Journal of Rock Mechanics and Geotechnical Engineering 14 (2022) 123–143

Contents lists available at ScienceDirect

Journal of Rock Mechanics and Geotechnical Engineering

journal homepage: www.jrmge.cn

Full Length Article

Real-time prediction of rock mass classification based on TBM operation big data and stacking technique of ensemble learning

Shaokang Hou, Yaoru Liu*, Qiang Yang

State Key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing, 100084, China

ARTICLE INFO

Article history:
Received 8 January 2021
Received in revised form
27 March 2021
Accepted 9 May 2021
Available online 7 July 2021

Keywords:
Tunnel boring machine (TBM) operation data
Rock mass classification
Stacking ensemble learning
Sample imbalance
Synthetic minority oversampling technique (SMOTE)

ABSTRACT

Real-time prediction of the rock mass class in front of the tunnel face is essential for the adaptive adjustment of tunnel boring machines (TBMs). During the TBM tunnelling process, a large number of operation data are generated, reflecting the interaction between the TBM system and surrounding rock, and these data can be used to evaluate the rock mass quality. This study proposed a stacking ensemble classifier for the real-time prediction of the rock mass classification using TBM operation data. Based on the Songhua River water conveyance project, a total of 7538 TB M tunnelling cycles and the corresponding rock mass classes are obtained after data preprocessing. Then, through the tree-based feature selection method, 10 key TBM operation parameters are selected, and the mean values of the 10 selected features in the stable phase after removing outliers are calculated as the inputs of classifiers. The preprocessed data are randomly divided into the training set (90%) and test set (10%) using simple random sampling. Besides stacking ensemble classifier, seven individual classifiers are established as the comparison. These classifiers include support vector machine (SVM), k-nearest neighbors (KNN), random forest (RF), gradient boosting decision tree (GBDT), decision tree (DT), logistic regression (LR) and multilayer perceptron (MLP), where the hyper-parameters of each classifier are optimised using the grid search method. The prediction results show that the stacking ensemble classifier has a better performance than individual classifiers, and it shows a more powerful learning and generalisation ability for small and imbalanced samples. Additionally, a relative balance training set is obtained by the synthetic minority oversampling technique (SMOTE), and the influence of sample imbalance on the prediction performance is discussed.

1. Introduction

Tunnel boring machines (TBMs) are widely used in the construction of underground engineering. Compared with the drill and blast method, TBMs have the advantages of fast construction speed and minor environmental disturbance, which is suitable for constructing long-distance tunnels (Zheng et al., 2016; Liu et al., 2020a). However, TBMs are sensitive to geological conditions, and the uncertainty of rock mass and adverse geological conditions are the main risks in TBM excavation (Hamidi et al., 2010; Hasanpour et al., 2017; Zhou et al., 2021a). Therefore, evaluation of rock mass quality is of great significance to the safety and efficiency of tunnel construction. On the one hand, at the design stage, TBM type and support form selection are determined according to the rock mass classification obtained from geological prospecting. On the other hand, in the construction process, the parameters of TBM are adjusted adaptively according to the rock mass classes (Gong et al., 2016). Before the tunnel construction, there are many geological prospecting means, which can roughly describe the geological and surrounding rock conditions of the construction site (Li et al., 2017). However, due to the fact that the space between the cutterhead and tunnel face is narrow, it is challenging to acquire the surrounding rock parameters through traditional exploration and in situ testing methods (Liu et al., 2020b). Consequently, the limited rock parameters are not sufficient for the adjustment and optimisation of TBM operation parameters. Therefore, it is crucial to put forward a method that can accurately and real-time predict the rock mass classification in front of the tunnel face.

For rock mass classification, different scholars have proposed many representative theoretical methods. For example, Bieniawski

* Corresponding author.
E-mail address: liuyaoru@tsinghua.edu.cn (Y. Liu).
Peer review under responsibility of Institute of Rock and Soil Mechanics, Chinese Academy of Sciences.

https://doi.org/10.1016/j.jrmge.2021.05.004

(1973) proposed the rock mass rating (RMR) system in 1973 after investigating more than 300 tunnels. RMR scores the rock mass quality, mainly considering the uniaxial compressive strength (UCS) of rock mass, rock quality designation (RQD), joint spacing, joint condition (JC), groundwater state and correction coefficient to determine the total score of rock mass, and divides the rock mass quality into five grades. The Q system for rock mass quality assessment proposed by Barton et al. (1974) is also an earlier method of rock mass classification, which considers the integrity of rock mass, groundwater condition, in situ stress, joint characteristics, and uses six parameters to determine the rock mass quality index reflecting the stability of surrounding rock. Furthermore, by considering the influence of structure and discontinuity surface conditions on the mechanical properties of rock mass based on the Hoek-Brown criterion, Hoek (1994) proposed the geological strength index (GSI) to realise the classification of rock mass quality. In 2002, Barton (2002) revised the Q system and explained the corresponding relationship between the new Q system and the RMR system. Besides the above methods, in China, the mainly used methods include the basic quality (BQ) method and the hydropower classification (HC) method (GB50487-2008, 2008; GB/T50218–2014, 2014). Different rock mass classification methods have been widely used in tunnel, mining and other underground engineering. However, the traditional theoretical rock mass classification methods are usually used at the preconstruction stage, and the related indices are difficult to be obtained during the tunnel construction process (Huang et al., 2013). Additionally, for most rock mass classification methods, the mapping relationship between the indices and rock mass classes is unclear, and the randomness of index distribution is hardly considered (Zheng et al., 2020).

In addition to theoretical classification methods, many researchers have introduced artificial intelligence methods for evaluating rock mass quality in recent years. These methods also explore the relationship between the factors that may affect the performance and operational parameters of TBM (Zhou et al., 2021a; Chen et al., 2021), minimizing the subjectivity and inaccuracy of artificial evaluation. Gholami et al. (2013) used the index parameters of the RMR system as the inputs of machine learning models to predict the RMR for the tunnel surrounding rock, and it showed that the machine learning models have more reliable prediction results than the use of empirical correlations. Salimi et al. (2017) established the correlation between the field penetration index (FPI) and rock mass quality parameters, e.g. UCS, RQD and JC using the regression tree model. Santos et al. (2021) used factor analysis to extract three common factors from the indices of the RMR system, and based on this, an artificial neural network (ANN) classifier was established to predict the rock mass classification. Zheng et al. (2020) established a classifier based on a least-squares support vector machine (LSSVM) optimised by a bacterial foraging optimisation algorithm (BFOA). Also, they used geological prediction and rock strength resilience results as the inputs of the classifier to predict the rock mass classes. Zhao et al. (2019) proposed a data-driven framework to predict the geological type thickness of an urban subway based on the values of seven physical-mechanical indices. Jalalifar et al. (2014) established the two rock mass classification models based on the fuzzy inference system and the multi-variable regression analysis to predict the accurate RMR, and the fuzzy model showed better prediction accuracy than the regression model.

Currently, the traditional classification methods have been widely used, and the research of machine learning models based on the parameters of traditional classification methods or parameters of the geological forecast beforehand also achieved good progress (Alimoradi et al., 2008; Shi et al., 2014). However, the parameters of the theoretical rock mass classification methods need to be obtained through field and laboratory tests, which cannot be easily collected in real time during the TBM tunnelling (Huang et al., 2013). Therefore, it is unable to achieve real-time and fast prediction of rock mass classes through the above measuring parameters. In the actual engineering practice, there are some geological forecast beforehand that can predict the rock mass conditions in front of the tunnel face. However, the geological forecast beforehand needs additional time and equipment, which will increase the cost of the project. Furthermore, TBM is a large equipment and occupies most of the space near the tunnel face, thus it is challenging to install the equipment of the geological forecast beforehand (Li et al., 2020). TBM can be seen as a large-scale rock testing machine, and the tunnelling and rock breaking process of TBM is essentially a process of rock-TBM interaction (Yang et al., 2016). Therefore, in the TBM tunnelling process, the change of machine operation parameters results from the interaction between the TBM system and surrounding rocks (Zhang et al., 2019). Many studies have shown that the TBM operation parameters can be used to reflect the rock mass conditions (Yagiz, 2006; Hassanpour et al., 2011; Salimi et al., 2018; Liu et al., 2020c). Additionally, during TBM tunnelling, a large volume of mechanical information of the TBM can be automatically collected by various sensors (Jung et al., 2019). Therefore, it is feasible to predict the rock mass classification in front of the tunnel face in real time based on the TBM operation parameters as the inputs of machine learning models.

