



Rock recognition and identification for selective mechanical mining: a self-adaptive artificial neural network approach

Rachel Xu¹ · Ewan J. Sellers¹ · Ebrahim Fathi-Salmi¹

Received: 20 November 2021 / Accepted: 11 June 2023 / Published online: 22 June 2023
© Crown 2023

Abstract

In situ characterisation of rock is crucial for mine planning and design. Recent developments in machine learning (ML) have enabled the whole learning, reasoning, and decision-making process to be more efficient and accurate. Despite these developments, the application of ML in rock-cutting is at an early stage due to the lack of mining applications of mechanised excavation leading to limited availability of data sets and the lack of the expert knowledge required when fine-tuning models. This study presents a novel approach for rock identification during mechanical mining by applying a self-adaptive artificial neural network (ANN) model to classify the rock types for selective cutting, in which datasets from two novel cutting operations (actuated disc cutting (ADC) and oscillating disc cutting (ODC)) were employed to test and train a model. The model was also configured with the Bayesian optimization algorithm to determine optimal hyperparameters in an automated manner. By comparing the performance of each evaluation, the model was trained to identify the best set of hyperparameters at which uncertainty is minimal. Further testing indicated the model is very accurate in classifying rock types for ADC as the accuracy, recall, and precision all equal unity. Some misclassifications occurred for ODC with the accuracy, recall, and precision ranging from 0.68 to 0.99. The promising results proved the model is a robust and scalable tool for classifying the rock types for selective cutting operations enabling the interpretation to be performed more precisely, selectively, and efficiently. Since mechanical cutting requires significant energy, any improvement in matching machine characteristics to the rock mass will increase productivity, and energy efficiency and reduce cost.

Keywords Automated mining · Artificial neural network · Bayesian optimisation · Selective rock cutting · Rock-type prediction

Abbreviations

w_{ij}	Weight between i^{th} neuron to j^{th} neuron
a_j	A net input for j^{th} neuron
x_j	An output for j^{th} neuron
f	Activation function
E_T	The norm of error in all samples
η	The learning rate
p_{nm}	The probability that model assigns to record n as label m

y_{nm}	The indicator if the record n is assigned to class m
N	The number of records
M	The number of class labels
w_{now}	The present set of vectors for the unknown weight parameter
w_{next}	The next set of vectors for the unknown weight parameter
\mathbb{R}	Set of real numbers
$h(x)$	A random variable assigned to x
PI	The improvement probability function
EI	The expected improvement function
UCB	Upper confidence bound function
$h(x^+)$	$h(x^+)$ Is the value of the best sample so far and x^+ is the location of that sample
$\mu(x)$ and $\sigma(x)$	The mean and the standard deviation at x
Φ and ϕ	The CDF and PDF of the standard normal distribution

✉ Rachel Xu
Rachel.Xu@csiro.au
Ewan J. Sellers
Ewan.Sellers@csiro.au
Ebrahim Fathi-Salmi
Ebrahim.Fathisalmi@csiro.au

¹ Hard Rock Mining, Mining3 and CSIRO Mineral Resources, Brisbane, QLD 4069, Australia

δ	The amount of exploration during optimisation for EI
ζ	The tuneable parameter that used to balance exploitation against exploration for UCB
SE	Specific energy
ICR	Instantaneous cutting rate
r	Cutter radius
d	Cutting depth
P	Observation is positive
N	Observation is negative
TP	Observation is positive, and prediction is also positive
TN	Observation is negative, and prediction is also negative
FN	Observation is positive, while prediction is negative
FP	Observation is negative, while prediction is positive

Introduction

As the deluge of the lean strategy continues to impact practically every commercial and scientific domain, the mining industry is also experiencing a transformational shift to extract the deposit at a greater depth (Löw 2015). Considering the variability in ore grade and the cost of extraction, Mining3 (previously known as CRC Mining) recently launched an idea of extracting and processing the targeted ore content in-place (Bryan 2017). This idea is known as in-place mining (IPM), and it can minimise the movement of rock while maximising the amount of recoverable high-value grade. Depending on the ore body and mining method, IPM encompasses three different schemes: (a) in-line recovery (ILR); (b) in situ recovery (ISR); and (c) in-mine recovery (IMR) (Bryan 2017; Mining3 2017; Mousavi and Sellers 2019). In a simple description, ILR incorporates technologies such as selective cutting and ore upgrading to selectively extract and pre-concentrate the material at a location close to the surface; ISR involves pumping the pregnant solution into the ground to dissolve the deeply buried ore grades; IMR is a coupled process where the pre-fragmented rock blocks in the designed stopes are subjected to leaching (Bryan 2017; Mining3 2017; Mousavi and Sellers 2019). Preparing the stopes for IMR needs special treatment via cutting and blasting. In the ILR scheme, the most common method to extract useful minerals is to selectively blast the rock blocks near the contact areas of ore bodies and gangue. This process is oftentimes (a) cyclic; (b) hostile if the ore body is deposited under high in situ stress; and (c) less predictable. Besides, controlling the energy distribution in blasting to avoid overbreak and underbreak could

be somewhat challenging. Problems such as grade loss, ore dilution, and stability are often the results of poor energy control (Konicek et al. 2013). Rather than being blasted, Hood et al. (2005) concluded that if the targeted minerals could be selectively cut from the deposit, mining operations could become more efficient, reliable, and predictable. Yet, the existing cutting technologies (i.e. drag bits and roller discs) suffer from the twin issues of excessive wear rate and high reaction forces when cutting abrasive and strong rocks (Ramezanzadeh and Hood 2010). With the emergence of novel technologies such as undercutting, ODC, and ADC, Vogt (2016) indicated selective cutting shows great potential for future mining.

Despite the novelty, the practicality of these emerging technologies is often constrained by the estimation of in-place geological resources. It is well known that MWD data is an indication of the spatial distribution of rock mass conditions. A better understanding of MWD data not only is beneficial to the blast design but also to the downstream processes such as grade engineering (Sellers et al. 2019). There existed an extensive amount of studies focusing on extracting or interpreting the rock characteristics based on the MWD data. For example, researchers at Mining3 performed numerous laboratory and field studies for rock mass characterisation based on the MWD techniques for blasting design (Segui and Higgins 2002; Smith 2002; Cooper et al. 2011). There were also some analytical approaches using wavelet transformation to solve MWD related rock mass characterisation tasks (Li et al. 2007; Chin 2018).

According to Lewis and Vigh (2017), the reliability and computational cost are the two primary factors for the better developments of inversion technology; adequate data mining techniques such as ML might provide an alternative. Substantial efforts have been devoted to implementing soft computing to model different metrics that affect MWD and rock-cutting operations. Among all, performance prediction and rock type classification were two popular themes. Examples such as the work done by Akin and Karpuz (2008) and Basarir et al. (2014) estimated the rate of penetration (ROP) using ANN and adaptive neuro-fuzzy inference system (ANFIS). Hegde et al. (2020) also employed a data-driven approach such as the random forest (RF) to predict the specific energy (SE) associated with drilling operations with highly accurate prediction. Anifowose et al. (2015) concluded that, compared with traditional regression modeling, the non-parametric nature and the flexibility of the above-mentioned machine learning algorithms make the prediction of drilling operations more efficient and accurate. Other than the performance prediction, studies such as Kadkhodaie-Ilkhchi et al. (2010), Klyuchnikov et al. (2019), Romanenkova et al. (2020), and Zaitouny et al. (2020) used fuzzy system (FS), ANN, decision tree (DT), and recurrent neural network (RNN) algorithms to classify the rock types

associated with MWD data. The successful implementation of soft computing in estimating the lithology from MWD data is of great importance in identifying the productive and non-productive layers to optimise the drilling operations.

For rock cutting, studies were heavily skewed to performance prediction rather than rock type classification. For example, to predict *ROP*, Grima et al. (2000) used ANFIS; Mahdevari et al. (2014), Fattahi and Babanouri (2017) used support vector regression (SVR); Yagiz (2008), Zhang et al. (2020), Zhou et al. (2020), and Koopialipoor et al. (2020) used ANN. For *SE* prediction, Ghasemi et al. (2014) employed a fuzzy system (FS); Salimi et al. (2018) employed the classification and regression tree (CART). Later, Yilmazkaya et al. (2018) concluded that employing soft computing to rock cutting shows better accordance with the actual measurement than traditional regression models. Despite the unparalleled surge in performance prediction, little research has used soft computing techniques to classify the rock type associated with the cutting data. This leaves a gap as information from historical data is crucial for cutting operations to be conducted more precisely, selectively, and efficiently. It is also worth mentioning that the successful implementation of any ML algorithm is often subjected to some hand-tuning to find the best configuration. As emphasised by Bergstra et al. (2013), such parameter tuning is somewhat critical to a method's full potential. Zhang et al. (2020) also pointed out that the influences of hyperparameters were seemingly overlooked when investigating the potential of ML algorithms for rock cutting or MWD.

To shed light upon the knowledge gaps, this paper intended to develop a self-adaptative ANN model to (1) learn from the recently acquired laboratory and field cutting data for ODC and ADC and (2) facilitate the further development of these novel cutting techniques for the selective cutting and breakage of ores for intelligent mining. The developed model was configured with a Bayesian optimisation algorithm to support the automated hyperparameter optimisation. The results indicated that the proposed ANN model is capable of identifying the rock types associated with ODC and ADC with impressive precision. The outcome of this paper could (1) facilitate the further development of these novel cutting technologies and (2) enable mining engineers to effectively and precisely map the high-value ore grade based on a much denser database when planning a selective cutting operation.

Literature review

The nature of this paper necessitated the discussion of the background information about the novel cutting technologies developed over the years, as well as the

fundamental of the artificial neural network and Bayesian optimisation algorithm.

Background information about the selective cutting technologies

Essentially, the foundation of selective cutting lies with the concept of undercutting where a conventional rolling disc is used as a drag bit. By directly creating tensile stress while constantly rotating its interface with the rock, Ramezanzadeh and Hood (2010) stated that the undercutting technology adopts the advantages of drag bits (low reaction force) and roller discs (low wear rate). Through a series of laboratory testing and field trials, Ramezanzadeh and Hood (2010) also pointed out that though the machine weight and power required for the undercutting approach are substantially less than what was observed in conventional TBM, reducing the excavation costs remains an issue. Meanwhile, manufacturers such as Wirth (2013) and Sandvik Mining (2007) have been continuing work in this area to improve the cutting efficiency for mechanical cutting of hard rock mining. This promoted the second generation of undercutting technology, known as oscillating disc cutting.

The ODC technology uses an undercutting disc with an internal drive added to oscillate the disc at a small amplitude. The reason behind adding the complication of oscillation motion to an undercutting disc is that cyclic loading can induce fatigue cracking, therefore weakening the rock. The experimental investigations by Karekal (2013) exhibited force reductions in ODC, while Kovalyshen (2015) predicted such force reductions from an analytical perspective. ODC technology has been licenced to Joy Global by CRCMining since 2006 and later was rebranded as DynaCut (Sundar 2016). Recently, Tadic et al. (2018) conducted a series of field trials using the DynaCut in a sandstone quarry located in Helidon, Queensland, as shown in Fig. 1a. The geotechnical assessment indicated that three types of sandstones exist in the site, including (i) the low strength sandstone (LSS) with UCS ranges from 11 to 35 MPa; (ii) the medium strength sandstone (LSS) with UCS ranges from 29 to 56 MPa; and (iii) the high strength sandstone (HSS) with UCS varies between 47 and 85 MPa. To investigate the potential of processing cutting data for the real/semi-real-time characterisation of rocks, this paper used the ODC data collected from this field trial to train and test the developed ANN model at a field scale.

