q_{TBM} \), to predict the ROP and AR incorporating Q-system and average cutter force in link with the strength of the rock mass, in which the joint structure and rock compressive or tensile strength, cutter life index (CLI), rock stress level are also accounted for (Barton, 1999).

One of the most crucial parameters which should be estimated, or predicted by the TBM performance models is the rate of penetration. The different definitions of the ROP have been developed decades ago up to now. Generally, ROP is defined as the ratio of the distance of the tunnel excavated to the time of operation for the excavation during the construction of the tunnel and shows as follows in units of meters per hour (m/h) or millimeters per minute (mm/min):

\[
ROP = \frac{\text{The distance excavated}}{\text{TBM operating time}} = \frac{L}{T_b} \quad \text{Eq.1}
\]

In other words, the ROP may obtain based on the distance excavated per the revolution of the TBM cutter head and stands as an instantaneous rate of penetration or the averaged ROP over each cycle of thrust cylinder, and it is defining as \( \text{Eq.2} \) where \( ROP_{\text{rev}} \) denotes the rate of penetration per revolution of the cutter head in mm/rev, and the RPM is the rate of cutter head revolutions per minute expressed in rev/min.

\[
ROP_{\text{rev}} = \frac{1000 \times ROP}{60 \times \text{RPM}} \quad \text{Eq.2}
\]

Over the past five decades, a series of studies have been conducted for the estimation of TBM ROP and the development of various equations to calculate this critical parameter. However, the tunneling industry has demanded such research due to the variabilities that
exist in individual tunnel projects. On the other hand, due to the involvement of TBM tunneling with so many parameters, developing performance models based on absolute mathematical methods may not meet the maximum accomplishment of mechanized tunneling, so it needs to use advanced ways, which be able to take into account different types of influencing parameters. To be a little more specific to show such complexities, look at the parameters of tunneling projects which affect the TBM performance (Deketh et al., 1998; Den Hartog et al., 1997).

The response to the demand has been increasing worldwide since computer-based and programming-based techniques, like artificial intelligence (AI), came to the attention of researchers and industries; that started in the early 21st century when Grima Alvarez employed artificial neural networks (ANN) and adaptive neuro-fuzzy inference system (ANFIS) to predict TBM performance (Grima et al., 2000). These techniques have drawn various attention where other researchers also used to estimate TBM performance by employing different input parameters including rock type, uniaxial compressive rock strength (UCS), Brazilian tensile strength (BTS), cutter life index (CLI), Young’s modulus, BI, DPW, α, RQD, percentage of Quartz, RMR, TBM thrust and torque, joints spacing and conditions (Js, Jc), punch slope index (PSI), cohesion, internal friction angle, Poisson’s ratio, density, RPM, cutter torque (CT), thrust force (TF), and AR (Afradi et al., 2019; Benardos, 2008; Eftekhari et al., 2010; Gholami et al., 2012; Gholamnejad & Tayarani, 2010; Oraee et al., 2012; Salimi & Esmacili, 2013; Torabi et al., 2013; Yagiz et al., 2009; Zhu et al., 2021); the prediction performance of such models is varied from 0.69 to 0.939 from the R-squared point of view. Or using other derived algorithms based on neural networks (NN) like deep NN (DNN) (Koopialipoor et al., 2019), probabilistic NN (PNN) (Harandizadeh et al., 2021), back-propagation NN (BPNN) (Yan et al., 2023), convolutional NN (CNN) (L. Li et al., 2022). Employing different algorithms by introducing evolutionary optimization methods is the other employing of AI in the TBM
performance prediction that has been utilized by various researchers. It started by using particle swarm optimization (PSO) employed by Yagiz and Karahan (Yagiz & Karahan, 2011), and went further by them applying the Differential Evolution algorithm and Gray Wolf optimizer to predict the ROP by using the parameters of DPW, α, UCS, and BI (Yagiz & Karahan, 2015). Other optimization techniques plus their combination with different algorithms have got the researchers’ attention, where Armaghani et al. used the combined methods of PSO-ANN and Imperialism Competitive Algorithm (ICA)-ANN to predict the rate of penetration (Armaghani et al., 2017). Using optimization methods has gone further so that various techniques either individually or by combining with other algorithms have been employed to develop models for estimation for ROP; such methods include biogeography-based optimization, moth flame optimization, social spider optimization, multiverse optimization combined with extreme Gradient Boosting (XGBoost) (Zhou et al., 2021), and grasshopper optimization (Akbarzadeh et al., 2022). Other AI techniques have been utilized to predict the TBM performance are support vector machine-based techniques such as support vector regression (SVR) (Mahdevari et al., 2014; Salimi et al., 2016; Yang et al., 2020), least-squares support vector machine (LS-SVM) (Ge et al., 2013), and finally the algorithm based on decision trees algorithm including classification and regression tree (CART) (Salimi et al., 2019). Obviously, there are some other techniques, particularly two or several algorithms called “hybrid methods” employed to predict the ROP, which is a summary of previously developed models and their input parameters presented in Table 1.