In most of the existing researches, different individual classifiers are often used to predict rock mass classification. However, the number of valuable data in engineering fields is relatively small, and the data proportion of different rock mass classes is usually quite different in practice. Therefore, rock mass class prediction belongs to the problem of small and imbalanced samples. For this kind of problem, the individual classifiers are easy to cause overfitting for the majority class samples, and the prediction performance is often poor for minority class samples (Ganganwar, 2012; Sainin et al., 2017). Ensemble learning is a powerful technique that integrates multiple individual classifiers to form a robust classifier. Many studies show that ensemble learning models have a strong generalisation ability and better performance on imbalanced datasets (Salunkhe and Mali, 2016; Feng et al., 2020). In addition to adopting the ensemble learning strategies, another way to overcome the sample imbalance problem is using oversampling algorithm or undersampling to change the sample proportion of different classes (Brun et al., 2018). Undersampling is a method to improve the sample imbalance by removing some majority class samples (Fan et al., 2017), while the method of oversampling is to generate some minority class samples to improve the sample imbalance (Viloria et al., 2020). When the total number of samples is small, the oversampling method may seem to be preferred, and the commonly used methods are the synthetic minority oversampling technique (SMOTE) (Chawla et al., 2002) and its improved algorithms (Panda, 2017).

In this study, the stacking technique of ensemble learning is introduced. By taking support vector machine (SVM), k-nearest neighbors (KNN), random forest (RF) and gradient boosting decision tree (GBDT) as the base classifiers and the GBDT as the meta-classifier, a stacking ensemble classifier is proposed for real-time prediction of rock mass classification during TBM tunnelling process. A database is established based on the Songhua River diversion tunnel project in China, including 802-d TBM operation data and corresponding rock mass classification information. Through the data preprocessing and feature selection, a total of 7538 sample sets are obtained, and 10 crucial features are selected as the input features of classifiers. Besides stacking ensemble classifiers, seven individual classifiers (i.e. SVM, KNN, RF, GBDT, decision tree (DT),

multi-layers perception (MLP) and logistic regression (LR)) are established, and the hyper-parameters of each classifier are optimised by grid search method. Then, based on the randomly divided training set (90%) and test set (10%), the prediction performance of different classifiers is evaluated, and the advantages of stacking ensemble classifier over individual classifiers are analysed. Additionally, the influence of sample imbalance on the prediction effect is discussed.

2. Method description

2.1. Stacking ensemble learning

The ensemble learning classifier is relative to the individual classifier. By integrating multiple homogeneous or heterogeneous 'weak' classifiers, the generalisation ability and robustness of an individual learner are effectively improved (Sun et al., 2020). Many studies have shown that the ensemble learning classifier has better prediction performance than a single classifier, and has been widely used in different problem scenarios (Díez-Pastor et al., 2015; Sun and Trevor, 2018). Based on different integration strategies, ensemble learning can be divided into three algorithms: bagging, boosting and stacking (Polikar, 2012). Bagging usually considers homogeneous weak learners trained independently and combined based on a specific deterministic averaging process (Breiman, 1996). Boosting also considers homogeneous weak learners. It trains these weak learners sequentially in a highly adaptive way, and combines them based on specific deterministic strategies (Friedman, 2001). Unlike Bagging and Boosting, Stacking considers heterogeneous weak learners, and it combines multiple classification models via a meta-learner (Wolpert, 1992; Kardani et al., 2020). Fig. 1 shows the principle of the stacking ensemble classification model. The stacking ensemble learning framework comprises two classifiers, including base classifiers (level-I) and meta-classifier (level-II). Firstly, the original dataset is used to train the multiple base classifiers. In the training process, in order to reduce the risk of over-fitting, the *k*-fold cross-validation (CV) method (Kohavi, 1995) is generally used to train each base classifier. Then, the output of the base classifiers constitutes a new dataset, and the meta-classifier is fitted based on the new dataset to obtain the final prediction results. The specific steps of the stacking algorithm are as follows:

(1) The original dataset is randomly divided into original training set D and original test set T.
(2) Each base classifier is trained based on k-fold CV method. The original training set D is randomly divided into k equal parts (D_1, D_2, …, D_k). Take turns to use one part of them as the test set and the remaining $k-1$ parts as the training set. The k is set as 5 in this study, which means repeating the above process 5 times. The combination of the prediction results of the base classifiers is taken as the new training set D^* of the meta-classifier.
(3) Each base classifier is used to predict the original test T, and the predicted results are averaged as the new test set T^* of the meta-classifier.
(4) Using the new training set D^* and new test set T^* to train and test the meta-classifier, and the meta-classifier outputs the final prediction results.

2.2. Introduction to the base classifiers

For stacking ensemble learning, selecting the appropriate base classifiers and meta-classifier is the key to ensure the prediction effect. In order to compare the prediction effect and generalisation ability of the stacking model, we select several commonly used classification models, including SVM, DT, KNN, RF and GBDT. Due to the advantages of mature theory and high efficiency, KNN and SVM are widely used and have good application effect in many fields (Liao and Vemuri, 2002; Durgesh and Lekha, 2010). RF and GBDT are tree-based algorithms based on bagging and boosting, respectively. RF can be trained in parallel, which significantly improves computational efficiency. Moreover, the outputs of the RF model are determined by majority voting of all DTs (Breiman, 2001). In comparison, the DTs of GBDT are generated serially. The output of GBDT is to add up the prediction results of all DTs or add them up weighted (Friedman, 2001). From the perspective of bias and variance, RF mainly reduces error variance, and GBDT can reduce both bias and variance. Thus, a good combination of the two algorithms can ensure the effectiveness of the results. Therefore,

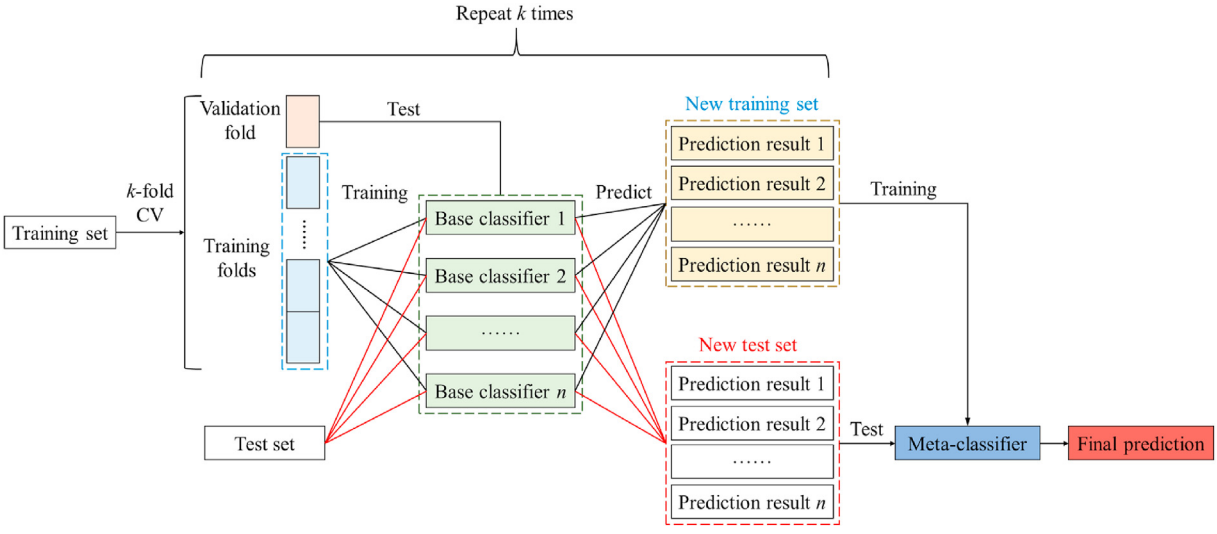

Fig. 1. Workflow of stacking ensemble learning combing k-fold cross-validation.