The concept of actuated disc cutting has been proposed as an extension to both undercutting and oscillating disc cutting technologies. An actuated disc cutter attacks rock in an undercut manner with the disc itself actuated around a secondary axis, as shown in Fig. 1b. Rather than oscillating around a secondary axis with limited frequencies and amplitudes like an ODC, the actuation motion actuates



Fig. 1 Source of the data collected for this study with (a) field trials conducted for ODC using the DynaCut (b) laboratory testings conducted for ADC using Wobble (Tadic et al. 2018; Xu et al. 2021)

the disc cyclically in the direction other than the linear undercutting motion, inducing off-centric revolutions of the disc at a wide range of frequencies and amplitudes. Dehkoda and Detournay (2017) first proposed an analytical model to understand the mechanics of ADC. Further experimental studies by Dehkoda and Detournay (2019) reported the parametric effects of ADC key variables on the average thrust force and specific energy. Rock cutting tests were performed on two types of sedimentary rocks (Savonnière limestone (SL) with UCS ranges from 14 to 19 MPa and Gosford sandstone (GS) with UCS ranges from 28 to 31 MPa), using a CSIRO customised ADC unit, known as Wobble. Wobble is equipped with force, torque, and displacement transducers to monitor the cutting process. Xu (2019) and Xu and Dehkoda (2019) evaluated the ADC-induced fragmentation process by considering the force dynamics and quantifying the fragments generated. To test the idea of rock characterisation based on cutting data, the ADC data collected from ADC laboratory tests was used in this paper to train and test the developed ANN model at a laboratory scale.

Fundamentals of the Artificial neural network and Bayesian optimisation algorithm

The history of ANN can be traced back to 1943 when McCulloch and Pitts (1943) intended to develop a computing system to mimic human biological systems. By replicating the capabilities of the biological neural network, the artificial neurons receive input from synapses and send output when weight is exceeded (Shahin et al. 2002; Shahin et al. 2004). The most frequently used type of ANN is the feed-forward multiplayer perception (MLP). This type of ANN consists of

three basic layers, known as (1) the input layer; (2) the hidden layer; and (3) the output layer. The input layer receives the information from raw data and passes it down to the next layer (i.e. the hidden layer). Once all the hidden layers finish the calculation, the output layer will deliver the result. Though those layers have different functions, the way the information is shared and is determined by the neurons presented in each layer. Each neuron first receives a piece of information and then assigns a random weight to that information. By summarising all the input values multiplied by their corresponding connection weights, the net input for a neuron can be calculated as shown in Eq. (1).

$$a_j = \sum_{i=1}^{\infty} x_i w_{ij} + \theta_j \quad (1)$$

where x_i is the i th neuron in the previous layer, w_{ij} is the weight that connects the i th neuron in the previous layer and j th neuron in the current layer, and θ_j is a bias term that influences the horizontal offset of the function (fixed value of 1).

Once the net put for a neuron is calculated, an activation function will be further applied to determine the output value, referring to Eq. (2).

$$x_j = f(a_j) \quad (2)$$

There are many activation functions available. Based on different purposes, they can be further divided into four categories known as (1) bounded; (2) continuous; (3) monotonic; (4) and continuous. The most commonly used activation function is the rectilinear linear unit function (ReLU). This is because ReLU is rarely saturated by its gradient, which greatly accelerates the convergence of stochastic gradient descent. Other possible activations are the sigmoid

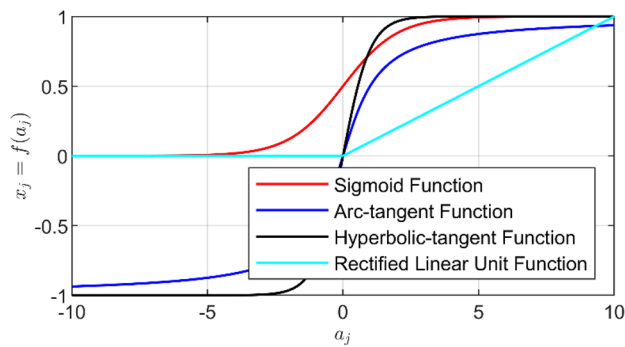


Fig. 2 Different types of activation functions used in ANNs

function, arc-tangent function, and hyperbolic tangent function (Fig. 2).

An ANN model must undergo a training phase to be able to learn the possible relationships between the inputs and outputs. The goal of the ANN training is to adjust the internal weights of the ANN implicitly (Ghaboussi 2018). By optimising the weights, an ANN model seeks to minimise the difference between the neural network outputs and the desired outputs. Different approaches can be used in solving the optimization problem. The classical approach is the back-propagation method where the gradient of the error function is evaluated with respect to its weight and then used to update the weights to improve the response, as shown in Eq. (3).

$$w_{next} = w_{now} - \eta \frac{\partial E_T}{\partial \omega} \tag{3}$$

where η is called the learning rate of the ANN, E_T is the norm of error in all samples, w_{now} is the present set of vector for the unknown weight parameters, and w_{next} is the next set of vector for the unknown weight parameters (Hertz 2018).

The weights are originally initialised randomly. This process proceeds until a solution to Eq. (3) is reached. However, in some cases where the structure of the network is more complex than the nonlinearity between the inputs and the outputs, a problem called overfitting is observed. In this case, the training error is small, and the testing error is large. This often happens when the ANN “memories” the training data but cannot generalise well enough, as shown in Fig. 3. The reasons for overfitting could be (1) the number of hidden neurons used, (2) or the training data is insufficient. Too many hidden neurons give the ANN numerous degrees of freedom in the input/output relationship. Underfitting, on the other hand, happens when both the training and testing errors are large. This occurs when the ANN is poorly trained. Underfitting is corrected by either adding more hidden neurons or adding more training data. Good learning occurs when both the training and

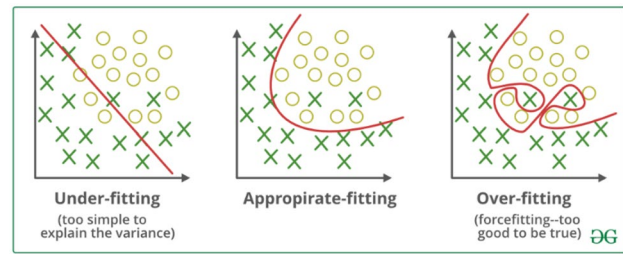


Fig. 3 Overfitting and underfitting of ANN models (Géron 2019)

testing error are small. In this case, the ANN has learned the training data and can generalise for inputs that it has never “seen” before.

Over the year, several attempts have been made to improve the performance of the back-propagation algorithm. Among all, simulated annealing (Sexton et al. 1999; Wang et al. 1999), the genetic algorithm (Goldberg and Holland 1988), and Bayesian optimisation (Rasmussen 2003; Lizotte 2008; Murphy 2012; Snoek et al. 2012) have shown great potential. In particular, simulated annealing is a stochastic global method that searches the optimal based on the likelihood of accepting the current point when compared with other points; the genetic algorithm continually imitates the mechanics of natural selection and natural genetics until no further progress can be made; Bayesian optimisation encodes a prior belief between inputs and outputs, updates the belief based on the laws of probability as information accumulates, and uses the updated belief to guide the optimisation process. Table 1 summarises the pros and cons of the above-mentioned optimisation algorithms. From a practical point of view, Lizotte (2008) and Snoek et al. (2012) concluded that Bayesian has some advantages over the others because (1) it is not restricted to explicitly modeling the objective and (2) it keeps track of each past evaluation which in turn makes the optimisation process more efficient.

To better understand Bayesian optimisation, one has to understand its two major components: (1) a probabilistic model that describes the prior beliefs and (2) an acquisition function that evaluates the next point based on the prior beliefs. For the probabilistic model, most literature chose the Gaussian process (GP) due to its flexibility and tractability. A GP process is a random process where any point $x \in \mathbb{R}$ is assigned a random variable $h(x)$ and where the joint distribution of a finite number of these variables $p[h(x_1), h(x_2), \dots, h(x_N)]$ itself is also Gaussian. For the acquisition function, some of the popular ones are known as the probability of improvement (PI), the expected improvement (EI), and the upper confidence bound (UCB), as shown in Eqs. (4)–(6). It is worth mentioning that this research uses the PI as the acquisition function due to its simplicity.

Table 1 Summary of the popular optimisation algorithms

Algorithm types	Advantages	Disadvantages
Simulated annealing	<ul style="list-style-type: none"> • A rather versatile algorithm as it is independent of the restrictive properties of the model • It is very flexible as a given algorithm can be tuned to solve other problems 	<ul style="list-style-type: none"> • The precision of the input data is crucial to the outcomes • The trade-off between accuracy and computational power
Genetic algorithm	<ul style="list-style-type: none"> • It tends to avoid local minima and find the global (i.e. the search is not restricted to a single part of the problem space) • It is capable of optimising a lot of parallel measures simultaneously (multi-objective) 	<ul style="list-style-type: none"> • Less likely to find a solution when dealing with a large population of many subjects • It is generally computationally expensive
Bayesian optimisation	<ul style="list-style-type: none"> • It tracks the full information provided by the history of the optimisation to make the future search more efficient • It is rather cheap to compute 	<ul style="list-style-type: none"> • The outcomes are very sensitive to the selection of the prior distributions

- The probability of improvement is defined as

$$PI(x) = P(h(x) \geq h(x^+)) = \Phi(Z)$$

$$Z = \frac{\mu(x) - h(x^+)}{\sigma(x)} \quad (4)$$

where $h(x^+)$ is the value of the best sample up to the present and x^+ is the location of that sample and $\mu(x)$ and $\sigma(x)$ are the mean and the standard deviation of the GP posterior predictive at x , respectively.

- Expected improvement is defined as

$$EI(x) = \begin{cases} (\mu(x) - h(x^+) - \delta)\Phi(Z) + \sigma(x)\phi(Z) \\ 0 \end{cases}$$

$$Z = \begin{cases} \frac{\mu(x) - h(x^+) - \delta}{\sigma(x)} \\ 0 \end{cases} \quad (5)$$

where Φ and ϕ are the CDF and PDF of the standard normal distribution, respectively, δ determines the amount of exploration during optimisation and higher δ values lead to more explorations, and a recommended default value for δ is 0.01, and other notations have been defined above.

- Upper confidence bound is also defined as

$$UCB(x) = \mu(x) - \zeta h(x^+) \quad (6)$$

where ζ is a tuneable parameter that is used to balance the exploitation against the exploration of the acquisition function.

Methodology

An overview in Fig. 4 illustrated the workflow of developing the predictive model. A detailed discussion of each step has been further presented in this section.

Data acquisition and definition of input variables

The datasets used in this paper are the testing results from two test campaigns: (1) the fifty cutting tests conducted at a laboratory using ADC and (2) the two hundred and forty-eight cutting trials conducted at the field scale using ODC. As discussed above, the laboratory investigations conducted by Xu (2019) intended to understand the ADC-induced fragmentation under the influences of different operating conditions, while the field trials conducted by Tadic et al. (Tadic et al. 2018) evaluated the scalability of ODC. Despite the objectives, both studies evaluated the performance of ODC and ADC in terms of specific energy and instantaneous cutting rate. As the parametric analyses were not the priority for Tadic et al. (Tadic et al. 2018), all the cutting tests were conducted at the same oscillating amplitude and frequency at various cutting depths with two cutters of different radiuses on three types of sandstone, known as LSS, MSS, and MSS. On the other hand, the cutting tests conducted by Xu (2019) were performed on two types of sedimentary rocks, known as SL and GS, at various actuation amplitudes, frequencies, cutter radiuses, and cutting depths. Considering the designs of experiments for the above two studies were rather different, the input data for the ANN model, therefore, included the following variables for both scenarios for consistency purposes:

- Specific energy (SE) indicates the energy required to cut one unit volume of rock. The cutting process is considered to be more efficient if SE is less. The unit for SE here is MJ/m^3 .
- Instantaneous cutting rate (ICR) is the production rate during the period of cutting. The cutting process is considered to be more efficient if ICR is high. The unit for ICR here is m^3/hr .
- Cutter radius (r) has a unit of mm .
- Cutting depth (d) has a unit of mm .