<table>
<thead>
<tr>
<th>AI technique(s)</th>
<th>Input Parameters</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>RPM, TF, Dc</td>
<td>(Grima et al., 2000)</td>
</tr>
<tr>
<td>ANFIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENN</td>
<td>UCS, BI, α, Js</td>
<td>(Zhao et al., 2007)</td>
</tr>
<tr>
<td>PSO</td>
<td>UCS, BTS, BI, DPW, α</td>
<td>(Yagiz &amp; Karahan, 2011)</td>
</tr>
<tr>
<td>LS-SVM</td>
<td>UCS, BTS, PSI, DPW, α</td>
<td>(Ge et al., 2013)</td>
</tr>
<tr>
<td>SVR</td>
<td>DPW, α, BTS, BI, UCS, TF, CT, CP, SE</td>
<td>(Mahdevari et al., 2014)</td>
</tr>
<tr>
<td>MVR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>GWO</td>
<td>ICANN</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>GWO</td>
<td>ICA-ANN</td>
<td>ANN</td>
</tr>
<tr>
<td>GEP</td>
<td>DNN</td>
<td>RQD, RMR, UCS, BTS, RPM, TF, WZ</td>
</tr>
<tr>
<td>RVR</td>
<td>CART</td>
<td>UCS, DPW, RQD</td>
</tr>
<tr>
<td>GWO-FW-MKL-SVR</td>
<td>BBO-FW-MKL-SVR</td>
<td>CEP, TF, CT, RPM C, ϕ, Es, the ratio of the boulder, UCS, RQD</td>
</tr>
<tr>
<td>MKL-SVR</td>
<td>SVR</td>
<td>RQD, RMR, UCS, BTS, RPM, TF, WZ</td>
</tr>
<tr>
<td>LSTM</td>
<td>GMDH</td>
<td>TF, CT, RPM, the earth pressure at the middle left and right side of the chamber</td>
</tr>
<tr>
<td>GOA-SVM</td>
<td>Copula-BPNN</td>
<td>UCS, BTS, PSI, DPW, α</td>
</tr>
<tr>
<td>LSTM-GWO</td>
<td>PLSR-BRT</td>
<td>UCS, BTS, PSI, DPW, RFC, α</td>
</tr>
<tr>
<td>BRT</td>
<td>PLSR-BPN</td>
<td>UCS, BTS, E0, Kv, υ, CAI, TF, RPM</td>
</tr>
<tr>
<td>SVR</td>
<td>ANN</td>
<td></td>
</tr>
</tbody>
</table>

**ANN**: artificial neural network, **ANFIS**: adaptive neuro fuzzy inference system, **ENN**: ensemble neural networks, **PSO**: particle swarm optimization, **LS-SVM**: least-squares support vector machine, **SVR**: support vector regression, **MVR**: Multivariate Regression Composer, **DE**: Differential Evolution, **HS**: Hybrid Harmony Search, **BFGS**: Broyden, Fletcher, Goldfarb, and Shanno optimization method, **GWO**: Grey Wolf Optimizer, **FCM**: Fuzzy C-means, **ICA**: imperialism competitive algorithm, **GEP**: gene expression programming, **DNN**: deep neural networks, **RVR**: relevance vector regression, **SVM**: support vector machine, **CART**: classification and regression tree, **FW-MKL**: feature weighted-multiple kernel, **BBO**: biogeography-based optimization, **RVM**: relevance vector machine, **XGBoost**: extreme gradient Boosting, **MFO**: moth flame optimization, **SSO**: social spider optimization, **SCA**: sine cosine algorithm, **MVO**: multi verse optimization, **MPMR**: minimax probability machine regression, **ELM**: extreme learning machine, and **FN**: functional network, **HENSM**: hybrid ensemble model, **LSTM**: long short-term memory networks, **FL**: fuzzy logic, **HSA**: harmony search algorithm, **GOA**: grasshopper optimization algorithm, **BPNN**: back-propagation neural network, **CNN**: convolutional Neural Network, **PLSR-BRT**: partial least squares regression- Boosted regression trees. **BI**: brittleness index, **Js**: Joints spacing, **CLI**: cutter life index, **Jc**: Joint conditions, **C**: cohesion, **ϕ**: internal friction angle, **υ**: Poisson’s ratio, **WZ**: Weathered Zone, **Jv**: Joints volume, **PSI**: Peak Slope Index,
While the prediction performance of all developed models is acceptable, the implementation and performance of some of those are not obvious as mathematical regression, so such models may be inferred as “black box” machine learning (ML) algorithms. However, what is important is one would be able to explain and interpret the developed model which is taken place within the model, as call the “explainability” and “interpretability” of the ML model. Having this in mind, it may be said that the algorithms based on decision trees and gradient Boosting are the favorable ones to model the ROP. The former can be illustrated on a tree to reach the target value, and the latter works in a specific manner in which the algorithms use weak learners in each iteration incorporated decision trees and go to the next one by having such information updating the input and values to estimate the final value for the target parameter. This provides an opportunity to develop a more explainable and interpretable, on the other words, “white box” ML model developed to predict the target feature. Provided the review and the summary of previously developed models, we can see only in a few articles that it has been tried to employ the gradient-based algorithms in the estimation of ROP, while these methods have drawn a lot of attention in the different fields due to their high-performance results and considerable accuracy. On the other hand, there are bagging-based algorithms that may be rivaled by gradient-boosting ones. While they are both ensemble models, their working processes are different from the modeling building, in which the bagging builds multiple models independently with parallel, then gives an averaged or voted best model (Sagi & Rokach, 2018), however, the gradient-based models try to reduce the prediction errors within its iteration process of building models in a sequence manner, in which by considering previous results, the next iteration would be updated to decrease the error of the prediction (Schapire & Freund, 2013). In summary, their main differences came into account from
the different points of view including models building approach, training method, training
dataset handling, and predictions combination. Overall, although there is no specific way
to say which the two algorithms would show better performance, it can be said that
bagging is more useful for a dataset that is prone to high variance and the target is to
decrease the model prediction variance (Grandvalet, 2004). In contrast, the gradient-
boosting approaches have shown acceptable outcomes in problems interacting with a
dataset having features with complex and non-linear relationships, especially in regression
ones (Xu et al., 2021).