SVM, KNN, RF and GBDT are used as the base classifiers in this study, and the GBDT is used as a meta-classifier to correct the bias of multiple classification algorithms to the training set.

2.2.1. Support vector machine (SVM)

SVM is a kind of machine learning method based on statistics theory, and it is often used to deal with classification problems (Vapnik, 2000). For linear binary classification, assuming that the training set is $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ $(i = 1, 2, \ldots, n, y \in \{-1, 1\})$, the basic idea of SVM is to construct a separating hyperplane $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0$, so that the samples of two different classes are separated, where b is the function bias of separating hyperplane. The support vector is the sample points close to the separating hyperplane. The optimal separate hyperplane maximizes the distance between the support vector of two different classes of samples and the separate hyperplane (Srivastava and Bhambhu, 2010). The problem of solving the optimal hyperplane is a constrained optimisation problem. By using the duality of Lagrange multipliers, it is transformed into the following optimisation problem:

$$\left.\begin{aligned} &\max_{\alpha}\left[\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha_i\alpha_j\boldsymbol{y}_i\boldsymbol{y}_j\right)\right] \\ &\text{s.t.}\ \sum_{i=1}^{n}\left(\alpha_i\boldsymbol{y}_j\right) = 0\ (0 \le \alpha_i \le C, i = 1, 2, \ldots, n) \end{aligned}\right\} \tag{1}$$

where α_i and α_j are the Lagrange coefficients, and C is the penalty coefficient.

The final optimal classification function is as follows:

$$f(x) = \operatorname{sgn}\left[\sum_{i=1}^{n}\left(\alpha_i^*\boldsymbol{y}_i\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right) + b^*\right] \tag{2}$$

where α_i^* is the optimal Lagrange coefficient, and b^* is the optimal value of b.

In linear indivisibility, SVM introduces a kernel function to map the data samples from low dimensional space to high dimensional space by transforming the inner product function, making the high-dimensional space linearly separable (Liu and Hou, 2019). In this study, the radial basis function (RBF) is used as the kernel function. For the problem of multi-classification, SVM achieves the classification goal by combining several two classifiers.

2.2.2. K-nearest neighbor (KNN)

KNN is a classic and straightforward machine learning classification algorithm (Altman, 1992). Assume that the $\boldsymbol{x}_q^{\mathrm{t}}$ is the vector to be classified. The basic principle of KNN is to firstly find the k vectors which are most similar to $\boldsymbol{x}_q^{\mathrm{t}}$ in the sample space. Then count the most frequent class of these k vectors, and $\boldsymbol{x}_q^{\mathrm{t}}$ is determined as this class. The similarity of two vectors is usually measured by their Euclidean distance, which can be calculated as follows:

$$D_{\mathrm{E}} = \sqrt{\sum_{q=1}^{d}\left(\boldsymbol{x}_q^{\mathrm{s}} - \boldsymbol{x}_q^{\mathrm{t}}\right)^2} \tag{3}$$

where D_{E} is the Euclidian distance, $\boldsymbol{x}_q^{\mathrm{s}}$ is the sample vector, and d is the dimension of the samples.

The KNN algorithm mainly depends on the limited adjacent samples rather than identifying the class field. Therefore, the KNN algorithm is more suitable for the sample set with more overlapping class fields (Imandoust and Bolandraftar, 2013).

2.2.3. Random forest (RF)

RF algorithm is a powerful supervised ensemble learning algorithm proposed by Breiman (2001). RF can be regarded as an improved bagging method, and it is developed based on DT theory (Zhou et al., 2017). The idea of RF is to use the bootstrap resampling method to extract multiple samples from the original samples, and construct a DT for each bootstrap sample. In a RF, each DT is randomly generated, and different DTs are independent of each other. For a classification problem, the final classification results are determined based on the majority vote of all DTs.

In the RF algorithm, generation of DTs involving node split algorithms, including ID3, C4.5 and CART (Myles et al., 2004). In this study, the CART algorithm is used to construct the RF. CART uses the Gini index to measure the importance of feature attributes to realize node split. Suppose that sample set D contains T classes and n features (X_1, X_2, …, X_n), then the Gini index is as follows:

$$\mathrm{Gini}(D) = 1 - \sum_{t=1}^{T} p_j^2 = 1 - \sum_{t=1}^{T}\left(\frac{|C_t|}{|D|}\right)^2 \tag{4}$$

where C_t is the subset of samples belonging to class t in sample set D. After a split of node k, the sample set D is divided into m parts (D_1, D_2, …, D_m) based on feature X_j ($j = 1, 2, \ldots, n$). The Gini index GI_k can be expressed as follows:

$$GI_k = \mathrm{Gini}(D, A) = \frac{|D_1|}{|D|}\mathrm{Gini}(D_1) + \cdots + \frac{|D_m|}{|D|}\mathrm{Gini}(D_m) \tag{5}$$

2.2.4. Gradient boosting decision tree (GBDT)

GBDT is an iterative DT-based algorithm based on boosting strategy (Friedman, 2001, 2002). With its strong generalisation ability, GBDT is widely used to solve classification and regression problems. The CART-based DT is usually used for constructing GBDT, and the DTs are iteratively constructed (Wang et al., 2016). In each iteration, a new DT is generated, and the residuals of the previous DT are used to train the current DT. Also, in each iteration, the gradient descent method is used to increase the learning weight on the incorrectly predicted samples, so that the error of the model to the objective function is smaller than that in the previous iteration (Kuhn and Johnson, 2013). The convergence condition of GBDT is that the model satisfies the preset classification error or reaches the upper limit of the number of DTs. Finally, these trained DT classifiers are integrated into a robust classifier.

2.3. Synthetic minority oversampling technique

For the imbalanced sample set, there are usually two data processing methods: oversampling and undersampling. The main idea of oversampling and undersampling is to increase the number of minority class samples and remove part of the majority class samples, respectively. Through oversampling or undersampling, the number of different classes can become relative balance (Elrahman and Abraham, 2013). However, unlike the professional fields such as natural language processing, which can easily obtain valuable massive data, there are relatively little valuable data for many problems in underground engineering fields. In this study, the number of valid TBM tunnelling data is relatively small. Therefore, it is not appropriate to remove the majority class samples by undersampling. Additionally, the sample number difference between different classes is relatively significant. Therefore, we use the SMOTE algorithm (Chawla et al., 2002) to process the original imbalanced training set.

SMOTE algorithm is a kind of oversampling technique for synthesising minority samples and can be regarded as an improved strategy of the random oversampling algorithm. Because random oversampling adopts the strategy of simply copying samples to increase the number of minority class samples, it is easy to produce the problem of over-fitting and reduce the generalisation ability of the classifier. To overcome this, the basic principle of SMOTE algorithm is to analyse the minority samples and synthesise new samples based on the minority samples to add to the dataset. Fig. 2 shows the schematic diagram of SMOTE oversampling. The specific steps of SMOTE oversampling are as follows:

(1) Suppose a minority class sample in the feature space (such as the blue ball in Fig. 2). For the minority class sample x_i, the Euclidean distances between x_i and all other minority class samples are calculated to obtain k nearest neighbors. Generally, k is taken as 5.
(2) Through the analysis of imbalanced samples, the sampling rate N is determined. For each minority class sample x_i, several samples are randomly selected from its k nearest neighbors, assuming that one of the selected nearest neighbor samples is $\tilde{x}_i$.
(3) For the nearest neighbor sample $\tilde{x}_i$ and the minority class sample x_i, a new sample x_{new} is synthesised at a random point on their connecting line. The calculation formula is as follows:

$$x_{new} = x_i + \text{rand}(0,\ 1)|x_i - \tilde{x}_i| \tag{6}$$

where $\text{rand}(0,\ 1)$ represents a random number between 0 and 1.