Fig. 4 The workflow of developing ANN model

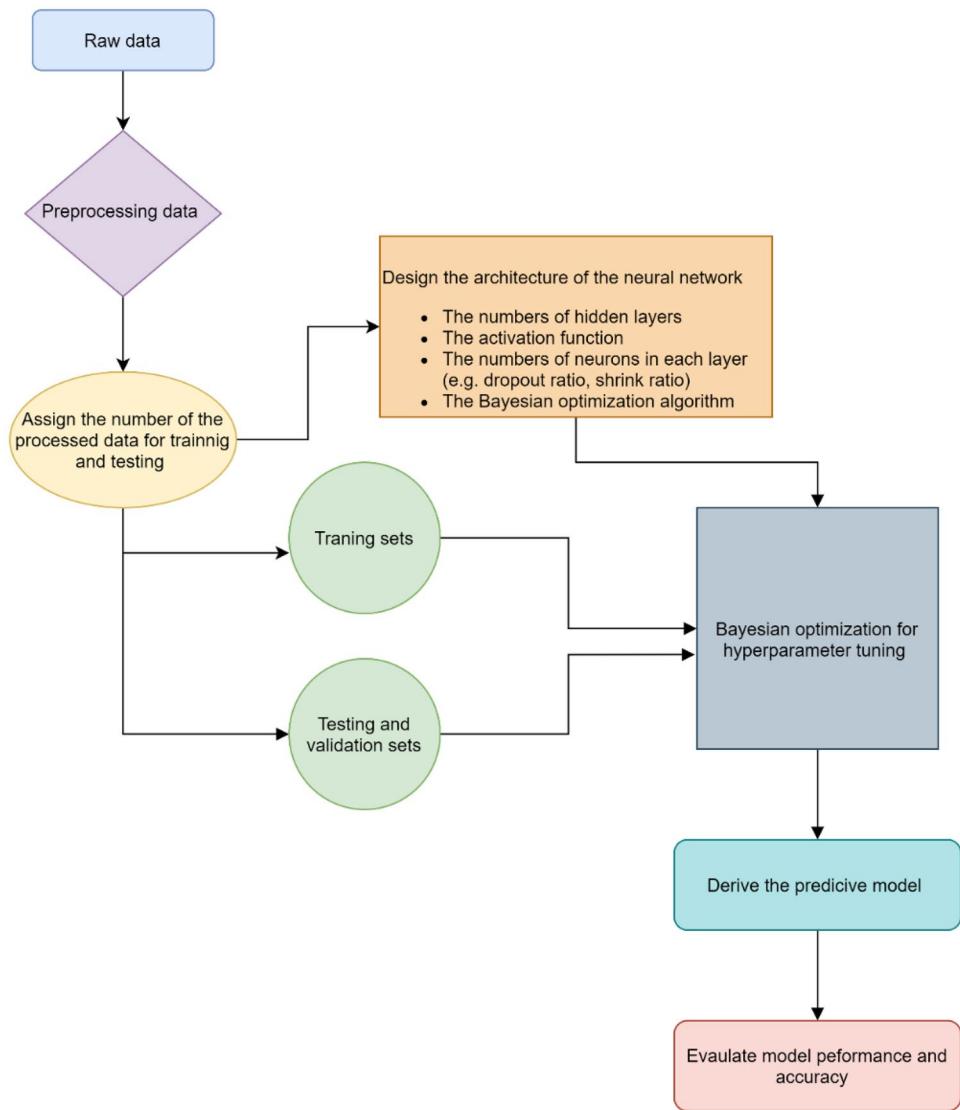


Table 2 provides a summary of the statistical information about the raw datasets. It can be seen that the observed *SE* for the laboratory trials conducted with ADC varies between 0.99 MJ/m^3 to 14.77 MJ/m^3 with a standard

Table 2 Descriptive statistics of the datasets

Cutting methods		<i>SE</i> , MJ/m^3	<i>ICR</i> , m^3/hr	<i>d</i> , mm	<i>r</i> , mm
ADC	Maximum	14.77	0.07	5.02	25.00
	Minimum	0.99	0.01	3.75	5.00
	Mean	5.16	0.02	4.03	15.70
	Standard deviation	3.44	0.02	0.15	7.07
ODC	Maximum	6.88	118.50	141.00	350.00
	Minimum	0.99	34.70	81.00	325.00
	Mean	3.69	78.39	117.88	336.39
	Standard deviation	1.21	18.12	11.25	12.45

deviation around 3.44 MJ/m^3 across the fifty experiments. While for the field trials conducted with ODC, *SE* exhibits a rather narrow spreading from 1.21 to 6.88 MJ/m^3 with a smaller standard deviation of 0.99 MJ/m^3 for the two hundred and forty-eight experiments. The same trend was observed for *r* with ADC showing more variability (i.e. a larger standard deviation when compared with ODC). As for *ICR*, we observed a totally different move as to *SE* and *r*. *ICR* for ODC varies significantly between 34.70 and $118.50 \text{ m}^3/\text{hr}$ with a standard deviation of $18.12 \text{ m}^3/\text{hr}$, while ADC only changes between 0.01 and $0.07 \text{ m}^3/\text{hr}$. Based on the above information, Eq. (7) presents the proposed ANN model where function *g* represents the architecture of the developed model *C* describing the outcome of the model (i.e. the rock types).

$$C = g(SE, ICR, r, d) \tag{7}$$

Pre-processing of raw data and assigning training and testing sets

Pre-processing is often necessary when developing a reliable ML model. As the raw inputs are often comprised of varying scales, converting them into the same scale can (1) reduce the estimation errors and (2) boost the processing time (Sola and Sevilla 1997). To normalise the input variables for ADC and ODC, a Z-score transformation was applied in this paper to avoid outliers. The formula for Z-score normalisation is below in Eq. (8) (Brase and Brase 2013). The normalised datasets for ADC and ODC can be further found in Appendix in Tables 7 and 8. After progressing, the dataset was randomly divided in half, where the training phase was performed on the first 50%, while the remaining 50% was used in the testing phase.

$$z = \frac{X - \mu}{\sigma} \quad (8)$$

where z is the normalised data, X is the raw data, μ is the mean value of feature X , and σ is the standard derivation of feature.

Design the architecture of the ANN model

The ANN model developed in this paper has the following four basic components:

- The number of hidden layers
- The number of neurons in each hidden layer, which includes the dropout percentage and the shrinkage percentage in each layer
- The activation function
- The learning rate

According to Bonilla et al. (2008), different setups of basic components change the architecture of an ANN model and thus alter the synaptic weighting for each input variable. The back-propagation algorithm then predicts the output based on the synaptic weighting of input variables. Despite being clumsy and time-consuming, literature reported that the optimisation of an ANN model is often acquired by using

the trial and error method (Bonilla et al. 2008; Horst and Pardalos 2013; Rajabi et al. 2017).

Steaming from above, the ANN model developed in this paper was coupled with the Bayesian algorithm to enable an efficient and robust optimisation process. Like simulated annealing and genetic algorithms, it is necessary to specify the domain range for each hyperparameter. The Bayesian algorithm often takes the range of each hyperparameter and generates a distribution function in searching for the best one. For the ANN model in this paper, the details of the domain range set for each hyperparameter were presented in Table 3. For the activation function, as discussed before, the ReLu has been employed due to its popularity in solving classification problems. Based on the above details, Fig. 5 provides an example of a typical ANN following the above methodology. It is worth mentioning that the actual model (or the fine-tuned model) is dependent on the output of Bayesian optimisation, which will be further discussed.

Training, testing, and evaluating model performance

The main issues associated with the training and testing of the ANN model are known as overfitting and underfitting. Thus, it is important to monitor the error function, known as the log-loss function for classification problems. To better evaluate the performance of the model, the k -fold cross-validation and confusion matrix were also employed in this paper.

Log-loss function

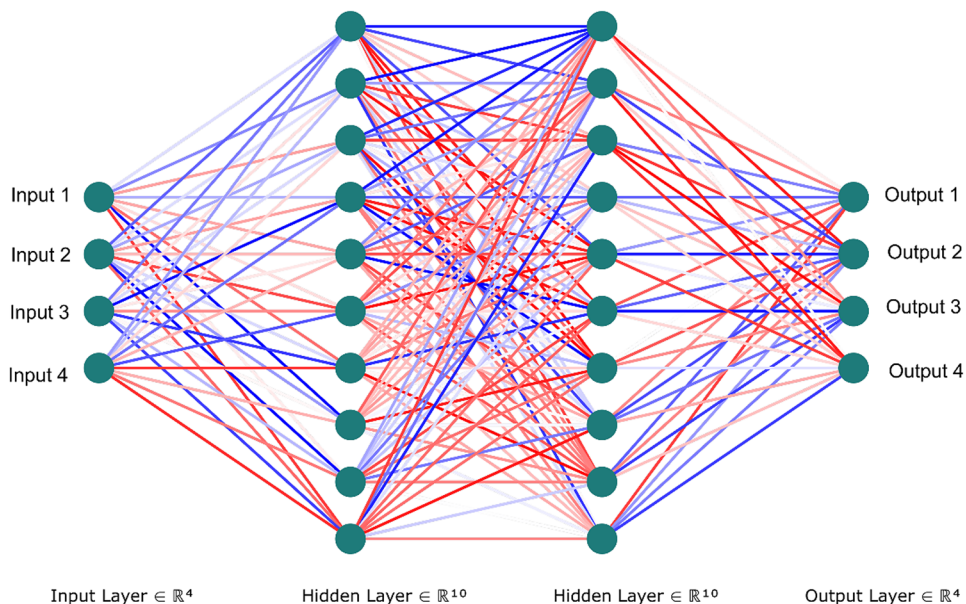
In a classification problem, the log-loss function or cross-entropy is often used to evaluate the performance of an algorithm. Essentially, log-loss compares the probability of the model against the ground truth. If the difference between prediction probability and the ground truth is significant, the model is then penalised for that prediction. Mathematically, the log-loss function is defined in Eq. (9):

$$\log - loss = -\frac{1}{N} \sum_{n=1}^N \sum_{m=1}^M y_{nm} \log p_{nm} \quad (9)$$

Table 3 Domain range of each hyperparameter

Hyperparameter type	Upper bound	Lower bound
The number of hidden layers	1	10
The number of neurons in each hidden layer	1	1000
The learning rate	0	0.1
The allowable dropout percentage in each layer	0.01	1
The allowable shrinkage percentage in each layer	0.01	1

Fig. 5 An example of ANN model developed



where P_{nm} is the probability that model assigns to record n as label m , N is the number of records, M is the number of class labels, and y_{nm} represents the true label m for record.

k-fold cross-validation

When dealing with a rather small training set, cross-validation is often employed to resample the sample to avoid overfitting and underfitting. The common procedures of k -fold cross-validation often involve (1) splitting the training set into k th smaller sets; (2) selecting one set; (3) training the model using the remaining $k-1$ sets; (4) testing the model against the selected one set; and (5) computing the average score for each step. For the ANN model in this paper, the raw data has been divided into half as training and testing sets. The training set has then been spitted into five folds, and the model was then trained on each fold and then validated by the rest of the folds; see Fig. 6.

Confusion matrix

Other than k -fold cross-validation, the confusion matrix also can be used to better visualise the performance of a model. When constructing a confusion matrix, the predictions are often plotted against the true lables, as shown in Table 4, where:

- Positive (P): Observation is positive.
- Negative (N): Observation is negative.
- True positive (TP): Observation is positive and prediction is also positive.
- True negative (TN): Observation is negative and prediction is also negative.

- False negative (FN): Observation is positive while prediction is negative.
- False positive (FP): Observation is negative while prediction is positive.