In this paper, five algorithms based on gradient boosting are used to predict the rate of
penetration including Gradient Boosting, Extreme Gradient Boosting, Light Gradient
Boosting Machine, CatBoost, and AdaBoost. Firstly, a brief overview of the selected
techniques is described, and the employed dataset description and preprocessing on it are
summarized. In the following steps, the results of the developed models according to seven
evaluation metrics have been assessed, and the best one was determined based on a ranking
process. Then, parameter tuning of the models was performed, and the best model from an
overall point of view (combined based on the results derived after and before hyper-
parameter implementation plus the tuning time) was discerned. In the next step, the most
influencing features on the target parameter have been identified using SHAP values and
dependency plots. Finally, the finding have been compared with the literature.

2 Methods

2.1 An Overview of Selected Techniques

2.1.1 Gradient Boosting

Gradient Boosting (GB) is used for both classification and regression problems since it is
an ensemble training strategy (Friedman, 2001; Natekin & Knoll, 2013). The main
advantage of the GB is to combine the weak learners, which are the regular decision trees,
and make a model able to predict with more precision. The main concept of gradient
Boosting is that the modeling is starting with an initialization, then the sequent models
justify according to the previous models’ errors, in which the gradient of the loss function
is computed based on the ensemble predictions, depicting the direction and the size of the
update needed to be reached to the minimization of the loss function. Then, each of the
models is trained to minimize the updated loss function to be fitted the negative gradient.
Another advantage of prioritizing the GB and GB-based methods is preventing overfitting
by employing the regularization techniques. In the whole process, the effect of each
developed model is reduced by shrinkage, and the updated model based on new features,
is selected at random according to feature importance. The final ensemble prediction is
produced by combining all of the predicting models (Freund et al., 1999). Further details
on the algorithm process can be found in the literature (Friedman, 2001).

2.1.2 Extreme Gradient Boosting
One of the most well-known algorithms developed based on the GB is the Extreme
Gradient Boosting (XGBoost) presented in 2015 which added more efficiency and model
prediction accuracy to the GB (Chen et al., 2015). Able to employ different loss functions,
XGBoost provides adaptability to sundry fields. With the capability to handle the relevant
loss functions for classification and regression problems, it has been able to train the
objectives effectively. Another advantage of XGBoost, like GB, is using regularization
techniques, such as L1 and L2, leading to improving the modeling performance and
preventing overfitting, by handling the complexities and reducing the impact of noisy and
irrelevant features. Moreover, it uses a productive process in tree construction, called
approximation tree learning, by concentrating on the most suitable splits for trees’
conditions rather than considering all possible divides. In addition, the Boosting round and
adaptive stopping criteria are provided in the algorithm (Chen & Guestrin, 2016). To
enhance the performance, XGBoost checks it on the testing dataset simultaneously with
the training steps, and the modeling is stopping when the condition will be met, or retake
the training process with the update on the number of iterations according to the last
performance observed. Another feature of this algorithm is parallelization, which used
tasks divided among several CPU cores to speed up the training process, particularly, when
there is a big dataset. In addition to the mentioned feature, one can be employed it on the
GPU leading to more modeling speed. The feature importance also can be analyzed easier
by the tools provided in the technique. Ultimately, XG proposes various loss functions,
tries to avoid overfitting by using regularization techniques, can build trees fast, supports
adaptive stopping criteria, allows for parallelization and GPU employing, and provides
options for model interpretability. The enthusiastic readers are able to find more details of
the approach in the mentioned references.

2.1.3 Light Gradient Boosting Machine
Another algorithm developed based on the GB is Light Gradient Boosting Machine
(LightGBM) which is most suitable for large-scale datasets (Ke et al., 2017). To handle
such big and high-dimensional datasets, it uses two gradient-based techniques of One-Side
Sampling (GOSS) and Exclusive Feature Bundling (EFB) (Zhang & Gong, 2020). GOSS
provides the algorithms with the ability to reduce the number of data points required to
reach an accurate model, by focusing on points with bigger gradients in the training
process. On the other hand, EFB enhances the training speed by feature grouping based on
their comparable values, lets using less memory. Furthermore, LightGBM uses a leaf-wise
tree building, in which it selects the leaf node having the largest gain and provides a
balanced and productive tree building. This leads to the decrease of tree depth, speeding
up the training process and improving the model performance. The "max_depth"
parameter has been introduced to handle the complexity and set a limit on tree depth. Like
the XGBoost, this technique also uses L1 and L2 regularization to prevent the model from
overfitting (Wen et al., 2021); furthermore, LightGBM includes integrated support for
parallel and distributed computing, enabling effective training on big datasets employing several CPU cores (Meng et al., 2021). The referenced publications give the working process of the technique.

2.1.4 AdaBoost

The ensemble learning technique known as AdaBoost (AB), or Adaptive Boosting, has become quite well-liked in the machine learning community. AdaBoost, which was proposed by Freund and Schapire in 1996, tries to build a strong classifier by repeatedly combining weak classifiers (Freund & Schapire, 1996). The AdaBoost basic principle is to emphasize the challenging examples by giving misclassified instances in each iteration a higher weight. AdaBoost successfully learns to concentrate on the instances that are difficult to classify correctly by iteratively training weak classifiers on the reweighted data. The weighted predictions of the weak classifiers are combined to create the final strong classifier. AdaBoost has several restrictions, though. Its sensitivity to noisy data and outliers could result in a drop in performance. Additionally, the caliber of the weak classifiers used has a significant impact on AdaBoost performance. To get the best results, weak learners must be carefully chosen and designed. AdaBoost has been widely used and has displayed outstanding performance in a variety of applications despite its drawbacks. Its success can be ascribed to its capacity to make use of numerous weak learners and adaptively distribute example weights. A number of tasks, including face detection, object recognition, and gene expression analyses, have demonstrated the algorithm's efficiency. The readers may find further details in the referenced article above.