2.4. Evaluation metrics of the model

In order to evaluate the prediction effect of the classifiers, different evaluation metrics have been put forward or used in evaluating the performance of machine learning models (Luque et al., 2019; Zhou et al., 2019). For the imbalance of samples, the study selects six evaluation metrics: accuracy (*ACC*), precision (*PRC*), recall (*REC*), F_1-score (F_1), Cohen's kappa coefficient (*Kappa*) and area under the receiver operating characteristic (ROC) curve (*AUC*) to evaluate the prediction performance and select the best classifier for rock mass classification.

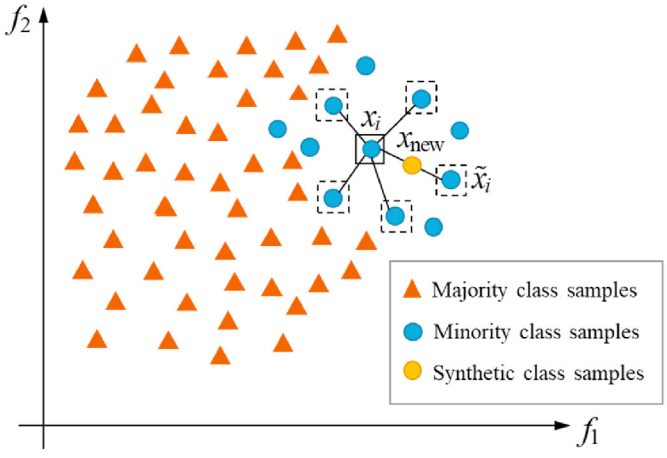

Fig. 2. Schematic diagram of SMOTE oversampling.

		Predicted class: II	Predicted class: III / IV / V
Actual class	II	True positive (*TP*): (Class II is correctly predicted as class II)	False negative (*FN*): (Class II is incorrectly predicted as Class III/IV/V)
	III / IV / V	False positive (*FP*): (Class III/IV/V is incorrectly predicted as class II)	True negative (*TN*): (Class III/IV/V is correctly predicted as class III/IV/V)

Fig. 3. Schematic diagram of the binary confusion matrix (taking the prediction of class II as the example).

ACC represents the proportion of correctly predicted samples to the total predicted samples, and the *ACC* metric is most widely used to evaluate the prediction performance of a classifier. However, for imbalanced classification tasks, *ACC* is incapable of reflecting the performance of classifiers. *REC* represents the proportion of correctly predicted samples of a certain class to all predicted samples of that class. *PRC* represents the proportion of correctly predicted samples of a certain class to the predicted samples of this class. It can be seen that there is a certain contradiction between *PRC* and *REC*, which reflects the discrimination ability of the model to positive samples and negative samples, respectively. F_1 is the composite metric of *REC* and *PRC*, which eliminates the one-sidedness of these two indices to a certain extent. These metrics can be calculated based on a confusion matrix, and the calculation formulae are as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$REC = \frac{TP}{TP + FN} \tag{8}$$

$$PRC = \frac{TP}{TP + FP} \tag{9}$$

$$F_1 = 2\frac{REC \cdot PRC}{REC + PRC} \tag{10}$$

where *TP* is the true positive, which represents the number of samples that are actually of positive class and correctly predicted as the positive class by the classifier; *FN* is the false negative, representing the number of samples that are actually of positive class but incorrectly predicted as the negative class; *FP* is the false positive, representing the number of samples that are actually of negative class but incorrectly predicted as the positive class; and *TN* is the true negative, representing the number of samples that are actually negative class and correctly predicted as negative class. Prediction of rock mass classification is a four-classification problem, and it can be regarded as four binary classification problems. To better understand the four symbols of *TP*, *FN*, *FP* and *TN*, the schematic diagram of the binary confusion matrix (taking the prediction of class II as the example) is shown in Fig. 3.

The above evaluation metrics (i.e. *ACC*, *REC*, *PRC* and F_1) are suitable for solving binary classification problems. The rock mass classification can be regarded as the combination of four binary classification problems. Therefore, the four evaluation metrics can be used to evaluate the prediction effect for each class. For the total prediction effect of each classifier, the $ACC_{\_Total}$ can also be calculated as the proportion of correctly predicted samples to total samples:

$$ACC_{\_Total} = \frac{n_{correct}}{n_{total}} \tag{11}$$

where $n_{correct}$ is the number of samples that are correctly classified, and n_{total} is the total number of samples.

While the other three metrics (i.e. $REC_{\_Total}$, $PRC_{\_Total}$ and F_{1_Total}) can be calculated by the weighted macro-average across classes as follows:

$$REC_{\_Total} = \sum_{i=1}^{T}(REC_i w_i) \tag{12}$$

$$PRC_{\_Total} = \sum_{i=1}^{T}(PRC_i w_i) \tag{13}$$

$$F_{1\_Total} = \sum_{i=1}^{T}(F_{1i} w_i) \tag{14}$$

where REC_i is the recall of class i, PRC_i is the precision of class i, F_{1i} is the F_1-score of class i, w_i is the proportion of the samples of class i to the total samples.

Cohen's kappa coefficient (*Kappa*) is a robust metric that measures the proportion of correctly classified units after the probability of change agreement has been removed (Cohen, 1960), which takes into account the probability that a pixel is classified by chance (Dong et al., 2013; Zhou et al., 2015, 2016). Compared with *ACC* metric, *Kappa* metric considers the sample imbalance to a certain extent. The *Kappa* coefficient can be calculated as

$$Kappa = \frac{P_0 - P_e}{1 - P_e} \tag{15}$$

where P_0 is the sum of the number of correctly classified samples in each class divided by the total number of samples, i.e. the overall classification accuracy rate, $ACC_{\_Total}$; and P_e is the expected proportion of samples correctly classified by chance. Assuming that the number of the real samples in each class is $a_1, a_2, \ldots, a_u$, the number of the predicted samples of each class is $b_1, b_2, \ldots, b_u$, the total number of the classes is u, and the total number of samples is n, P_e can be calculated as

$$P_e = \frac{a_1 b_1 + a_2 b_2 + \cdots + a_u b_u}{n^2} \tag{16}$$

Table 1
Relative strength of agreement corresponding to *Kappa* value (Landis and Koch, 1977).

Kappa	Strength of agreement
−1–0	Poor
0–0.2	Slight
0.21–0.4	Fair
0.41–0.6	Moderate
0.61–0.8	Substantial
0.81–1	Almost perfect

Table 1 shows the relative strength of agreement corresponding to the *Kappa* statistic (Landis and Koch, 1977). *Kappa* < 0.4 is an indication of poor agreement, while *Kappa* ≥ 0.4 is an indication of reasonable agreement.

The *AUC* from the ROC curve is also a metric that can be used to evaluate the prediction accuracy of classifiers (Bradley, 1997). The ROC curve plots the true positive rate (*TPR*, i.e. recall) against the false positive rate ($FPR = FP/(TN + FN)$). The values of *AUC* vary from 0.5 to 1, indicating the discrimination accuracy, which can be divided into five degrees (Bradley, 1997; Zhou et al., 2019): not discrimination (0.5–0.6), poor discrimination (0.6–0.7), fair discrimination (0.7–0.8), good discrimination (0.8–0.9), and excellent discrimination (0.9–1). The ROC curve and the *AUC* value are usually used for evaluation of binary classifiers. For the multiple classifiers, the micro-average ROC curve and macro-average *AUC* values are used as the evaluation metrics for the prediction performance. The micro-average ROC curve and its corresponding *AUC* value are obtained by stacking the results of all groups together, thus converting the multi-class classification into binary classification. The macro-average ROC curve and its corresponding *AUC* value are obtained by averaging all groups' results (one vs. rest), and linear interpolation was used between points of ROC curve (Wei et al., 2018). Compared with the micro-average *AUC*, the macro-average *AUC* is more influenced by the minority class samples.