Once the confusion matrix is ready, several performance parameters known as classification accuracy, recall, and precision will be computed based on the following equations; see Eqs. (10)-(12).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$recall = \frac{TP}{TP + FN} \tag{11}$$

$$precision = \frac{TP}{TP + FP} \tag{12}$$

Results

ANN model training history

For the ANN model proposed in this paper, Bayesian optimisation was further employed in search of the best sets of hyperparameters when training the model. To prevent overfitting and underfitting, the training dataset was split into five folds, where k -fold cross-validation was further conducted. Based on the given domain range (referring to Table 3), Bayesian optimisation computed thirty searches. Each search was first trained on the 4/5th of the training

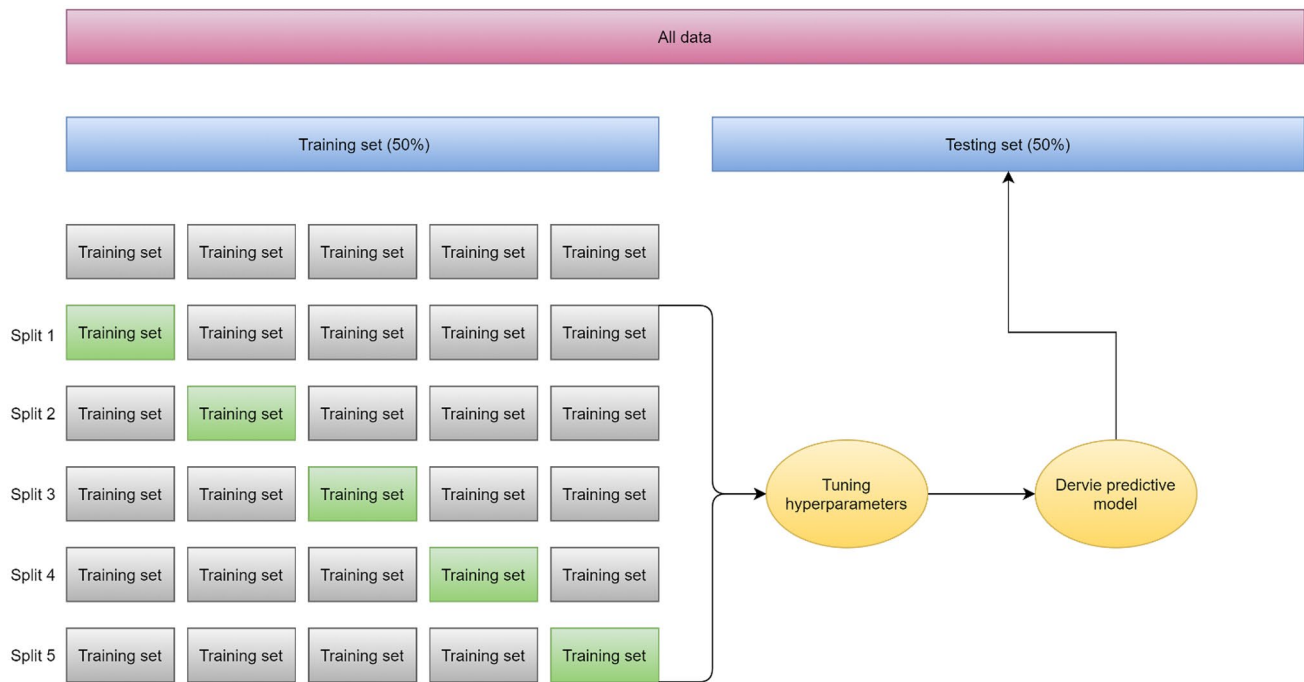


Fig. 6 The procedure of *k*-fold cross-validation for the ANN model

data and later cross-validated on the other 1/5th of the training data. For each search, the log-loss and the epochs (i.e. iterations) were recorded for each training and cross-validation (details of training history can be found in Table 9).

Figure 7 presents the training history for ADC, from which it can be concluded that at the 8th search, the mean log-loss from the five cross-validations is almost zero. This indicated the ANN model finds the best set of hyperparameters at the 8th search as the uncertainty associated with that set of hyperparameters is almost zero. The hyperparameters for the 8th search were attached in Table 5.

For ODC, the training history is shown in Table 9, where it was found that the “classifier” has the highest accuracy (i.e. the lowest uncertainty) at the 22nd search (details of training history can be found in Table 10). As the mean log-loss computed from five cross-validations was lesser than other searches, it can be concluded that the set of hyperparameters associated with the 22nd search is the most compatible set when training the dataset for ODC (Fig. 8). Details of the hyperparameters of the 22nd search are shown in Table 6.

Table 4 A simple guide of the structure of a confusion matrix

Case	Positive predicted	Negative predicted
Positive actual	TP	FN
Negative actual	FP	TN

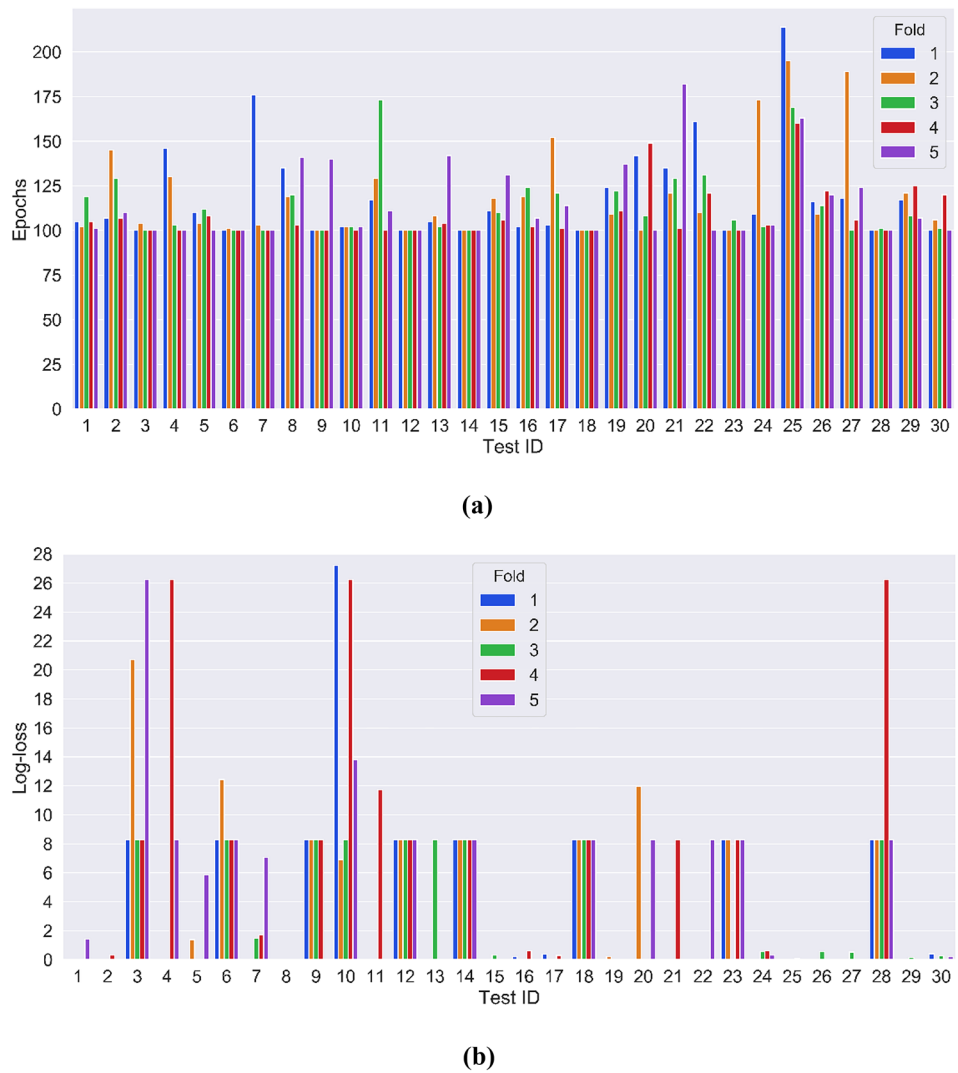
ANN model performance

To evaluate the model’s performance, the confusion matrix was computed. The accuracy, recall, and precision were also calculated using Eqs. (10)–(12). As shown in Fig. 9, from the knowledge gained from the training data, the model could accurately identify the two rock types associated with the testing data in the ADC case. In particular, the proposed ANN model indicated six sets of the testing data are associated with GS, while the other nineteen sets of testing data belong to SL. This prediction matched perfectly with the actual condition, which resulted in a perfect score for recall, precision, and overall accuracy. In contrast, for ODC, the ANN model correctly predicted that thirteen, thirty, and sixty-seven sets of testing data are associated with LSS, MSS, and HSS, respectively. The model, however, misclassified six sets of actual LSS into MSS, five sets of actual MSS into LSS, two sets of actual MSS into HSS, and one set of HSS into MSS (Fig. 10). Though there existed some misclassifications, the proposed ANN model for ODC still scores high in terms of recall, precision, and overall accuracy.

Discussion

ANN could be a powerful tool for the processing of data in mining engineering-related projects. This section of the paper further demonstrated the superiority of the model by comparing the performance of the proposed model with the conventional logistic regression model.

Fig. 7 Training history and hypermeters tuning for ADC where (a) the number epochs (i.e. iterations) was recorded for each training and cross-validation and (b) the log-loss was recorded for each training and cross-validation



Comparison with conventional logistic regression model

Logistic regression (LR) is one of the most fundamental algorithms for solving classification problems. By estimating the relationships between one dependent variable and other independent variables, logistic regression is easy

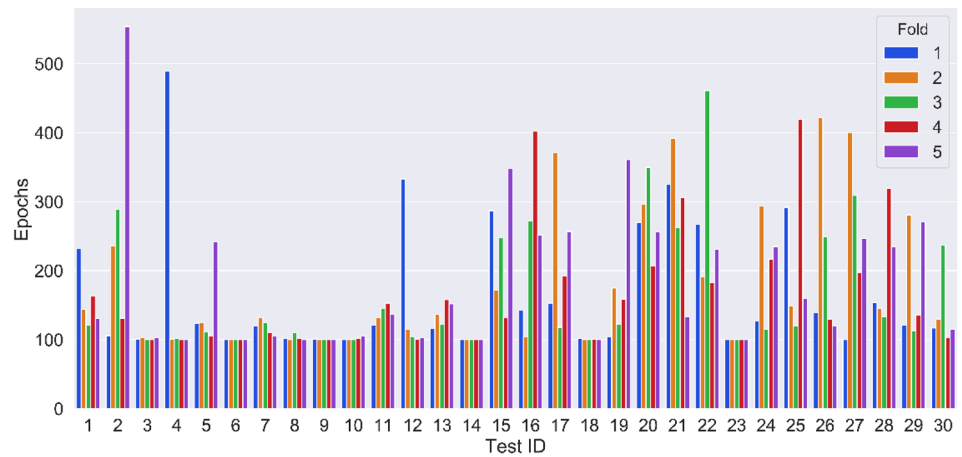
Table 5 The value for each hyperparameter of the 8th search of ANN model for ADC

Hyperparameter type	Value
The number of hidden layers	5.00
The number of neurons percentage in each hidden layer	0.11
The learning rate	0.88
The allowable dropout percentage in each layer	0.43
The allowable shrinkage percentage in each layer	0.08

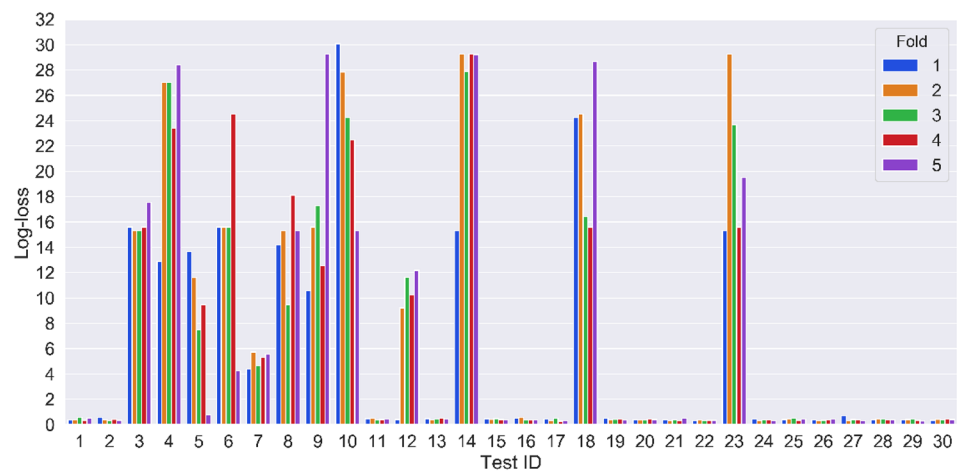
to implement and is often used as the baseline for any binary classification problem. Taking the same datasets as above, this paper further constructed a logistic regression model to predict the rock types associated with ADC and ODC operations. The modeling process is rather straightforward with detailed procedures available from the following literature: Chen et al. (2019), Vallejos and McKinnon (2013), and Subasi (2020). As can be seen from Figs. 11 and 12, the occurrences of misclassifications (both scenarios of ADC and ODC) for the logistic regression model are more frequent than that of the ANN-Bayesian model proposed. This further resulted in lower values of accuracy as.

- The overall accuracy of our ANN-Bay model: ADC=1.00 and ODC=0.89.
- The overall accuracy of logistic regression model: ADC=0.88 and ODC=0.73.

Fig. 8 Training history and hypermeters tuning for ODC where (a) the number epochs where (b) the log-loss was recorded for each training and cross-validation and (b) the log-loss was recorded for each training and cross-validation



(a)



(b)

It can be further concluded that, although LR is easy to implement, its performance is rather poor in comparison with the proposed ANN-Bayesian model. This is due to the fact that LR is not able to handle a large number of categorical features well.