2.1.5 CatBoost

CatBoost (CB) is an algorithm developed by Yandex, a Russian company, in 2017, according to the gradient Boosting and used a symmetric decision tree (DT) shown in Fig 1 (Dorogush et al., 2018; Prokhorenkova et al., 2018). It claims that able to give better results with a high speed without needing huge time-consuming data preprocessing
compared to the above gradient-based techniques. The method can manage different types of data from homogeneous, like sound, text, or video, to heterogeneous like tabular datasets used to predict a particular parameter according to observed or measured databases. This can model the outcome even with a small dataset, for example, with only 90 datasets. CatBoost can manage the categorical and even text parameters effectively in either classification or regression problems. It employs a method named ordered goal encoding technique, in which the categorical variables get numerical values based on their goal statistics and according to their appearance on the datasets. Such a technique removes time-consuming, intensive pre-processing like one-hot encoding to manage the categorical parameters which are used in the previously described GB-based algorithms (Yandex, 2017). The detailed working process of the CatBoost has provided in the references (Dorogush et al., 2018; Prokhorenkova et al., 2018).

2.2 Data Description

The dataset used in this study is from two groups including six tunnel projects of the Queens Water tunnel in New York City, US, the Milyang Dam and hydro-tunnel in South Korea, the Manapouri power station in New Zealand, and three Iranian tunnels including Karaj, Zargroos, and Ghomrood water conveyance tunnels. The main specifications of TBM used in the projects are presented in Table 2, and the input parameters for the ROP prediction models include three groups of (1) intact rock properties of UCS; (2) rock mass characteristics of Rock Type, and DPW (m); and (3) TBM operation parameters of thrust

![Diagram](image-url)
force (TF). To reduce the complexities of the modeling process, rock types are considered as five categories as presented in Table 3; sample data points and the descriptive statistics of features may be seen in Table 4 and Table 5, respectively. The scatter matrix analysis chart, as shown in Fig 2, reveals the correlation between the input features in the database as well as the relationship between those and the output: ROP. Additionally, Fig 3 presents the violin plots displaying the distribution of each variable, offering valuable insights into the presence of possible outliers. To use the dataset in the selected machine learning techniques, the values of the features have been normalized because they are from different tunnel projects. The normalization technique employed is the Min-Max approach (MinMaxScaler in Python).

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Queens</th>
<th>Manapouri</th>
<th>Milyang</th>
<th>Iranian (Karaj, Zagross, Ghomrood)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine Type</td>
<td>Open Beam</td>
<td>Open Beam</td>
<td>Open Beam</td>
<td>Double-shield</td>
</tr>
<tr>
<td>Diameter (m)</td>
<td>7.1</td>
<td>10.05</td>
<td>2.6</td>
<td>6.73</td>
</tr>
<tr>
<td>No. of Cutters</td>
<td>50</td>
<td>68</td>
<td>22</td>
<td>42</td>
</tr>
<tr>
<td>Cutter Type</td>
<td>DISC</td>
<td>DISC</td>
<td>DISC</td>
<td>DISC</td>
</tr>
<tr>
<td>Cutter Diameter (mm)</td>
<td>482.6</td>
<td>432</td>
<td>393.7</td>
<td>432</td>
</tr>
<tr>
<td>Cutter Tip Width (mm)</td>
<td>19.05</td>
<td>19.05</td>
<td>15.875</td>
<td>19</td>
</tr>
<tr>
<td>Maximum Cutter Load (kN)</td>
<td>311</td>
<td>267</td>
<td>178</td>
<td>267</td>
</tr>
<tr>
<td>Total Installed Thrust (kN)</td>
<td>15568</td>
<td>18150</td>
<td>38700</td>
<td>28134</td>
</tr>
<tr>
<td>Total Installed Power (kW)</td>
<td>3148</td>
<td>3450</td>
<td>630</td>
<td>2100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Tunnel(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category 1</td>
<td>Rhyodacite dike</td>
</tr>
<tr>
<td>Category 2</td>
<td>Granitoid (felsic) Gneiss and Orthogneiss</td>
</tr>
<tr>
<td>Category 3</td>
<td>massive Garnet Amphibolite and larger Mafic dikes</td>
</tr>
<tr>
<td>Category 4</td>
<td>Mafic- to Mesocratic Orthogneiss</td>
</tr>
</tbody>
</table>

Table 3. Rock-type categories were used in the modeling according to the types in the studying tunnels.