3. Database acquisition and preprocessing

3.1. Database acquisition

In this study, taking the No. 4 bid section of the Songhua River water conveyance project in China as the research object, the TBM operation database is established. Fig. 4 shows the location of the study area of the Songhua River water conveyance project. The construction section is located between the Chalu River and Yinma River, and the total length of the tunnel is 22,955 m. During the construction process, the length excavated by TBM is about 20,198 m, accounting for about 88%, and the rest section is constructed using drill and blast method. The mileage of the study area is from K71 + 855 to K48 + 900, the elevation range is from 264 m to 484 m, and the buried depth is from 85 m to 260 m. The design shape of the diversion tunnel section is circular. The open TBM with an excavation diameter of 8.03 m is used to excavate the tunnel. The main technical parameters of open TBM are listed in Table 2.

In the whole construction process of the TBM section, the operation data of TBM are collected once a second. From July 2015 to February 2018, a total of 802 d of TBM operation data were recorded. About 86,400 pieces of TBM operation data were collected every day, and 4.08 billion pieces of data were finally obtained to form the database. The actual performance of TBM equipment in different strata and operating conditions is recorded entirely. In the database, each piece of data contains 191 TB M machine parameters, time stamp information and the corresponding mileage. Fig. 5 shows the TBM systems and distribution of the acquisition parameters. Fig. 6 shows the variation of four key TBM operation parameters in a day. TBM takes a tunnelling cycle as a working unit, which can be defined as a process from TBM start-up to shut-down. In a whole TBM tunnelling cycle, the operation parameters increase from zero to a stable value for continuous excavation and then decrease to zero. During this period, TBM advances a certain distance forward, and the footage of each tunnelling cycle is about 1.8 m. It can be seen from Fig. 6 that there are 29 TB M tunnelling cycles on February 1, 2016.

In addition, according to the construction mileage, the lithology and rock mass classification along the tunnel are also recorded, as

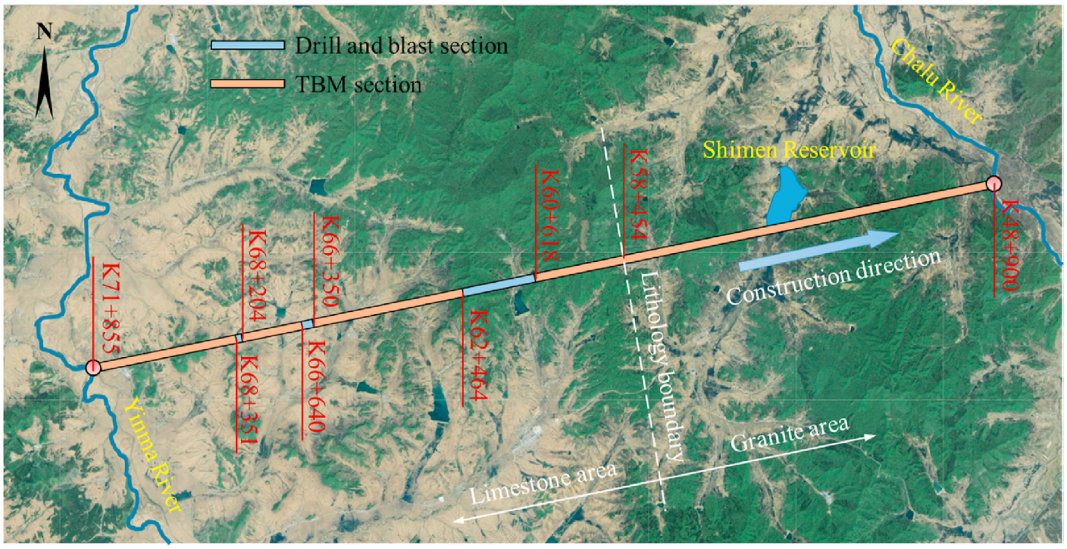

Fig. 4. Location of the study area of the Songhua River water conveyance project.

shown in Fig. 7. The construction site mainly includes two types of lithology, i.e. granite and limestone, accounting for 41.62% and 58.38%, respectively. The mileage of the lithology boundary is K58 + 454. Based on the HC method (GB50487-2008, 2008), the rock mass is classified into five classes, including I, II, III, IV and V, as shown in Table 3. In the HC method, the cumulative score T is used as the primary criterion for dividing the rock mass classes. The method comprehensively considers the factors of the ratings of rock strength, rock mass intactness degree, discontinuity conditions, groundwater condition and the attitude of the main discontinuity plane. Meanwhile, the strength-to-stress ratio, S, is also calculated below to account for the stress state effect on the surrounding rock:

$$S = \frac{R_c K_v}{\sigma_m} \tag{17}$$

where R_c is the UCS of intact saturated rock, K_v is the intactness index of rock mass, and σ_m is the maximum principal stress of surrounding rock.

In the study area, the proportions of rock mass class from II to V are 8.13% (419), 66.74% (5555), 20.03% (1439) and 5.1% (125), respectively.

3.2. Data preprocessing

The established database contains a large number of useless data. According to the variation law of TBM operation parameters, the raw data can be processed by constructing the state discriminant function (SDF) to remove the useless data and obtain the whole TBM tunnelling cycles (Wang et al., 2018). The SDF is written as

$$SDF = f(v)f(RS)f(F)f(Tc) \tag{18}$$

$$f(x) = \begin{cases} 0 & (x = 0) \\ 1 & (x \neq 0) \end{cases} \tag{19}$$

$$SDF = \begin{cases} 0 & (\text{useless data}) \\ 1 & (\text{data of TBM tunnelling cycle}) \end{cases} \tag{20}$$

Through the above treatment, a total of 7525 TB M tunnelling cycles without useless data were obtained. Fig. 8 shows a complete TBM tunnelling cycle and the selection of valuable data for classifiers. There is a strong correlation between TBM machine parameters and rock mass quality. The TBM tunnelling cycle can be divided into the rising phase and stable phase, as shown in Fig. 8. The data of the stable phase can better reflect the rock mass quality of the construction area. By analysing the data, the duration of the rising phase is usually short and less than 5 min. The operational parameters near the end of a TBM tunnelling cycle may be unstable. Therefore, the data of the first 400 s and the last 300 s of each TBM tunnelling cycle should be removed, and the rest operational data are valid to be selected for classifiers. However, the data in the selected area may have some outliers. In this section, the boxplot method based on the quartile and the interquartile ranges is used to eliminate the outliers (Carter et al., 2009). Suppose D_s is the data point in the selected area, then the criterion for judging outliers is as follows:

$$L_{upper} = Q_3 + 1.5IQR \tag{21}$$

$$L_{lower} = Q_1 - 1.5IQR \tag{22}$$

$$IQR = Q_3 - Q_1 \tag{23}$$

$$D_s = \begin{cases} \text{outliers} & (D_s < L_{lower} \text{ or } D_s > L_{upper}) \\ \text{non-outliers} & (L_{lower} \leq D_s \leq L_{upper}) \end{cases} \tag{24}$$

where Q_3 is the upper-quartile, Q_1 is the lower-quartile, IQR is the inter-quartile range, L_{upper} is the upper limit of non-outliers, and L_{lower} is the lower limit of non-outliers.

Finally, the mean value of the rest operation data without outliers is calculated as the inputs of the classifier to predict the rock mass classification.