Table 6 The value for each hyperparameter of the 22nd search of ANN model for ODC

Hyperparameter type	Value
The number of hidden layers	5.00
The number of neurons percentage in each hidden layer	0.17
The learning rate	0.08
The allowable dropout percentage in each layer	0.08
The allowable shrinkage percentage in each layer	0.28

Number of Actual	Number of Predicted		False Negative (FN)	Recall
	GS	SL		
GS	6	0	0	1.00
SL	0	19	0	1.00
False Postive (FP)	0	0	Overall accuracy	
Precision	1.00	1.00	1.00	

Fig. 9 The confusion matrix computed for ADC based on the proposed model

Number of Actual		Number of Predicted			False Negative (FN)	Recall
		LSS	MSS	HSS		
Number of Actual	LSS	13	6	0	6	0.68
	MSS	5	30	2	7	0.81
	HSS	0	1	67	1	0.99
False Postive (FP)		5	7	2	Overall accuracy	
Precision		0.72	0.81	0.97	0.89	

Fig. 10 The confusion matrix for computed for ODC based on the proposed model

Applications of the proposed ANN model

Applications of ML techniques in the context of mining and other branches of geoscience and geoen지니어ing are focused on data estimation and forecasting, whereas classical mathematical modelling methods are often constrained by the highly coupled and non-linear relationships between the inputs and the outputs. For the mining industry, knowledge of the locations of the high-value recoverable ore grade is very important to minimise extraction costs. Conventional mathematical modeling might not be applicable here as the field data is oftentimes complicated and huge and sometimes needs real-time processing. A neural network, however, can update its knowledge over time as more information accumulates. The application of this approach, therefore, can result in greater accuracy and more robust prediction than the conventional deterministic or statistical techniques. This paper investigated the applicability of the M, in particular, the self-adaptive ANN model, for the classification of rock

Number of Actual		Number of Predicted		False Negative (FN)	Recall
		GS	SL		
Number of Actual	GS	4	1	1	0.80
	SL	2	18	2	0.90
False Postive (FP)		2	1	Overall accuracy	
Precision		0.67	0.95	0.88	

Fig. 11 The confusion matrix for computed for ADC based on the logistic regression model

types associated with two types of the cutting method at a laboratory and a field scale. The results indicated that the proposed ANN model is robust and accurate in terms of identifying the rock types for actuated disc cutting (ADC) at a field scale. Further upscaling indicated the model is also compatible with the field observation for oscillating disc cutting (ODC). From the above analyses and results, it can be concluded that the proposed ANN model has the potential for:

- Mapping high-value recoverable ore grades more efficiently
- Optimising selective cutting operations
- Facilitating decision-making
- Reducing cutting costs based on the characteristics of rocks

Limitations of the proposed ANN model

Despite delivering promising results during training and testing, the proposed ANN model still has some limitations, most of which are concerned with (1) the stochastic nature of neural networks and (2) the quality of input data. The successful implementation of the neural network is always subjected to some sense of “randomisation”, that is, the stochastic assignment of a subset of the weights to continue the optimisation process. Therefore, it is difficult to control the flow of the model other than checking the output. Further limitations of ANN modeling originate from the data itself. The adage of “what comes in, what comes out” is well applicable to any ANN model. This paper also reflected the importance of input data to the performance of a model. As seen from Figs. 9 and 10, the results that were trained using laboratory data (i.e. ADC data) exhibit a higher level of accuracy when compared to the results obtained for field data (i.e. ODC data). This indicated that field data may suffer from some external viability that is difficult for computational algorithms to interpret, while the laboratory data offers good internal viability. One possible explanation might lie in the scale and homogeneity of rock specimens used for testing. When selecting samples for ADC laboratory tests, there is a chance that the samples were monolithic and monotonic. While when conducting field trials for ODC, the data collected from the machine might be affected by the often jointed, anisotropic, and heterogeneous rock masses. It is also noteworthy that two different cutting mechanics were reported in this paper and further research is required to better understand the impacts of cutting mechanics on the proposed model.

Fig. 12 The confusion matrix for computed for ODC based on the logistic regression model

		Number of Predicted			False Negative (FN)	Recall
		LSS	MSS	HSS		
Number of Actual	LSS	0	19	3	21	0.00
	MSS	0	36	1	1	0.97
	HSS	0	10	55	10	0.84
False Positive (FP)		0	29	4	Overall accuracy	
Precision		0.00	0.55	0.93	0.73	

Conclusion

This paper presented a self-adaptive ANN model for the characterisation of the rock types associated with some historical cutting data to benefit selective cutting for intelligent mining methods. The input data, known as the specific energy, instantaneous cutting rate, cutter radius, and cutting depth, were originated from the laboratory tests conducted for ADC and the field trials conducted for ODC. The results and observations can be summarised as below:

- With the help of Bayesian optimisation, the ANN model presented an architecture of A: 5.00-0.11-0.88-0.43-0.08 for ADC and an architecture of A: 5.00-0.17-0.08-0.08-0.28 for ODC, where each value corresponds to the number of hidden layers, the number of neurons percentage in each hidden layer, the learning rate, the allowable dropout percentage in each layer, and the allowable shrinkage percentage in each layer.
- The results obtained from the model with the above-mentioned architecture were highly encouraging. In particular, the model for ADC was extremely accurate in classifying the rock types with accuracy, recall, and precision all equal to one. For ODC, there existed some misclassifications; this resulted in an overall accuracy of 0.89. The recall and precision observed ranged from 0.68 to 0.99.
- Comparing the results for ADC and ODC, one can conclude that the proposed ANN model was very sensitive to the quality of input data. In specific, the model output for the laboratory tests conducted for ADC was more accurate than the field trials for ODC.
- The proposed ANN model seems quite promising for optimising the selective mining operation. However, like other ML algorithms, this ANN model was also stochastic by nature. This means that the system's flow

is untraceable, and the output from the model might be subjected to some slight variations for each execution.

It is worth mentioning that the Bayesian optimisation algorithm is quite sensitive to the selection of the prior distributions. When constructing the prior distributions, this study took the default mode of settings for simplicity. Therefore, exploring the different settings of the prior distributions and their effects on the accuracy of the model is yet another valuable topic that can be further discussed.

Field implementation of new mechanical mining technologies is often constrained by the estimation of in-place geological resources. This work developed a self-adopting neural network to classify the rock type associated with the measure-while-cutting data. Accurate and precise inversion by experienced geologists could be somewhat time-consuming and labour-intensive so the proposed framework takes advantage of the domain knowledge that is available and developed a neural network to learn from the existing knowledge, further enabling the inversion to be performed for selective cutting operations. The framework was trained and tested not only on the laboratory data but also on the field data. With the field data being fed to the neural network, the model can quickly identify and classify the rock types associated with the cutting.

This methodology provides the next step towards enabling the interpretation to be performed more precisely, selectively, and efficiently in practice with the machines that are now starting to be tested in mining operations. The value of being able to reduce the carbon footprint of mining operations by tuning the energy of the machine to create the appropriate fragmentation for transport and processing is evident, but cannot yet be quantified until further field data is available. The selective mining technologies using ODC have been tested by Hillgrove Mine in South Australia recently. Further study is underway to better understand the performance of the ANN model from a production scale.

Appendix

Table 7 Normalised input data for ADC

Normalised <i>d</i>	Normalised <i>ICR</i>	Normalised <i>SE</i>	Normalised <i>r</i>	Rock Type
6.74	-0.44	-0.70	1.32	SL
-0.07	-0.04	-0.48	1.32	SL
-0.03	1.71	-0.51	1.32	SL
-0.12	-0.21	-0.63	1.32	SL
-0.13	-0.83	-0.39	1.32	SL
-0.13	1.59	-0.46	1.32	SL
0.04	0.02	-0.65	1.32	SL
-0.22	1.59	-0.41	1.32	SL
-0.13	2.16	-0.36	1.32	SL
-0.09	-0.83	-0.44	1.32	SL
-0.16	0.24	-0.42	-0.81	SL
-0.10	0.13	-0.23	-0.81	SL
-0.11	-1.00	-0.48	-0.81	SL
-0.22	0.58	-0.21	-0.81	SL
-0.12	-1.05	-1.20	-0.81	SL
-0.07	-0.44	-1.21	-0.81	SL
-0.09	-0.94	-0.46	-0.81	SL
-0.16	-0.38	-1.21	-0.81	SL
-0.14	0.69	-0.40	-0.81	SL
-0.24	-0.66	-0.28	-0.81	SL
-0.11	-1.00	-0.49	1.32	SL
-0.18	-0.94	-0.21	1.32	SL
-0.23	1.31	-0.37	1.32	SL
-0.13	2.21	-0.47	1.32	SL
-0.03	-1.11	-0.21	-0.81	SL
-0.13	-1.05	-0.34	-0.81	SL
-0.14	-0.15	-0.35	-0.81	SL
-0.11	0.80	-0.33	-0.81	SL
-0.06	0.47	-0.10	-0.81	SL
-0.13	0.13	-1.18	-0.81	SL
-0.16	0.47	-1.19	-0.81	SL
-0.25	0.41	-1.16	-0.81	SL
-0.12	0.30	-0.28	-0.10	SL
-0.12	-1.05	-0.25	-0.10	SL
-0.11	-1.05	-0.21	-0.10	SL
-0.11	1.09	-0.25	-0.10	SL
-0.13	1.14	-0.28	-0.10	SL
-0.11	-0.94	-0.30	-0.10	SL
0.00	-1.05	1.06	-1.51	GS
-0.06	-1.00	0.35	-0.81	GS
0.08	-0.77	1.10	1.32	GS
-0.17	-0.94	1.57	-0.10	GS
-0.09	-0.66	2.24	-1.51	GS
0.00	-0.49	1.96	-0.81	GS
0.08	0.02	0.45	1.32	GS
-0.06	-0.44	2.18	-0.10	GS
-0.15	-0.89	2.80	-1.51	GS
0.36	-0.27	2.03	-0.81	GS
-1.89	2.66	1.43	1.32	GS
-0.19	0.92	1.91	-0.10	GS

Table 8 Normalised input data for ODC

Normalised <i>d</i>	Normalised <i>ICR</i>	Normalised <i>SE</i>	Normalised <i>r</i>	Rock Type
0.72	0.77	-0.77	-0.91	LSS
0.72	-0.05	-0.15	-0.91	LSS
0.72	0.08	-0.23	-0.91	LSS
0.72	0.02	-0.58	-0.91	LSS
0.72	-0.18	0.10	-0.91	LSS
0.72	0.09	-0.10	-0.91	LSS
0.72	0.65	-0.63	-0.91	LSS
0.72	0.03	0.05	-0.91	LSS
0.72	-0.21	0.19	-0.91	LSS
0.72	-0.50	-0.04	-0.91	LSS
0.72	0.09	0.03	-0.91	LSS
0.72	-0.60	0.56	-0.91	LSS
0.72	-0.03	-0.15	-0.91	LSS
0.72	-0.59	0.60	-0.91	LSS
0.46	1.66	-2.17	-0.91	LSS
0.46	1.72	-2.21	-0.91	LSS
0.46	-0.03	-0.09	-0.91	LSS
0.37	1.79	-2.22	-0.91	LSS
0.37	-0.08	0.07	-0.91	LSS
0.37	-0.39	0.43	-0.91	LSS
0.37	0.04	-0.18	-0.91	LSS
0.37	-0.57	0.63	-0.91	LSS
0.37	-0.21	0.18	-0.91	LSS
0.37	0.01	-0.11	-0.91	LSS
0.37	-0.63	0.61	-0.91	LSS
0.37	-0.40	0.26	-0.91	LSS
0.28	-0.47	0.22	-0.91	LSS
0.28	-0.63	0.27	-0.91	LSS
0.28	-0.88	0.92	-0.91	LSS
0.28	1.69	-2.13	-0.91	LSS
0.28	1.69	-2.15	-0.91	LSS
0.54	1.50	-2.07	-0.91	LSS
0.37	-0.69	0.41	-0.91	LSS
0.37	-0.54	0.50	-0.91	LSS
0.37	-0.79	0.60	-0.91	LSS
0.37	-0.73	0.50	-0.91	LSS
0.37	-0.37	0.43	-0.91	LSS
0.37	-0.54	0.37	-0.91	LSS
0.28	-0.40	0.75	-0.91	MSS
0.28	-0.59	0.77	-0.91	MSS
0.28	-0.21	0.28	-0.91	MSS
0.37	-0.81	0.99	-0.91	MSS
0.37	-0.26	0.41	-0.91	MSS
0.28	-0.42	0.49	-0.91	MSS
0.28	-0.36	0.31	-0.91	MSS
-1.41	-0.22	-0.26	-0.91	MSS
0.37	0.14	-0.32	-0.91	MSS
0.37	-0.09	-0.29	-0.91	MSS
0.37	-0.14	0.04	-0.91	MSS
0.72	-0.71	0.54	-0.91	MSS
0.72	-0.99	1.06	-0.91	MSS
0.81	-0.78	0.62	-0.91	MSS
0.72	-0.13	0.35	-0.91	MSS
0.72	-1.34	1.70	-0.91	MSS
0.46	-0.89	1.10	-0.91	MSS
0.37	-1.02	1.40	-0.91	MSS