Quartzite, Sandstone, Shale, Sandstone-Slate-Phyllite, Siliceous tuff, Vitric tuff, Vitric tuff-Sandstone, Sandstone-Siltstone, Biotite-rich and fine to medium grained granite;
<table>
<thead>
<tr>
<th>Tunnel</th>
<th>UCS (MPa)</th>
<th>Rock Type</th>
<th>DPW (m)</th>
<th>TF (kN)</th>
<th>ROP (m/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queens</td>
<td>131.47</td>
<td>Category 5</td>
<td>0.05</td>
<td>6455.01</td>
<td>2.08</td>
</tr>
<tr>
<td>Queens</td>
<td>185.40</td>
<td>Category 3</td>
<td>1.60</td>
<td>18876.37</td>
<td>1.51</td>
</tr>
<tr>
<td>Manapouri</td>
<td>138.6</td>
<td>Category 5</td>
<td>0.8</td>
<td>23210</td>
<td>0.69</td>
</tr>
<tr>
<td>Manapouri</td>
<td>120.8</td>
<td>Category 5</td>
<td>0.8</td>
<td>22720</td>
<td>1.75</td>
</tr>
<tr>
<td>Iranian-Karaj</td>
<td>100</td>
<td>Category 1</td>
<td>0.1</td>
<td>3010</td>
<td>5.68</td>
</tr>
<tr>
<td>Iranian-Karaj</td>
<td>75</td>
<td>Category 1</td>
<td>0.1</td>
<td>2370</td>
<td>4.18</td>
</tr>
<tr>
<td>Iranian-Zagroos</td>
<td>70</td>
<td>Category 5</td>
<td>0.4</td>
<td>5660</td>
<td>2.48</td>
</tr>
<tr>
<td>Iranian-Zagroos</td>
<td>25</td>
<td>Category 5</td>
<td>0.2</td>
<td>3210</td>
<td>2.41</td>
</tr>
<tr>
<td>Iranian-Ghomrood</td>
<td>50</td>
<td>Category 1</td>
<td>0.20</td>
<td>4260.00</td>
<td>3.11</td>
</tr>
<tr>
<td>Iranian-Ghomrood</td>
<td>60</td>
<td>Category 5</td>
<td>0.80</td>
<td>6450.00</td>
<td>2.13</td>
</tr>
<tr>
<td>Milyang</td>
<td>250</td>
<td>Category 4</td>
<td>1.6</td>
<td>3300</td>
<td>0.78</td>
</tr>
<tr>
<td>Milyang</td>
<td>210</td>
<td>Category 4</td>
<td>0.4</td>
<td>3300</td>
<td>1.12</td>
</tr>
</tbody>
</table>

Table 4. Sample dataset used for predicting the rate of penetration.

<table>
<thead>
<tr>
<th>ROP (m/h)</th>
<th>UCS (MPa)</th>
<th>DPW (m)</th>
<th>TF (kN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>579</td>
<td>579</td>
<td>579</td>
</tr>
<tr>
<td>mean</td>
<td>1.94</td>
<td>153.0664</td>
<td>0.941883</td>
</tr>
<tr>
<td>std</td>
<td>1.05</td>
<td>64.97552</td>
<td>0.677572</td>
</tr>
<tr>
<td>min</td>
<td>0.22</td>
<td>20</td>
<td>0.05</td>
</tr>
<tr>
<td>25%</td>
<td>1.23</td>
<td>119.6241</td>
<td>0.4</td>
</tr>
<tr>
<td>50%</td>
<td>1.76</td>
<td>149.7973</td>
<td>0.8</td>
</tr>
<tr>
<td>75%</td>
<td>2.39</td>
<td>190.3</td>
<td>1.6</td>
</tr>
<tr>
<td>max</td>
<td>5.88</td>
<td>300</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5. Descriptive statistics for the employed dataset of all projects.
Fig 2. Scatterplot matrix of selected features in the dataset alongside their correlation, as well as their histogram and distribution curve.
Fig 3. Violin plots of each input parameter (except for “Rock Type” which is a categorical feature) are involved in the employed dataset to model the algorithms.

2.3 Model Evaluation Metrics

For the evaluation of the developed models, various metrics have been employed not only to assess the individual models but also to rank those to find the best one from different points of view. The indices have been utilized shown in Table 6.

Table 6. Evaluation metrics are used to assess the developed models.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Equation</th>
<th>Performance indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-squared</td>
<td>$R^2$</td>
<td>$R^2 = 1 - \frac{\sum (y_m - \hat{y}_p)^2}{\sum (y_m - \bar{y}_m)^2}$</td>
</tr>
<tr>
<td>Adjusted R-squared</td>
<td>$A.R^2$</td>
<td>$A.R^2 = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$</td>
</tr>
<tr>
<td>Root Mean Square Error</td>
<td>RMSE</td>
<td>$RMSE = \sqrt{\frac{\sum (y_p - \hat{y}_m)^2}{N}}$</td>
</tr>
<tr>
<td>Mean Absolute Error</td>
<td>MAE</td>
<td>$MAE = \frac{1}{N} \sum</td>
</tr>
<tr>
<td>Mean Squared Logarithmic Error</td>
<td>MSLE</td>
<td>$MSLE = \frac{1}{N} \sum \log(y_m + 1) - \log(y_p + 1)$</td>
</tr>
<tr>
<td>Variance Account For</td>
<td>VAF</td>
<td>$VAF = 1 - \frac{\text{Var}(y_m - y_p)}{\text{Var}(y_m)}$</td>
</tr>
</tbody>
</table>
### Median Absolute Error

\[ \text{MedAE} = \text{median}(|y_{m1} - y_{p1}|, \ldots, |y_{mn} - y_{pn}|) \]

Smaller values are better

\( y_m \): measured values, \( y_p \): predicted values, \( N \): total dataset size, \( p \): number of independent variables, \( \text{Var} \): Variance

### 3 Results and Discussion

#### 3.1 Modeling Results

For estimation of the rate of penetration of hard rock TBM according to the available dataset, five algorithms derived from gradient Boosting theory were employed. In the first step, all models were trained with their default parameters, and then their performance has been evaluated based on the seven mentioned evaluation indices. Table 7 and Table 8 show the results obtained for the developed models without implementing hyper-parameter using 80% of the dataset as the training dataset and 20% for the testing dataset. In this study, to improve the generalization of the modeling, the default method (\text{random_state}) has not employed for the splitting process, instead, by dividing the dataset into various segments and applying \text{custom_train_test_split} for each segment, the possible sensitivity of the model has been already avoided. In addition, such a method leads to enhancing the stability of the developed model by doing train and testing on different data partitions. Ultimately, it prevented the model from data leakage by concatenating the testing and training datasets of each segment separately the same, keeping the testing dataset of each segment separately avoiding the potential bias during the model evaluation.