3.3. Selection of the input features

Selection of appropriate input features has an essential impact on the prediction effect of the model. Different scholars have proposed many different feature selection methods (Kumar and Minz, 2014). In this section, the Gini index in RF is used to carry out the feature selection.

Table 2
Main technical parameters of open TBM.

Parameter	Design value
Machine type	Open TBM
Total weight (t)	~180
Number of cutters	56
Cutterhead rated torque (kN m)	8410 (at 3.97 rev/min)
Cutterhead rotation speed (rev/min)	0–3.97–7.6
Excavation diameter (mm)	8030
Maximum thrust (kN)	23,260
Driving power (kW)	3500
Maximum support force of TBM boots (kN)	46,028
Maximum advance rate (mm/min)	120

The value of the Gini index is inversely proportional to the effect of node split. Therefore, the importance of features can be ranked by calculating mean decrease Gini (Shang et al., 2007). The variable importance measures (*VIM*) of a feature X_j on node k is as follows:

$$VIM_{jk}^{(\mathrm{Gini})} = GI_k - GI_\mathrm{l} - GI_\mathrm{r} \tag{25}$$

where GI_l and GI_r are the Gini indices of the new left and right nodes after node split, respectively.

Then, the importance of feature X_j on the i-th DT can be expressed as

$$VIM_{ij}^{(\mathrm{Gini})} = \sum_{k\in K} VIM_{jk}^{(\mathrm{Gini})} \tag{26}$$

where K is the node collection in the RF. Suppose that there are N_c trees in the RF, the importance of feature can be obtained as

$$VIM_{j}^{(\mathrm{Gini})} = \sum_{i=1}^{N_\mathrm{c}} VIM_{ij}^{(\mathrm{Gini})} \tag{27}$$

The normalised result of the importance score of feature X_j in the RF is finally obtained as

$$VIM_j = \frac{VIM_j^{(\mathrm{Gini})}}{\sum_{j=1}^{n} VIM_j^{(\mathrm{Gini})}} \tag{28}$$

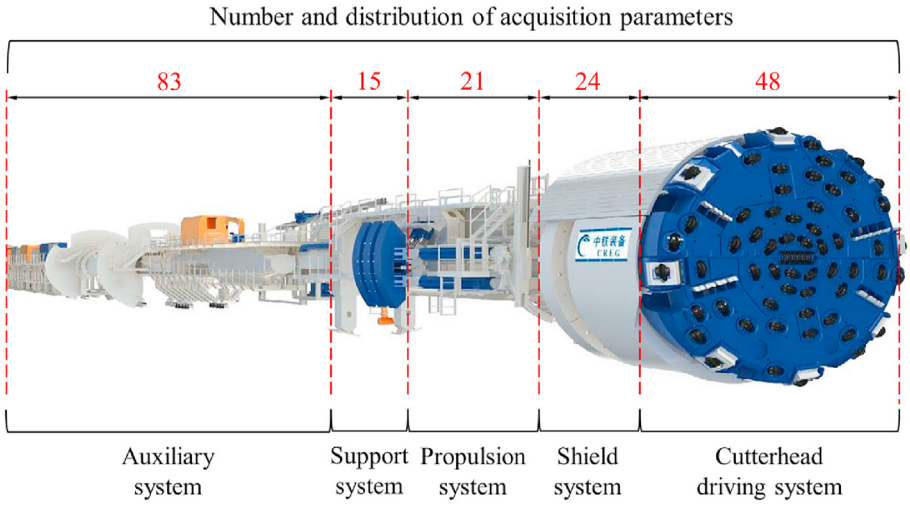

Fig. 5. TBM systems and the acquisition parameters distribution.

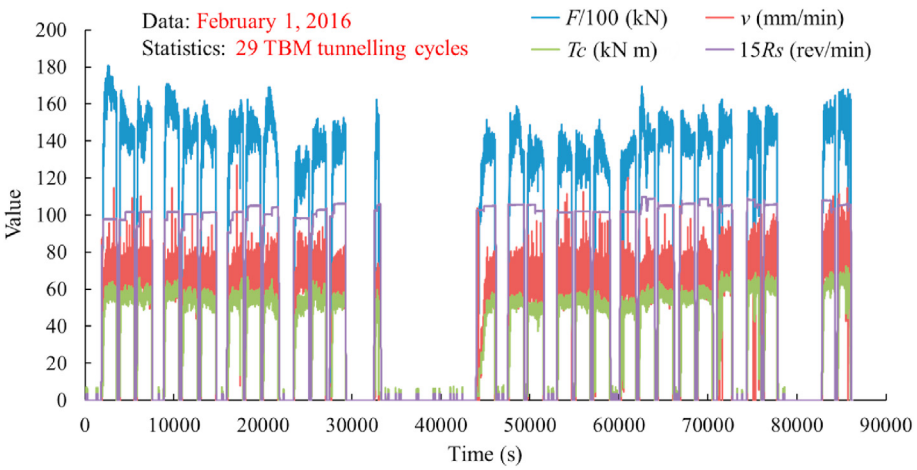

Fig. 6. Variation of four key TBM operation parameters in a day. *v* is the advance rate, *RS* is the cutterhead rotational speed, *F* is the total thrust, and *Tc* is the cutterhead torque.

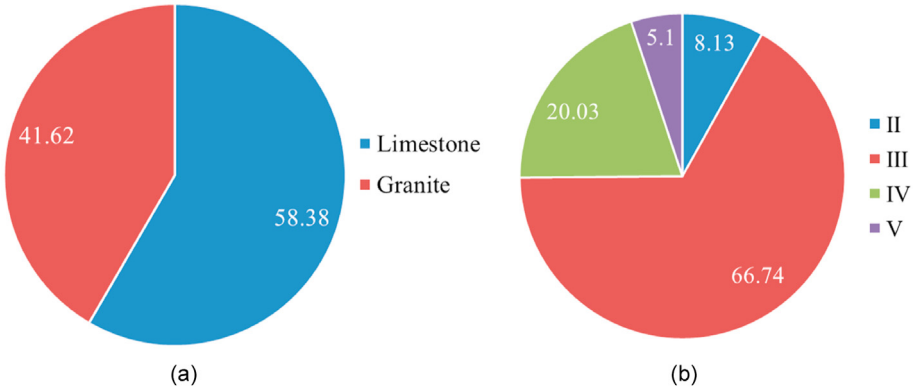

Fig. 7. Statistics of (a) lithology and (b) rock mass classification in the study area (unit: %).

Table 3
Descriptions of rock mass classification as per GB50487-2008 (2008).

Rock mass class	T	S	Stability evaluation of surrounding rock
I	$T > 85$	>4	Stable. The surrounding rock can be stable for a long time, and generally there is no unstable block
II	$85 \geq T > 65$	>4	Basically stable. The surrounding rock is stable as a whole and will not produce plastic deformation, and local block may fall off
III	$65 \geq T > 45$	>2	Local stability is poor. The surrounding rock will produce plastic deformation locally, and collapse or damage may occur without support. For intact soft rock, it may be temporarily stable
IV	$45 \geq T > 25$	>2	Unstable. The self-stability of surrounding rock is poor, and many kinds of large-scale deformation and failure may occur
V	$T \leq 25$	–	Extremely unstable. The surrounding rock is not self-stable, and the deformation and failure are obvious

Note: When the strength-to-stress ratio S of the rock mass in classes I, II, III and IV is less than the specified value in Table 3, the rock mass classification shall be reduced by one grade.

The importance of feature finally calculated is the relative value, and the sum of the *VIM* values of all features is equal to 1.