Table 8 (continued)

0.37	-0.83	1.11	-0.91	MSS
0.37	-0.72	0.74	-0.91	MSS
0.28	-1.21	1.46	-0.91	MSS
0.54	1.56	-2.03	-0.91	MSS
0.37	-0.89	1.11	-0.91	MSS
0.37	-1.06	1.37	-0.91	MSS
0.28	-0.13	0.13	-0.91	MSS
0.28	-1.00	1.22	-0.91	MSS
0.28	-0.82	0.84	-0.91	MSS
0.28	-1.07	1.30	-0.91	MSS
0.37	-0.29	0.50	-0.91	MSS
0.37	-0.47	0.67	-0.91	MSS
0.28	-0.87	1.05	-0.91	MSS
0.37	-0.76	0.91	-0.91	MSS
0.37	-0.68	0.69	-0.91	MSS
0.37	-0.57	0.61	-0.91	MSS
0.28	0.46	-0.32	-0.91	MSS
0.28	0.73	-0.36	-0.91	MSS
0.28	-0.44	0.36	-0.91	MSS
0.37	-0.27	0.22	-0.91	MSS
0.46	-0.08	-0.08	-0.91	MSS
0.46	0.05	-0.10	-0.91	MSS
0.37	0.05	0.11	-0.91	MSS
0.37	0.11	-0.01	-0.91	MSS
0.28	-0.85	0.83	-0.91	MSS
0.28	-0.66	0.71	-0.91	MSS
0.54	-0.88	0.86	-0.91	MSS
0.54	-0.14	0.12	-0.91	MSS
0.37	0.47	-0.23	-0.91	MSS
0.37	0.92	-0.34	-0.91	MSS
0.28	-1.01	1.00	-0.91	MSS
0.28	0.19	0.10	-0.91	MSS
0.37	1.19	-1.17	-0.91	MSS
0.28	0.18	-0.64	-0.91	MSS
0.54	0.73	-1.48	-0.91	MSS
0.46	1.45	-1.94	-0.91	MSS
0.37	1.71	-2.20	-0.91	MSS
0.37	1.55	-1.56	-0.91	MSS
0.28	1.70	-2.15	-0.91	MSS
0.28	1.40	-1.85	-0.91	MSS
0.37	1.84	-2.08	-0.91	MSS
0.37	1.16	-1.23	-0.91	MSS
0.28	1.50	-1.74	-0.91	MSS
0.28	0.21	-0.73	-0.91	MSS
0.28	1.63	-1.66	-0.91	MSS
0.37	0.49	-0.62	-0.91	MSS
0.37	2.21	-1.36	-0.91	MSS
0.37	2.14	-2.15	-0.91	MSS
0.37	1.01	-0.83	-0.91	MSS
0.37	0.01	-0.19	-0.91	MSS
0.28	1.26	-1.12	-0.91	MSS
0.54	1.78	-1.23	-0.91	MSS
0.54	1.61	-1.41	-0.91	MSS
0.28	1.54	-1.23	-0.91	MSS
0.28	1.29	-1.16	-0.91	MSS
0.54	1.23	-0.77	-0.91	HSS

Table 8 (continued)

0.54	1.36	-0.69	-0.91	HSS
0.28	0.70	-0.49	-0.91	HSS
0.28	0.44	-0.35	-0.91	HSS
0.28	0.28	-0.42	-0.91	HSS
0.28	0.50	-0.48	-0.91	HSS
0.46	1.84	-1.60	-0.91	HSS
0.46	1.21	-1.08	-0.91	HSS
0.28	1.26	-1.32	-0.91	HSS
0.28	0.94	-0.93	-0.91	HSS
0.28	0.59	-0.85	-0.91	HSS
0.28	1.18	-0.84	-0.91	HSS
0.46	1.01	-0.76	-0.91	HSS
0.46	1.39	-1.02	-0.91	HSS
0.28	0.64	-0.78	-0.91	HSS
0.28	0.85	-0.68	-0.91	HSS
0.28	0.56	-0.54	-0.91	HSS
0.28	0.87	-0.68	-0.91	HSS
0.37	1.14	-0.84	-0.91	HSS
0.37	1.03	-0.69	-0.91	HSS
0.28	0.30	-0.51	-0.91	HSS
0.28	0.41	-0.32	-0.91	HSS
0.28	0.88	-1.08	-0.91	HSS
0.28	0.06	-0.34	-0.91	HSS
2.05	1.36	-0.78	1.09	HSS
2.05	0.18	-0.16	1.09	HSS
2.05	-0.10	-0.31	1.09	HSS
2.05	-1.06	0.99	1.09	HSS
0.28	0.27	-0.30	1.09	HSS
0.46	0.43	-0.47	1.09	HSS
0.28	0.78	-1.06	1.09	HSS
0.28	0.55	-0.24	1.09	HSS
0.28	0.52	-0.51	1.09	HSS
-2.30	-0.63	0.47	1.09	HSS
-2.30	-0.46	0.42	1.09	HSS
-2.39	-0.48	0.49	1.09	HSS
-2.39	-0.70	0.70	1.09	HSS
-2.30	-0.76	0.94	1.09	HSS
-2.30	-0.70	0.55	1.09	HSS
-2.39	-0.38	0.29	1.09	HSS
-2.30	-0.79	0.88	1.09	HSS
-2.30	-2.21	1.65	1.09	HSS
-2.30	-2.20	1.65	1.09	HSS
-2.39	-2.21	1.44	1.09	HSS
-2.30	-2.31	1.75	1.09	HSS
-2.39	-2.40	1.89	1.09	HSS
-2.39	-2.39	2.21	1.09	HSS
-2.30	-2.32	1.63	1.09	HSS
-2.39	-2.41	2.30	1.09	HSS
0.37	1.77	-1.83	1.09	HSS
0.28	1.55	-1.48	1.09	HSS
0.37	1.66	-1.79	1.09	HSS
0.37	1.78	-1.51	1.09	HSS
0.37	0.85	-1.14	1.09	HSS
0.37	1.78	-1.42	1.09	HSS
0.28	0.65	-0.76	1.09	HSS
0.28	1.05	-1.31	1.09	HSS

Table 8 (continued)

0.28	-0.10	-0.45	1.09	HSS
0.37	0.25	-0.82	1.09	HSS
0.37	0.46	-0.62	1.09	HSS
0.37	0.77	-0.63	1.09	HSS
0.37	-0.21	0.23	1.09	HSS
0.37	0.10	-0.06	1.09	HSS
0.37	0.00	-0.10	1.09	HSS
0.28	-0.18	0.27	1.09	HSS
0.28	-0.52	0.42	1.09	HSS
0.28	-0.34	0.34	1.09	HSS
0.37	-0.08	-0.15	1.09	HSS
0.28	-0.11	0.00	1.09	HSS
0.28	0.22	-0.12	1.09	HSS
0.28	-0.44	0.32	1.09	HSS
-1.41	0.54	-0.21	1.09	HSS
-1.41	0.46	-0.35	1.09	HSS
0.28	1.82	-1.15	1.09	HSS
0.28	-0.12	0.03	1.09	HSS
0.37	0.26	-0.12	1.09	HSS
0.46	1.84	-0.93	1.09	HSS
0.46	0.61	-0.58	1.09	HSS
0.46	1.07	-0.74	1.09	HSS
0.28	0.89	-0.62	1.09	HSS
0.28	-0.04	-0.37	1.09	HSS
0.37	0.12	-0.16	1.09	HSS
0.37	0.29	-0.10	1.09	HSS
0.37	0.27	-0.20	1.09	HSS
0.54	0.52	-0.55	1.09	HSS
0.54	-0.51	0.35	1.09	HSS
0.54	-0.06	0.27	1.09	HSS
0.28	0.30	-0.45	1.09	HSS
0.28	0.09	0.01	1.09	HSS
0.28	-0.05	0.51	1.09	HSS
0.37	0.49	-0.57	1.09	HSS
0.37	-0.46	0.50	1.09	HSS
0.37	-0.46	0.64	1.09	HSS
0.37	1.72	-0.92	1.09	HSS
0.37	-0.16	0.04	1.09	HSS
0.37	0.81	-0.04	1.09	HSS
0.28	0.05	-0.38	1.09	HSS
0.28	-0.19	0.23	1.09	HSS
0.37	0.18	0.38	1.09	HSS
0.37	0.21	-0.07	1.09	HSS
0.28	0.13	-0.01	1.09	HSS
0.28	-0.08	0.63	1.09	HSS
0.37	0.38	-0.49	1.09	HSS
0.37	0.66	-0.53	1.09	HSS
0.37	-0.24	0.38	1.09	HSS
0.28	0.13	-0.06	1.09	HSS
0.28	-0.04	-0.11	1.09	HSS
0.28	-0.42	0.20	1.09	HSS
0.28	1.00	-0.67	1.09	HSS
0.37	-0.81	0.91	1.09	HSS
0.28	-0.89	0.83	1.09	HSS
0.28	-0.79	0.83	1.09	HSS
0.28	-0.71	0.92	1.09	HSS

Table 8 (continued)

0.28	-0.61	0.75	1.09	HSS
-2.39	-0.99	0.83	1.09	HSS
-2.39	-0.97	0.92	1.09	HSS
-2.39	-0.90	0.92	1.09	HSS
-2.39	-1.22	1.41	1.09	HSS
-2.39	-1.26	1.58	1.09	HSS
-2.39	-1.13	1.33	1.09	HSS
-2.39	-1.25	1.35	1.09	HSS
-2.39	-1.27	1.46	1.09	HSS
-3.19	-1.98	2.10	1.09	HSS
-3.28	-1.57	1.88	1.09	HSS
-1.32	-1.85	1.86	1.09	HSS
-1.32	-1.26	1.30	1.09	HSS
-1.50	-2.00	2.63	1.09	HSS
-1.41	-1.81	2.17	1.09	HSS
-1.41	-1.81	2.17	1.09	HSS
-3.28	-1.16	0.96	1.09	HSS
-3.10	-0.73	0.46	1.09	HSS
-1.32	-1.42	1.34	1.09	HSS
-1.32	-1.09	0.93	1.09	HSS
-1.41	-1.67	1.73	1.09	HSS
-1.50	-1.06	0.79	1.09	HSS
0.37	-1.03	0.62	1.09	HSS
0.37	-0.81	0.81	1.09	HSS

Table 9 Training history of ANN model proposed for ADC

Test	Fold	Target	Dropout	lr	neuronPct	neuronShrink	Score	Epochs
1	1	0.28	0.21	0.72	0.01	0.31	0.00026	105
	2						0	102
	3						0.00003	119
	4						0.00029	105
	5						1.41284	101
2	1	0.00	0.07	0.09	0.19	0.35	0.00044	107
	2						0	145
	3						0.00002	129
	4						0.30087	107
	5						0	110
3	1	14.37	0.2	0.54	0.43	0.69	8.28931	100
	2						20.72327	104
	3						8.28931	100
	4						8.28931	100
	5						26.24947	100
4	1	6.91	0.10	0.88	0.04	0.67	0	146
	2						0	130
	3						0	103
	4						26.24947	100
	5						8.28931	100
5	1	1.44	0.21	0.56	0.15	0.21	0	110
	2						1.38155	104
	3						0	112
	4						0	108
	5						5.83787	100
6	1	9.12	0.40	0.97	0.32	0.70	8.28931	100
	2						12.43396	101
	3						8.28931	100
	4						8.28931	100
	5						8.28931	100
7	1	2.05	0.44	0.89	0.09	0.05	0	176
	2						0	103
	3						1.48015	100
	4						1.71862	100
	5						7.05413	100
8	1	0.00	0.08	0.88	0.11	0.43	0	135
	2						0	119
	3						0	120
	4						0	103
	5						0	141
9	1	6.63	0.48	0.53	0.70	0.32	8.28931	100
	2						8.28931	100
	3						8.28931	100
	4						8.28931	100
	5						0	140
10	1	16.50	0.34	0.83	0.03	0.75	27.22141	102
	2						6.90776	102
	3						8.28931	102
	4						26.24947	100
	5						13.81551	102