The selected segment size in this process is 100 leading to four segments as there are 579 data points in the original dataset. The reason for the selection of such a segment size came from trial and error attempts with different sizes: 10-110. The size of 100 (17% of the dataset) showed better and more plausible results compared to the others from different points of view: R2 and RMSE. In addition, selecting a segment size too low can lead to more biased in the splitting dataset. It should be mentioned that in this model development to justify one categorical feature in the dataset (Rock Type) the “One-hot Encoded” was employed. Meanwhile, since CatBoost models can handle categorical features, the Rock
Type parameter was used in the model as a categorical feature, in which a training “Pool” is defined during the CatBoost model development via introducing the categorical.

**Table 7. Evaluation metrics of the developed models for training dataset (without hyper-parameter).**

<table>
<thead>
<tr>
<th>Model</th>
<th>R2</th>
<th>A.R2</th>
<th>RMSE</th>
<th>MAE</th>
<th>MSLE</th>
<th>VAF</th>
<th>MedAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GB</td>
<td>0.8319</td>
<td>0.8314</td>
<td>0.0729</td>
<td>0.0543</td>
<td>0.0029</td>
<td>0.8319</td>
<td>0.0401</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.9896</td>
<td>0.9895</td>
<td>0.0181</td>
<td>0.0094</td>
<td>0.00023</td>
<td>0.9896</td>
<td>0.0047</td>
</tr>
<tr>
<td>LightGBM</td>
<td>0.9472</td>
<td>0.9471</td>
<td>0.0409</td>
<td>0.0291</td>
<td>0.00093</td>
<td>0.9472</td>
<td>0.0206</td>
</tr>
<tr>
<td>AB</td>
<td>0.6322</td>
<td>0.6301</td>
<td>0.0108</td>
<td>0.0841</td>
<td>0.00627</td>
<td>0.6426</td>
<td>0.0678</td>
</tr>
<tr>
<td>CB</td>
<td>0.9763</td>
<td>0.9762</td>
<td>0.0274</td>
<td>0.0188</td>
<td>0.00044</td>
<td>0.9762</td>
<td>0.0139</td>
</tr>
</tbody>
</table>

**Table 8. Evaluation metrics of the developed models for testing dataset (without hyper-parameter).**

<table>
<thead>
<tr>
<th>Model</th>
<th>R2</th>
<th>A.R2</th>
<th>RMSE</th>
<th>MAE</th>
<th>MSLE</th>
<th>VAF</th>
<th>MedAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GB</td>
<td>0.8048</td>
<td>0.8032</td>
<td>0.0518</td>
<td>0.0486</td>
<td>0.0025</td>
<td>0.8058</td>
<td>0.0364</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.9866</td>
<td>0.9864</td>
<td>0.0165</td>
<td>0.0093</td>
<td>0.00021</td>
<td>0.9866</td>
<td>0.0049</td>
</tr>
<tr>
<td>LightGBM</td>
<td>0.9411</td>
<td>0.9406</td>
<td>0.0345</td>
<td>0.0256</td>
<td>0.00077</td>
<td>0.9411</td>
<td>0.0196</td>
</tr>
<tr>
<td>AB</td>
<td>0.5554</td>
<td>0.5481</td>
<td>0.0949</td>
<td>0.0749</td>
<td>0.00531</td>
<td>0.5792</td>
<td>0.0613</td>
</tr>
<tr>
<td>CB</td>
<td>0.9740</td>
<td>0.9738</td>
<td>0.0229</td>
<td>0.0168</td>
<td>0.00037</td>
<td>0.9740</td>
<td>0.0131</td>
</tr>
</tbody>
</table>

At first glance, it is noticed that the XGBoost model provides the best results with an overall rank of 70 since its evaluation metrics show better performance and accuracy than the others. However, to determine developed models' ranking based on their performance, a ranking process of 5 to 1 has been employed which shows the best performance according to the performance indicators of individual metrics: rank “5” is the best, and rank “1” displays the least one. Then the summation of the assigned ranking to each index was obtained (for example, for the sake of ranking of CatBoost based on the training dataset we have ranks of 4 for R2, 4 for A.R2, 4 for RMSE, 4 for MAE, 4 for MSLE, 4 for VAF, 4 for MedAE, which its final rank obtains 28). Accordingly, as can see in Fig 4, second to XGBoost (with a rank of 70), CatBoost (with a rank of 56) gave the best results, followed by LightGBM (with a rank of 42), Gradient Boosting (with a rank of 28) and AdaBoost (with a rank of 14, showing not acceptable results) from the overall ranking point of view: training and testing dataset evaluating combined. It may be observed that the performance based on training is better than on the testing dataset for all models. To better show the performance of the five developed models, the measured and predicted
values of ROP for both the training and testing dataset are depicted in Fig 5 for the GB and XGBoost models and in Fig 6 for the LightGBM, AdaBoost, and CatBoost.