On the other hand, if the two features are highly correlated, they have similar trends and may carry similar information. The existence of such features will degrade the performance of some classifiers. Therefore, after sorting the features by the variable importance measures of RF, the highly correlated features are eliminated based on the Pearson correlation coefficient, which can be calculated as Eq. (29). In this section, If the Pearson correlation coefficient between the two feature is greater than 0.9, the two features are considered to be highly correlated, and we only keep one of them.

$$r = \frac{\sum_{q=1}^{N}(X_q - \overline{X})(Y_q - \overline{Y})}{\sqrt{\sum_{q=1}^{N}(X_q - \overline{X})^2}\sqrt{\sum_{q=1}^{N}(Y_q - \overline{Y})^2}} \tag{29}$$

where r is the Person correlation coefficient, X_i and Y_i are the two different variables, $\overline{X}$ is the mean value of X_i, $\overline{Y}$ is the mean value of Y_i, and N is the number of variables.

Based on the above data processing method, conducting the feature selection for 191 TB M operation parameters, and finally, 10 features are selected as the inputs of classifiers, including cutterhead rotational speed (n), pitch angle of gripper shoes (*Pags*), gear sealing pressure (*Gsp*), pressure of gripper shoes (*Pgs*), output frequency of main drive motor (*Ofdm*), internal pump pressure (*Ipp*), penetration rate (*Pr*), control pump pressure (*Cpp*), torque penetration index (*TPI*), and roll position of gripper shoes (*Rpgs*). Fig. 9 shows the normalised importance of the selected 10 features. The statistics of each selected features are shown in Table 4. The statistical indicators of each feature include minimum value (Min),

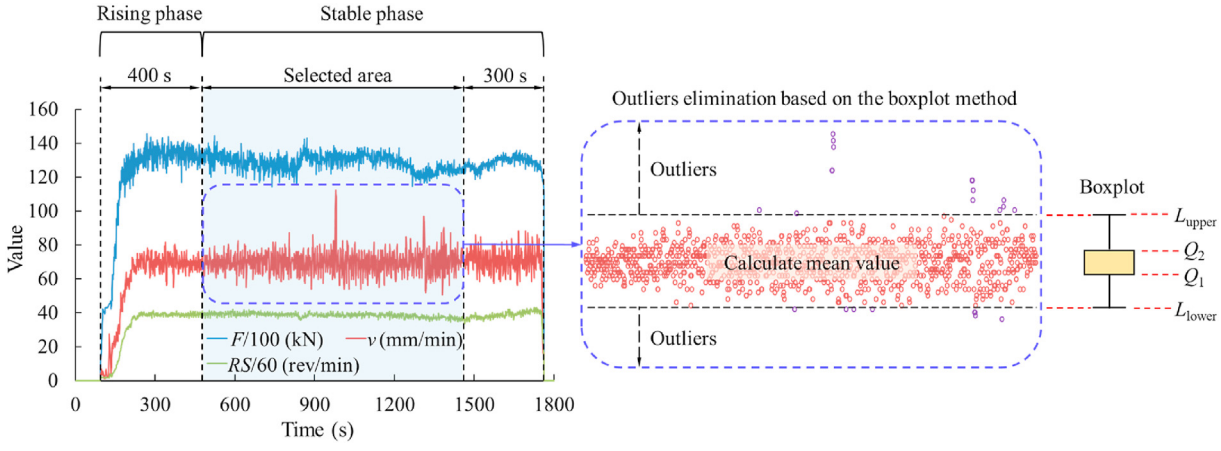

Fig. 8. A complete TBM tunnelling cycle and the selection of valid data for classifiers.

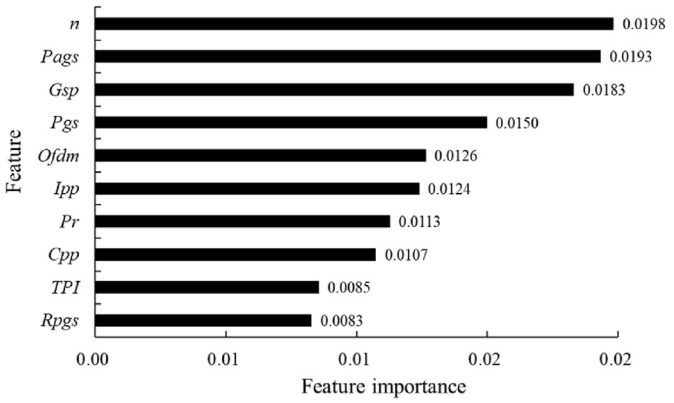

Fig. 9. Normalised importance of the selected 10 features.

maximum value (Max), mean value (Mean) and standard deviation (Std).

The physical meanings of the 10 selected features or their relevance with the rock mass quality are as follows (Liu et al., 2021; Jing et al., 2019): *n* and Pr reflect the rock-breaking efficiency of TBM. *Gsp* reflects the change of control valve and flowmeter for the TBM lubrication system caused by the variety of rock mass quality. *Pgs*, *Pags* and *Rpgs* reflect the state of reaction force on the TBM when advancing under different rock mass qualities. *Cpp* and *Ipp* reflect the flow and pressure outputs of the TBM thrust hydraulic system, respectively. *TPI* is the cutterhead torque required to advance unit penetration rate, reflecting the rock mass boreability. *Ofdm* is a parameter of the TBM variable frequency drive motor, influencing the cutterhead rotational speed through the reduction ratio relationship.

4. Case study and analysis

4.1. Model establishment

In this section, we firstly establish seven individual classifiers, including SVM, KNN, RF, GBDT, DT, LR, and MLP. Then, by using SVM, KNN, RF and GBDT as the base classifiers, and the GBDT as the meta-classifier, the stacking ensemble classifier is established. Seven individual classifiers are used for comparison with the stacking ensemble classifier. Fig. 10 shows the flowchart of the rock mass prediction for each classifier. All classifiers are implemented by the TensorFlow package in PyCharm using the Python language. Moreover, the training and testing of all classifiers were processed by a computer with a CPU of Intel(R) Core(TM) i7-7700 K @ 4.20 GHz in a Windows environment.

According to Section 3, 7538 TBM tunnelling cycles are obtained, and the ten features are selected as the inputs of the classifier. The rock mass class is used as the outputs of the classifier. In order to eliminate the influence of the data magnitude and dimension difference, it is necessary to carry out the data normalisation for each feature before the model training. In this section, the *Z*-score normalisation method is used to normalise the input features, making the mean value and the standard deviation of each feature to be 0 and 1, respectively. The calculation formula is as follows:

$$x_Z = \frac{x - \mu}{\sigma} \tag{30}$$

where x is an input parameter, x_Z is the input parameter after normalisation, μ is the mean value of the input samples, and σ is the standard deviation of the input samples.

Since the input and output of the machine learning model should be numerical data, it is needed to carry out an encoding process for the labelled data. In this section, the one-hot encoding method (Potdar et al., 2017) is adopted, and the rock mass classes of II, III, IV and V are encoded as (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0) and (0, 0, 0, 1), respectively, as listed in Table 5. Then, the preprocessed dataset is divided into a training set and a test set using simple random sampling. There are 6784 samples in the training set and 754 samples in the test set, accounting for 90% and 10%, respectively.

After the above data preparation, the stacking ensemble and individual classifiers are established. The training set is used to construct 10-fold CV dataset, and the hyperparameter optimisation is carried out based on the 10-fold CV dataset. Finally, the test set is used to test each classifier, and each classifier is evaluated based on the evaluation metrics in Section 2.4.

Table 4
Statistics of the selected features.