Table 9 (continued)

Test	Fold	Target	Dropout	lr	neuronPct	neuronShrink	Score	Epochs
11	1	2.35	0.11	0.81	0.09	0.31	0	117
	2						0	129
	3						0	173
	4						11.7326	100
	5						0	111
12	1	8.29	0.22	0.41	0.55	0.75	8.28931	100
	2						8.28931	100
	3						8.28931	100
	4						8.28931	100
	5						8.28931	100
13	1	1.66	0.23	0.66	0.06	0.27	0	105
	2						0	108
	3						8.28931	102
	4						0	104
	5						0	142
14	1	8.29	0.01	0.81	0.97	0.71	8.28931	100
	2						8.28931	100
	3						8.28931	100
	4						8.28931	100
	5						8.28931	100
15	1	0.00	0.08	0.13	0.18	0.33	0.00049	111
	2						0	118
	3						0.30042	110
	4						0.00006	106
	5						0	131
16	1	0.16	0.07	0.08	0.18	0.30	0.19192	102
	2						0	119
	3						0.00061	124
	4						0.59704	102
	5						0.00032	107
17	1	0.13	0.07	0.1	0.13	0.35	0.36669	103
	2						0	152
	3						0.00004	121
	4						0.29903	101
	5						0	114
18	1	8.29	0.16	0.64	0.64	0.75	8.28931	100
	2						8.28931	100
	3						8.28931	100
	4						8.28931	100
	5						8.28931	100
19	1	1.7	0.03	0.12	0.17	0.33	0.00003	124
	2						0.20005	109
	3						0	122
	4						0.00001	111
	5						0	137
20	1	4.05	0.11	0.85	0.05	0.43	0	142
	2						11.93984	100
	3						0	108
	4						0	149
	5						8.28931	100

Table 9 (continued)

Test	Fold	Target	Dropout	lr	neuronPct	neuronShrink	Score	Epochs
21	1	1.66	0.00	0.55	0.17	0.48	0	135
	2						0	121
	3						0	129
	4						8.28931	101
	5						0	182
22	1	1.66	0.07	0.90	0.16	0.42	0	161
	2						0	110
	3						0	131
	4						0	121
	5						8.28931	100
23	1	6.63	0.12	0.45	0.55	0.95	8.28931	100
	2						8.28931	100
	3						0	106
	4						8.28931	100
	5						8.28931	100
24	1	0.30	0.06	0.10	0.16	0.33	0.00039	109
	2						0	173
	3						0.56597	102
	4						0.61852	103
	5						0.30422	103
25	1	0.00	0.13	0.02	0.18	0.65	0.00001	214
	2						0.00003	195
	3						0.0002	169
	4						0.10053	160
	5						0.00004	163
26	1	0.11	0.05	0.12	0.23	0.32	0	116
	2						0	109
	3						0.55704	114
	4						0	122
	5						0	120
27	1	0.11	0.06	0.09	0.18	0.28	0	118
	2						0	189
	3						0.5274	100
	4						0	106
	5						0	124
28	1	11.88	0.07	0.51	0.89	0.57	8.28931	100
	2						8.28931	100
	3						8.28931	101
	4						26.24947	100
	5						8.28931	100
29	1	0.03	0.06	0.14	0.17	0.39	0	117
	2						0	121
	3						0.14002	108
	4						0	125
	5						0	107
30	1	0.17	0.41	0.16	0.02	0.17	0.41058	100
	2						0.00094	106
	3						0.25082	101
	4						0.00073	120
	5						0.18647	100

Table 10 Training history of ANN model proposed for ODC

Test	Fold	Target	Dropout	lr	neuronPct	neuronShrink	Score	Epochs
1	1	0.43	0.21	0.72	0.01	0.3093	0.3436	232
	2						0.367	144
	3						0.5777	121
	4						0.3413	163
	5						0.5172	131
2	1	0.40	0.07	0.09	1944	0.3521	0.5722	106
	2						0.3843	236
	3						0.3043	289
	4						0.4122	131
	5						0.3336	554
3	1	15.88	0.20	0.54	0.43	0.6884	15.5982	101
	2						15.3196	103
	3						15.3196	100
	4						15.5982	100
	5						17.5479	103
4	1	23.75	0.1	0.88	0.04	0.6738	12.8869	489
	2						27.0182	101
	3						27.0182	102
	4						23.3972	100
	5						28.4109	100
5	1	8.60	0.21	0.56	0.15	0.2061	13.6484	124
	2						11.6031	125
	3						7.4941	111
	4						9.475	105
	5						0.7805	242
6	1	0.4	0.97	0.32	0.70	0.40	15.5982	100
	2						15.5982	100
	3						15.5982	100
	4						24.5114	100
	5						4.2329	100
7	1	5.12	0.44	0.89	0.09	0.05	4.3868	120
	2						5.7219	132
	3						4.6483	125
	4						5.3175	110
	5						5.5455	105
8	1	14.48	0.08	0.88	0.11	0.43	14.2055	102
	2						15.3196	100
	3						9.4703	110
	4						18.105	102
	5						15.3196	100
9	1	17.05	0.48	0.53	0.7	0.32	10.5845	101
	2						15.5982	100
	3						17.2694	100
	4						12.5342	100
	5						29.2465	100
10	1	0.34	0.83	0.03	0.7526	0.34	30.0822	100
	2						27.8432	100
	3						24.2329	100
	4						22.4829	102
	5						15.3196	105

Table 10 (continued)

Test	Fold	Target	Dropout	lr	neuronPct	neuronShrink	Score	Epochs
11	1	0.42	0.24	0.73	0.01	0.26	0.4713	121
	2						0.481	132
	3						0.3452	145
	4						0.3943	153
	5						0.4278	137
12	1	8.73	0.20	0.56	0.18	0.22	0.3937	333
	2						9.2165	115
	3						11.6455	104
	4						10.2311	101
	5						12.1782	103
13	1	0.44	0.24	0.71	0.01	0.30	0.4669	116
	2						0.3578	137
	3						0.4249	122
	4						0.5129	158
	5						0.4476	152
14	1	24.18	0.01	0.81	0.97	0.71	15.3196	100
	2						29.2465	100
	3						27.8539	100
	4						29.2465	100
	5						29.2192	100
15	1	0.42	0.23	0.74	0.01	0.29	0.4569	287
	2						0.4054	172
	3						0.4709	248
	4						0.4035	132
	5						0.3641	348
16	1	0.43	0.06	0.06	0.17	0.35	0.5063	143
	2						0.5499	104
	3						0.3461	272
	4						0.3525	402
	5						0.3744	252
17	1	0.36	0.07	0.05	0.21	0.35	0.44	153
	2						0.2971	371
	3						0.4752	118
	4						0.2689	192
	5						0.3084	257
18	1	21.89	0.16	0.64	0.41	0.75	24.2329	102
	2						24.5114	100
	3						16.4338	100
	4						15.5982	101
	5						28.6895	100
19	1	0.40	0.06	0.06	0.2	0.39	0.4752	104
	2						0.3704	175
	3						0.4064	122
	4						0.4193	159
	5						0.3485	361
20	1	0.39	0.11	0.05	0.18	0.36	0.3897	270
	2						0.3489	296
	3						0.3476	350
	4						0.4434	207
	5						0.3957	256

Table 10 (continued)

Test	Fold	Target	Dropout	lr	neuronPct	neuronShrink	Score	Epochs
21	1	0.37	0.05	0.08	0.17	0.33	0.3842	325
	2						0.2957	392
	3						0.3913	262
	4						0.2878	306
	5						0.4931	133
22	1	0.33	0.08	0.08	0.17	0.28	0.3179	267
	2						0.3432	191
	3						0.2997	461
	4						0.3338	183
	5						0.3411	231
23	1	20.67	0.12	0.45	0.55	0.9497	15.3196	100
	2						29.2465	100
	3						23.6758	100
	4						15.5982	100
	5						19.4977	100
24	1	0.36	0.03	0.09	0.19	0.28	0.4524	127
	2						0.3153	294
	3						0.3699	115
	4						0.3639	217
	5						0.3151	235
25	1	0.42	0.11	0.05	0.23	0.41	0.3951	292
	2						0.4471	149
	3						0.5141	120
	4						0.3395	419
	5						0.4236	160
26	1	0.36	0.04	0.1	0.14	0.27	0.3759	139
	2						0.2859	422
	3						0.3373	249
	4						0.3609	130
	5						0.4193	120
27	1	0.41	0.06	0.12	0.16	0.28	0.6884	100
	2						0.288	400
	3						0.4026	309
	4						0.3848	197
	5						0.295	247
28	1	0.40	0.21	0.72	0.01	0.28	0.3868	154
	2						0.448	145
	3						0.4262	133
	4						0.3592	319
	5						0.3934	235
29	1	0.37	0.05	0.06	0.16	0.28	0.3823	121
	2						0.4	281
	3						0.4339	113
	4						0.3401	136
	5						0.2788	271
30	1	0.39	0.41	0.16	0.02	0.17	0.337	117
	2						0.4077	130
	3						0.3698	237
	4						0.4431	103
	5						0.3948	115

Acknowledgements The first author acknowledges the financial support from Mining3 and CSIRO towards the completion of this manuscript. The first author also duly acknowledges Dr. Jeff Heaton for providing the information and resources that facilitate the development of the self-adaptive ANN model, Dr. Sevda Dehkoda for providing the experimental data on ADC, and Dr. Isaac Dzakpata for providing the field data on ODC.

Funding Open access funding provided by CSIRO Library Services.

Code availability Code for “Rock Recognition and Identification for Selective Mechanical Mining: A Self-adaptive Artificial Neural Network Approach” is developed by Rachel Xu. The entire repository uses Python as the backend language. To be able to run the source codes, the installation of libraries such as Tensorflow, Keras, NumPy, Scipy, Sklearn, and Skimage are necessary. There are no specific hardware requirements for the codes. The source codes are available for download at the link: <https://github.com/rachelxu9531/Code-Public-Soft-Computing.git>.

Data availability The data used for train the self-adaptive ANN is listed in Tables 7 and 8. Training history has also been reported in Tables 9 and 10.