![Models Ranking Without Hyper-parameter](image)

**Fig 4.** Ranking of the developed models based on their performance according to seven evaluation metrics before parameter tuning.
Fig 5. Results of (a) Gradient Boosting and (b) XGBoost models for the training and testing dataset.
Fig 6. Results of (a) LightGBM, (b) AdaBoost, and (c) CatBoost models for the training and testing dataset.
3.2 Discussion

3.2.1 Hyper-parameter Tuning

For finding the possible improvement in the prediction, parameter tuning of the developed models has been carried out associated with five-fold cross-validation incorporated with the GridSearchCV function in “sklearn” of Python. By the way to parameter tuning of the CatBoost model, the “optuna” platform have employed to reach better results. The hyper-parameters of each model, their values showing the best model, the R2 and RMSE of the best models applying to the training and testing datasets, and the time taken to perform the parameter tuning are presented in Table 9. Moreover, the ranking of the parameterized models is shown in Fig 7, in which the GB showed the best model (with a rank of 20), followed by XGBoost (with a rank of 16) and CatBoost (with a rank of 12), LightGBM (with a rank of 8), and the least rank belongs to AdaBoost (with a rank of 4). It may be noticed that the hyper-parameter is not improved the prediction according to the two evaluation metrics of R2 and RMSE in the most developed models. Meanwhile, to be assured of the model accuracy it is recommended to perform the parameter tuning. The overall ranking –obtained according to the results of both before and after the hyper-parameter, plus taking into account the time of hyper-parameter tuning for each model– showed that the XGBoost and CatBoost provide the high-performance models with the rank of 87 and 70, respectively, as shown in Fig 8.

<table>
<thead>
<tr>
<th>Model</th>
<th>Time (sec)</th>
<th>Parameters</th>
<th>Value</th>
<th>R2_best</th>
<th>RMSE_best</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Training</td>
<td>Testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Training</td>
<td>Testing</td>
</tr>
<tr>
<td>GB</td>
<td>158</td>
<td>n_estimators</td>
<td>500</td>
<td>0.9913</td>
<td>0.9889</td>
</tr>
<tr>
<td></td>
<td></td>
<td>learning_rate</td>
<td>0.1</td>
<td>0.0167</td>
<td>0.0149</td>
</tr>
<tr>
<td></td>
<td></td>
<td>max_depth</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XGBoost</td>
<td>566</td>
<td>n_estimators</td>
<td>300</td>
<td>0.9884</td>
<td>0.9849</td>
</tr>
<tr>
<td></td>
<td></td>
<td>learning_rate</td>
<td>0.1</td>
<td>0.0192</td>
<td>0.0174</td>
</tr>
<tr>
<td></td>
<td></td>
<td>max_depth</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reg_alpha</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reg_lambda</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LightGBM</td>
<td>64</td>
<td>n_estimators</td>
<td>200</td>
<td>0.9206</td>
<td>0.9079</td>
</tr>
<tr>
<td></td>
<td></td>
<td>learning_rate</td>
<td>0.1</td>
<td>0.0502</td>
<td>0.0432</td>
</tr>
<tr>
<td></td>
<td></td>
<td>num_leaves</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>max_depth</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>min_child_samples</td>
<td>20</td>
<td>n_estimators</td>
<td>500</td>
<td>learning_rate</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td>----</td>
<td>--------------</td>
<td>-----</td>
<td>---------------</td>
</tr>
<tr>
<td>AB</td>
<td>iterations</td>
<td>1205</td>
<td>learning_rate</td>
<td>0.08844</td>
<td>depth</td>
</tr>
<tr>
<td></td>
<td>colsample_bylevel</td>
<td>0.86831</td>
<td>min_data_in_leaf</td>
<td>65</td>
<td>subsample</td>
</tr>
</tbody>
</table>

Fig 7. Ranking of the developed models based on their performance according to the evaluation metrics after implementing the hyper-parameter.
On the other hand, while CatBoost took much more time for the parameter tuning, it does not need to convert the categorical features by using one-hot encoding; this saves time during the model development, however, for the other algorithms should be used such a converting. This special feature of CatBoost provides an opportunity to find the feature importance without time-consuming back encoding the one-hot encoding of the categorical features. Therefore, to take advantage of such a property, the feature importance obtained from SHAP (SHapley Additive exPlanations) values were obtained and depicted in Fig 9. From this Fig, it is obvious that the feature TF has the highest impact on the ROP prediction, followed by BI and Rock Type. In other words, while the BI, as a rock-related parameter, has been showing a second high impact on the estimation of the ROP, it may be included by using rock type in the prediction models, since it is a coherent characteristic of the rock itself derived from the rock properties.
The “bee swarm” plot illustrated by SHAP values (Fig 9(b)), sorted the variables by their mean absolute SHAP value in descending order with the most important features at the top. Each point in the plot represents a data point plotted against its impact on the predicted value of the ROP; the color of each data represents the relative abundance, ranging from low (blue) to high (red); for instance, the higher TF values (red with positive SHAP values), the higher ROP has been predicted. In other words, positive SHAP values indicate a change in the expected model prediction towards the measured ROP. However, the importance of the Rock Type as a categorical feature has been shown as the fourth important variable impacting the ROP. It should be noted that the interconnectivity of the input parameters has affected the prediction values; so, due to the dependency of the rock mass and intact rock properties on the rock type, it can be concluded that these parameters,
second to the type of rock, can impact on the rate of prediction while their importance is before the rock type. It should be mentioned that the overall feature importance obtained from other algorithms are same as the option CatBoost gives.

To better understand the impact of each parameter on the ROP, dependency plots for four features were plotted as seen in Fig 10. As can be observed in Fig 10(a), the higher magnitudes of the feature TF have a positive effect on the ROP, showing agreement with its physical implication, which the higher the TF values, the higher the rate of penetration. On the other hand, based on its definition, this parameter is the direct interaction of machine and encountered rock, which emphasizes the importance of the Rock Type feature indirectly. To be noticed the influence on the ROP stem from the type of rocks, Fig 10(b) provides a perspective in which the rock type Category1 has a positive impact on the ROP, meaning the excavation in such types of rock would perform at the higher rate of penetration; this category includes two rock groups of (1) Rhyodacite dike, and (2) Quartzite, Sandstone-Slate-Phyllite, Sandstone, Shale, Sandstone-Siltstone, Siliceous tuff, Vitric tuff, Vitric tuff-Sandstone, Vitric tuff. On the contrary, the rocks belonging to Category2, Category3, and Category4 show more and less negative impacts on the ROP, displaying that the excavation in such rocks is more difficult than in the other types leading to less ROP magnitudes. This conclusion can be proven from two points of view: first, the belonging rocks to such categories have high UCS values and second, the category showed high average DPW, meaning the weak planes in the rock mass have more distance and the boring process are more difficult, leading to less ROP. For the other features in the model, the dependency plots reveal their impact on the ROP; as expected, by increasing the rock strength (UCS, Fig 10(c)) the ROP values have decreased. The results obtained from the DPW dependency plot also is in agreement with the physical implications.
Fig 10. SHAP dependency plots for the six features show their impact on the ROP.

3.2.2 Comparison with Literature

For the sake of comparing and verifying the prediction performance of the developed models in this with the publications whose input features are almost the same as this study, a couple of articles have been selected which used AI-based techniques to predict the ROP. To start, it can notice that the R2 for a developed model by Yagiz et al. employing ANN provided an R2 of 0.95 for the four input parameters of UCS, BTS, BI, DPW, and Alpha (Yagiz et al., 2009). It is noticed that the performance shown by the first two best-
developed models in this study (XGBoost and CatBoost, with R2s of 0.9866 and 0.9740, respectively) outperformed the mentioned model.

Another study by Eftekhar et al. (Eftekhar et al., 2010) used the features of UCS, category of rock, percentage of Quartz, BTS, RQD, RMR system, thrust, and torque to estimate the rate of penetration by employing ANN again, showed an R2 of 0.83 while the four of the developed models in this study (XGBoost, CatBoost, GB, and LightGBM) demonstrate better results. While in this research, the of using BI to predict the ROP has been shown at low importance when the UCS and BTS are available, three developed models [XGBoost, CatBoost, and LightGBM] by using the dataset containing BI also outperformed the model developed by Fattahi in which ANFIS–FCM employed to estimate the ROP by UCS, BI, DPW, and Alpha (Fattahi, 2016) whose R2 was 0.831.

In another publication in which Bayesian employed by Adoko et al. to predict the rate of penetration by applying the features of UCS, BI, DPW, and Alpha, an R2 of 0.93 has been obtained (Adoko et al., 2017), again at least two models, of XGBoost and CatBoost, developed here showed higher accuracy in the prediction from R2 point of view. And the last paper in which almost the same input parameters of UCS, BI, DPW, and Alpha were used for the estimation of the ROP presented by Zare Naghadehi et al. (Naghadehi et al., 2018) through GEM, an R2 of 0.7230 calculated which is lower than the R2 of the developed models in this study except for AdaBoost.

An overall view proposes that the performance of the developed models except the AdaBoost model outperformed most of the developed models by the other researchers available in the literature. From another point of view, the models developed here to predict the ROP can be classified in “WhiteBox” machine learning techniques which may be explainable and interpretable compared to the previous (mentioned) models which may be categorized into “BlackBox” ones. Meanwhile, due to the complexities of the TBM
tunneling and TBM performance, every developed model should be used with caution as
well as upgraded by using new and more reliable datasets to be a more general model for
the estimation.

4 Conclusion

By employing a ranking process regarding the evaluation metrics performance indication,
the developed models were ranked, in which the XGBoost showed the highest rank,
followed by CatBoost. To better understand the performance of developed models, the
predicted and measured values of the rate of penetration have been illustrated for the five
models. No such high improvement has been seen through the hyper-parameter tuning.
Meanwhile, in this situation, by using the same ranking of the previous step (considering
only two evaluation indices of R2 and RMSE), GB outperformed the other models. In the
overall ranking, with the consideration of both before and after parameter tuning, the
XGBoost and CatBoost give the highest ranks and 87 and 70, respectively, in which the
time of parameter tuning also has taken into account as an evaluation means in the ranking
process.

Furthermore, the CatBoost model was used to display the features' importance due to its
time-efficient manner in handling the categorical features, since there is no need for extra
operation for the one-hot encoded feature that was used in the other models. By calculating
SHAP values and plotting the bee swarm SHAP summary plot, the most influential feature
has been obtained as the TBM thrust force (TF), which its impact is positive means the
higher its value, the higher the ROP prediction compared to the measured values. This
conclusion agrees not only with the physical meaning of the TF but also with the SHAP
dependency plot of it versus the ROP. On the other hand, the Rock Type feature showed
the third importance for the ROP estimation after UCS at the second rank, and by
employing the SHAP dependency plot, it has been shown that the lower values for the
ROP would be obtained when the tunnel excavation would be implemented in rock types of Category 2, 3 and 4, which is also proven by the rocks high UCS as well as the average value of DPW for such categories which is high enough leading to reduce the rate of penetration. In conclusion, it is concluded that the developed algorithms and performance of the obtained models are very-well and algorithms could be used for similar types of rock and cases in practice.

**Funding** This work was supported by the Faculty Development Competitive Research Grant program of Nazarbayev University in Kazakhstan, Grant Number 021220FD5151.

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

**Data availability** The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

**References**


Oraee, K., Khorami, M. T., & Hosseini, N. (2012). Prediction of the penetration rate of TBM using adaptive neuro fuzzy inference system (ANFIS). Proceeding of SME annual meeting and exhibit,