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Akin S, Karpuz C (2008) Estimating drilling parameters for diamond bit drilling operations using artificial neural networks. *Int J Geomech* 8:68–73. [https://doi.org/10.1061/\(asce\)1532-3641\(2008\)8:1\(68\)](https://doi.org/10.1061/(asce)1532-3641(2008)8:1(68))
- Anifowose F, Labadin J, Abdulraheem A (2015) Improving the prediction of petroleum reservoir characterization with a stacked generalization ensemble model of support vector machines. *Appl Soft Comput* 26:483–496. <https://doi.org/10.1016/j.asoc.2014.10.017>
- Basarir H, Tutluoglu L, Karpuz C (2014) Penetration rate prediction for diamond bit drilling by adaptive neuro-fuzzy inference system and multiple regressions. *Eng Geol* 173:1–9. <https://doi.org/10.1016/j.enggeo.2014.02.006>
- Bergstra J, Yamins D, Cox D (2013) Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: *Proceedings Of The 30th International Conference On Machine Learning*, Atlanta, Georgia, pp 115–123
- Bonilla Ev, Chai Km, Williams C (2008) Multi-task gaussian process prediction. In: *Proceedings Of The Advances In Neural Information Processing Systems*, Vancouver, Canada, pp 153–160
- Brase CH, Brase CP (2013) *Understanding basic statistics*. Cengage Learning, Boston, Massachusetts
- Bryan I (2017) *The building blocks to in place mining*. Unpublished Research Report Mining3, Brisbane, Australia, pp 1–27
- Chen W, Shahabi H, Shirzadi A, Hong H, Akgun A et al (2019) Novel hybrid artificial intelligence approach of bivariate statistical-methods-based kernel logistic regression classifier for landslide susceptibility modeling. *Bull Eng Geol Environ* 78:4397–4419. <https://doi.org/10.1007/S10064-018-1401-8>
- Chin W (2018) *Measurement while drilling: signal analysis, optimization and design*. John Wiley & Sons, Hoboken, New Jersey
- Cooper C, Doktan M, Dean R (2011) Monitoring while drilling and rock mass recognition. In: *Proceedings Of The Crc Mining Conference*, Brisbane, Queensland, pp 1–12
- Dehkoda S, Detournay E (2017) Mechanics of actuated disc cutting. *Rock Mech Rock Eng* 50:465–483. <https://doi.org/10.1007/s00603-016-1121-y>
- Dehkoda S, Detournay E (2019) Rock cutting experiments with an actuated disc. *Rock Mech Rock Eng* 52:3443–3458. <https://doi.org/10.1007/s00603-019-01767-y>
- Fattahi H, Babanouri N (2017) Applying optimized support vector regression models for prediction of tunnel boring machine performance. *Geotech Geol Eng* 35:2205–2217. <https://doi.org/10.1007/s10706-017-0238-4>
- Géron A (2019) *Hands-on machine learning with scikit-learn, keras, and tensorflow: concepts, tools, and techniques to build intelligent systems*. O'reilly Media, Sebastopol, California
- Ghaboussi J (2018) *Soft Computing In Engineering*. Crc Press, Boca Raton, Florida
- Ghasemi E, Yagiz S, Ataei M (2014) Predicting penetration rate of hard rock tunnel boring machine using fuzzy logic. *Bull Eng Geol Environ* 73:23–35. <https://doi.org/10.1007/s10064-013-0497-0>
- Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3:95–99. <https://doi.org/10.1023/A:1022602019183>
- Grima MA, Bruines PA, Verhoef PNW (2000) Modeling tunnel boring machine performance by neuro-fuzzy methods. *Tunn Undergr Space Technol* 15:259–269. [https://doi.org/10.1016/s0886-7798\(00\)00055-9](https://doi.org/10.1016/s0886-7798(00)00055-9)
- Hegde C, Pircz M, Millwater H, Daigle H, Gray K (2020) Fully coupled end-to-end drilling optimization model using machine learning. *J Pet Sci Eng* 186:66–81. <https://doi.org/10.1016/j.petrol.2019.106681>
- Hertz JA (2018) *Introduction to the theory of neural computation*. Crc Press, Boca Raton, Florida
- Hood M, Guan Z, Tiryaki N, Li X, Karekal S (2005) The benefits of oscillating disc cutting. In: *Proceedings Of The Australian Mining Technology Conference*, Fremantle, Western Australian, pp 267–275
- Horst R, Pardalos PM (2013) *Handbook of global optimization*. Springer, New York City, New York
- Kadkhodaie-Ilkhchi A, St M, Ramos F, Hatherly P (2010) Rock recognition from Mwd data: a comparative study of boosting, neural networks, and fuzzy logic. *Ieee Geosci Remote Sens Lett* 7:680–684. <https://doi.org/10.1109/lgrs.2010.2046312>
- Karekal S (2013) Oscillating disc cutting technique for hard rock excavation. In: *Proceedings Of The 47th U.S. Rock Mechanics/Geomechanics Symposium*, San Francisco, California, pp 2395–2402
- Klyuchnikov N, Zaytsev A, Gruzdev A, Ovchinnikov G, Antipova K et al (2019) Data-driven model for the identification of the rock type at a drilling bit. *J Pet Sci Eng* 178:506–516. <https://doi.org/10.1016/j.petrol.2019.03.041>
- Konicek P, Soucek K, Stas L, Singh R (2013) Long-hole destress blasting for rockburst control during deep underground coal mining.

- Int J Rock Mech Min Sci 61:141–153. <https://doi.org/10.1016/j.ijrmms.2013.02.001>
- Koopialipour M, Fahimifar A, En G, Momenzadeh M, Dj A (2020) Development of a new hybrid ann for solving a geotechnical problem related to tunnel boring machine performance. *Eng Comput* 36:345–357. <https://doi.org/10.1007/s00366-019-00701-8>
- Kovalyshen Y (2015) Analytical model of oscillatory disc cutting. *Int J Rock Mech Min Sci* 77:378–383. <https://doi.org/10.1016/j.ijrmms.2015.04.015>
- Lewis W, Vigh D (2017) Deep learning prior models from seismic images for full-waveform inversion. In: *Proceedings Of The Seg Technical Program Expanded Abstracts Houston, Texas*, pp 1512–1517
- Li CW, Mu DJ, Li AZ, Liao QM, Qu JH (2007) Drilling mud signal processing based on wavelet. In: *Proceedings Of The 2007 International Conference On Wavelet Analysis And Pattern Recognition*, pp 1545–1549
- Lizotte Dj (2008) Practical Bayesian optimization. Ph.D. Dissertation, University Of Alberta
- Lööv J (2015) Lean production in mining - an overview. Ph.D. Dissertation, Lulea University Of Technology
- Mahdevari S, Shahriar K, Yazig S, Akbarpour Shirazi M (2014) A support vector regression model for predicting tunnel boring machine penetration rates. *Int J Rock Mech Min Sci* 72:214–229. <https://doi.org/10.1016/j.ijrmms.2014.09.012>
- McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115–133. <https://doi.org/10.1007/bf02478259>
- Mining3 (2017) In place mining – a transformational shift in metal extraction. <https://www.mining3.com/Place-Mining-Transformational-Shift-Metal-Extraction/#:~:Text=Mining3%20is%20leading%20a%20transformational,A%20low%20capital%20intensity%20mine.> Accessed 01 Sep 2020
- Mousavi A, Sellers E (2019) Optimisation of production planning for an innovative hybrid underground mining method. *Resour Pol* 62:184–192. <https://doi.org/10.1016/j.resourpol.2019.03.002>
- Murphy KP (2012) *Machine learning: a probabilistic perspective*. Mit Press, Cambridge, Massachusetts
- Rajabi M, Rahmannedj R, Rezaei M, Ganjalipour K (2017) Evaluation of the maximum horizontal displacement around the power station caverns using artificial neural network. *Tunn Undergr Space Technol* 64:51–60. <https://doi.org/10.1016/j.tust.2017.01.010>
- Ramezanzadeh A, Hood M (2010) A state-of-the-art review of mechanical rock excavation technologies. *J Min Environ* 1:29–39. <https://doi.org/10.22044/jme.2010.4>
- Rasmussen CE (2003) *Gaussian processes in machine learning*. Springer, Berlin, Heidelberg
- Romanenkova E, Zaytsev A, Klyuchnikov N, Gruzdev A, Antipova K et al (2020) Real-time data-driven detection of the rock-type alteration during a directional drilling. *Ieee Geosci Remote Sens Lett* 17:1861–1865. <https://doi.org/10.1109/lgrs.2019.2959845>
- Salimi A, Faradonbeh RS, Monjezi M, Moormann C (2018) Tbm performance estimation using a classification and regression tree (Cart) technique. *Bull Eng Geol Environ* 77:429–440. <https://doi.org/10.1007/s10064-016-0969-0>
- Sandvik Mining (2007) Sandvik Dp1100. Accessed 30 Aug 2019
- Segui JB, Higgins M (2002) Blast design using measurement while drilling parameters. *Fragblast* 6:287–299. <https://doi.org/10.1076/frag.6.3.287.14052>
- Sellers Ej, Salmi Ef, Usami K, Greyvensteyn I, Mousavi A (2019) Detailed rock mass characterization - a prerequisite for successful differential blast design. In: *Proceedings Of The 5th Iserm Young Scholars' Symposium On Rock Mechanics And International Symposium On Rock Engineering For Innovative Future*, Okinawa, Japan, pp 1–6
- Sexton RS, Dorsey RE, Johnson JD (1999) Beyond backpropagation: using simulated annealing for training neural networks. *J Organ End User Comput* 11:3–10
- Shahin MA, Maier HR, Jaksa MB (2002) Predicting settlement of shallow foundations using neural networks. *J Geotech Geoenviron Eng* 128:785–793. [https://doi.org/10.1061/\(asce\)1090-0241\(2002\)128:9\(785\)](https://doi.org/10.1061/(asce)1090-0241(2002)128:9(785))
- Shahin MA, Maier HR, Jaksa MB (2004) Data division for developing neural networks applied to geotechnical engineering. *J Comput Civ Eng* 18:105–114. [https://doi.org/10.1061/\(asce\)0887-3801\(2004\)18:2\(105\)](https://doi.org/10.1061/(asce)0887-3801(2004)18:2(105))
- Smith B (2002) Improvements in blast fragmentation using measurement while drilling parameters. *Fragblast* 6:301–310. <https://doi.org/10.1076/frag.6.3.301.14055>
- Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In: *Proceedings Of The Advances In Neural Information Processing Systems*, Long Beach, California, pp 2951–2959
- Sola J, Sevilla J (1997) Importance of input data normalization for the application of neural networks to complex industrial problems. *Ieee Trans Nucl Sci* 44:1464–1468. <https://doi.org/10.1109/23.589532>
- Subasi A (2020) *Practical machine learning for data analysis using python*. Academic Press
- Sundar L (2016) Dynacut oscillating disc cutter technology achieving breakthroughs. <https://www.australianmining.com.au/dynacut-oscillating-disc-cutter-technology-achieving-breakthroughs/>. Accessed 10 Sep 2020
- Tadic D, Quidim J, Dzakpata I (2018) Dynacut fundamental development and scalability testing-phase 1 final report. Unpublished Research Report Australian Coal Association Research Program, Brisbane, Australia, pp 1–80
- Vallejos JA, Mckinnon SD (2013) Logistic regression and neural network classification of seismic records. *Int J Rock Mech Min Sci* 62:86–95. <https://doi.org/10.1016/j.ijrmms.2013.04.005>
- Vogt D (2016) A review of rock cutting for underground mining: past, present, and future. *J S Afr Inst Min Metall* 116:1011–1026. http://www.scielo.org.za/scielo.php?script=sci_abstract&pid=S2225-62532016001100006
- Wang F, Vk D, Xi C, Zhang Q-J (1999) Neural network structures and training algorithms for Rf and microwave applications. *Int J Rf Micro Comput-Aided Eng* 9:216–240
- Wirth (2013) Wirth mobile tunnel miner. http://www.infomine.com/suppliers/PublicDoc/AkerWirth/Mobile_Tunnel_Miner_en.pdf. Accessed 30 Aug 2019
- Xu R (2019) Experimental study of rock fragmentation with an actuated undercutting disc. Mphil. Dissertation, University Of New South Wales
- Xu R, Dehkhoda S (2019) Effect of actuation on rock fragmentation in undercutting discs. In: *Proceedings Of The 53rd U.S. Rock Mechanics/Geomechanics Symposium*, New York City, New York, pp 1–8
- Xu R, Dehkhoda S, Hagan PC, Oh J (2021) Evaluation of cutting fragments in relation to force dynamics in actuated disc cutting. *Int J Rock Mech Min Sci* 146:104850. <https://doi.org/10.1016/j.ijrmms.2021.104850>
- Yazig S (2008) Utilizing rock mass properties for predicting tbm performance in hard rock condition. *Tunn Undergr Space Technol* 23:326–339. <https://doi.org/10.1016/j.tust.2007.04.011>
- Yilmazkaya E, Dagdelenler G, Ozcelik Y, Sonmez H (2018) Prediction of mono-wire cutting machine performance parameters using artificial neural network and regression models. *Eng Geol* 239:96–108. <https://doi.org/10.1016/j.enggeo.2018.03.009>
- Zaitouny A, Small M, Hill J, Emelyanov I, Mb C (2020) Fast automatic detection of geological boundaries from multivariate log data using recurrence. *Comput Geosci* 135:104362. <https://doi.org/10.1016/j.cageo.2019.104362>

Zhang Q, Hu W, Liu Z, Tan J (2020) Tbm performance prediction with Bayesian optimization and automated machine learning. *Tunn Undergr Space Technol* 103:493–507. <https://doi.org/10.1016/j.tust.2020.103493>

Zhou J, Yazdani Bejarbaneh B, Jahed Armaghani D, Mm T (2020) Forecasting of Tbm advance rate in hard rock condition based

on artificial neural network and genetic programming techniques. *Bull Eng Geol Environ* 79:2069–2084. <https://doi.org/10.1007/s10064-019-01626-8>