Feature	Unit	Min	Max	Mean	Std
n	kN m	4	7.62	6.54	0.71
Pags	°	−4.9	3	−0.13	1.59
Gsp	kN	1.26	11.18	6.47	1.52
Pgs	kN	215.82	351.28	302.67	22.08
Ofdm		0	96	81.93	10.54
Ipp	kN	0.46	5.74	3.98	1.17
Pr	mm/rev	0.89	18.65	9.93	2.21
Cpp	kN	122.3	131.59	129.11	1.34
TPI	kN rev/mm	14.77	777.86	261.62	94.57
Rpgs	°	−4.75	2.64	−0.64	1.02

4.2. Hyper-parameter optimisation of classifiers

Different machine learning models have different hyper-parameters, which should be set before the model training. Hyper-parameters are the essential factor affecting the performance of machine learning models (Feurer and Hutter, 2019). The hyper-parameters of different models that we mainly consider are as follows:

(1) For the SVM classifier, the key hyper-parameters are the penalty coefficient *C* and the RBF kernel coefficient *g*. Hyper-parameter *C* reflects the tolerance of the SVM model to errors, and hyper-parameter *g* determines the distribution of the data mapped to the new feature space.
(2) For the KNN classifier, the key hyper-parameters are the *n_neighbours* and the *weights*. Hyper-parameter *n-neighbours* is the number of neighboring points when determining the sample classification, and it can be determined from 1 to 15. Hyper-parameter *weights* is the distance-based voting weight of neighboring points, and it can be set as distance or uniform, considering the weight or not, respectively.
(3) For the DT classifier, the key hyper-parameters are the *criterion*, *min_samples_split* and *min_samples_leaf*. Hyper-parameter *criterion* is the feature selection criterion of decision. Hyper-parameter *min_samples_split* is the minimum number of samples required to split an internal node, and hyper-parameter *min_samples_leaf* is the minimum number of samples required in a terminal node for a split to be valid.
(4) For the RF classifier, the key hyper-parameters are *min_samples_split*, *min_samples_leaf*, *n_estimators*, *max_depth* and *max_features*. Among them, hyper-parameters *min_samples_split* and *min_samples_leaf* have the same meaning as DT classifier. Hyper-parameter *n_estimators* is the number of the DTs, hyper-parameter *max_depth* is the maximum depth of

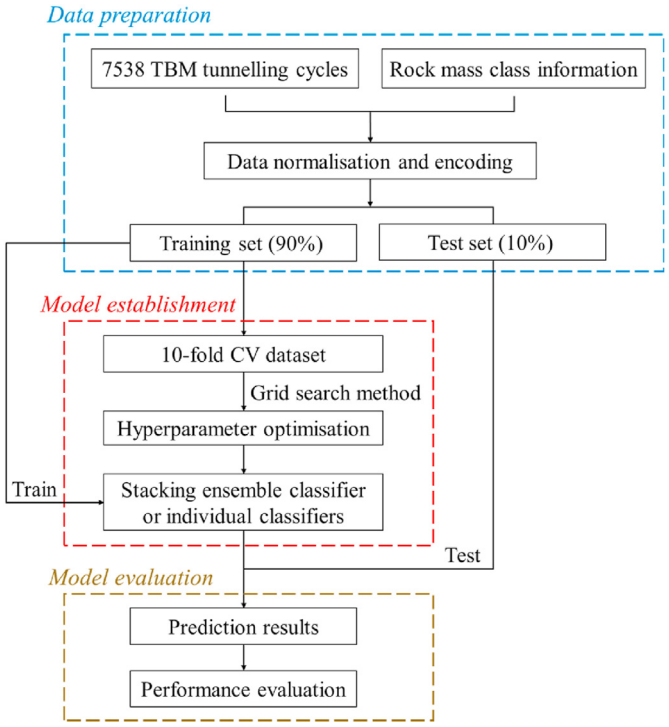

Fig. 10. Flowchart of the rock mass prediction for each classifier.

Table 5
One-hot encoding results of each rock mass class.

Rock mass class	One-hot encoding result
II	(1, 0, 0, 0)
III	(0, 1, 0, 0)
IV	(0, 0, 1, 0)
V	(0, 0, 0, 1)

each DT, and hyper-parameter *max_feature* is the number of features randomly selected for each DT.

(5) For the GBDT classifier, the key hyper-parameters are *learning_rate*, *n_estimators*, *max_depth* and *max_features*. Hyper-parameter *learning_rate* is the weight reduction coefficient of each weak learner, and the other three hyper-parameters are the same as RF classifier.

(6) For the LR classifier, the key hyper-parameters are *max_iter*, *c* and *solver*. Hyper-parameter *max_iter* is the maximum number of iteration, hyper-parameter *c* is the reciprocal of the regularisation coefficient, and hyper-parameter *solver* determines the optimisation method of a loss function.

(7) For the MLP classifier, the key hyper-parameters are *learning_rate*, *max_iter* and *activation*. The meanings of the first two parameters are the same as mentioned above. Hyper-parameter *activation* is the activation function of neurons.

There are several commonly used methods for hyper-parameters tuning, including the grid search method (Wistuba et al., 2015), metaheuristic algorithms (e.g. particle swarm optimisation (PSO), grey wolf optimisation (GWO), whale optimisation algorithm (WOA), moth flame optimisation (MFO), and multi-verse optimisation (MVO)) (Zhou et al., 2021a, b), hold-out method, random search method, and leave-one-out method. (Kardani et al., 2020). In this section, the hyper-parameter optimisation is conducted for each classifier based on 10-fold CV accuracy as the evaluation index. The optimal hyper-parameters of each classifier are tuned by the grid search method. Table 6 shows the optimisation results of hyper-parameters. The hyper-parameters of stacking ensemble classifier are set based on the optimisation results of corresponding individual classifiers. The optimal hyper-parameters are used to set each classifier before the model training. In addition to the optimised hyper-parameters, the other initialisation hyper-parameters of each classifier are set as the default value of each classifier function in Scikit-learn libraries.

4.3. Prediction results and performance evaluation

In this section, 90% of the TBM operation data and corresponding rock mass class are used as the training set to train the established eight classifiers. The remaining 10% of the data are used as the test set to test the trained classifiers. In order to ensure the comparability among the classifiers, all the classifiers are established based on the same training and test sets, and the training and testing process of each classifier is repeated 10 times to determine the model performance. The values of different evaluation metrics are obtained by calculating the mean values of 10 repeated tests. Table 7 lists the calculation time consumed of different classifiers under optimal hyper-parameters. It can be seen that the training times of GBDT and stacking ensemble classifiers are relatively long, which are up to 56.367 s and 68.295 s, respectively. The training time of the other six classifiers is less than 3 s. However, the prediction time consumed of each classifier for the test set is less than 0.3 s, which can be considered as 'real-time' prediction based on the trained classifiers.

Fig. 11 shows the prediction results of different classifiers on the test set. As can be learned from the figures, the misclassification ratio of samples belonging to classes II and V is high, and the misclassification ratio of samples belonging to the other two classes is relatively low. The reason for the above phenomenon is that the sample set is imbalanced. For the total samples, trainging set samples and test set samples, the samples belonging to classes II and V are all less than 10%. Additionally, it can be easily seen that the proposed stacking ensemble classifier has the best prediction performance on rock mass classification among all classifiers, with fewest misclassification samples.

Table 6
Optimisation results of hyper-parameters.

Classifier	Optimal setting of initialisation hyper-parameters
SVM	*kernel* = 'rbf', *C* = 6.86, *g* = 47.03
KNN	*n_neighbours* = 5, *weights* = 'distance'
RF	*min_samples_split* = 5, *min_samples_leaf* = 2, *n_estimators* = 300, *max_depth* = 14, *max_features* = 6
GBDT	*learning_rate* = 0.2, *n_estimators* = 200, *max_depth* = 15, *max_features* = 5, *min_samples_split* = 5, *min_samples_leaf* = 2
DT	*min_samples_split* = 6, *min_samples_leaf* = 4, *criterion* = 'Gini'
LR	*solver* = 'sag', *c* = 1, *max_iter* = 400
MLP	*learning_rate* = 0.01, *max_iter* = 800, *activation* = 'relu', hidden_layer_sizes = (15, 15, 15)

Table 7
Calculation time consumed of different classifiers under optimal hyper-parameters.

Classifier	Traning time consumed (s)	Prediction time consumed (s)
SVM	1.282	0.159
KNN	0.205	0.026
RF	7.213	0.051
GBDT	56.367	0.071
DT	0.273	0.002
LR	0.242	0.001
MLP	2.699	0.002
Stacking	68.295	0.331

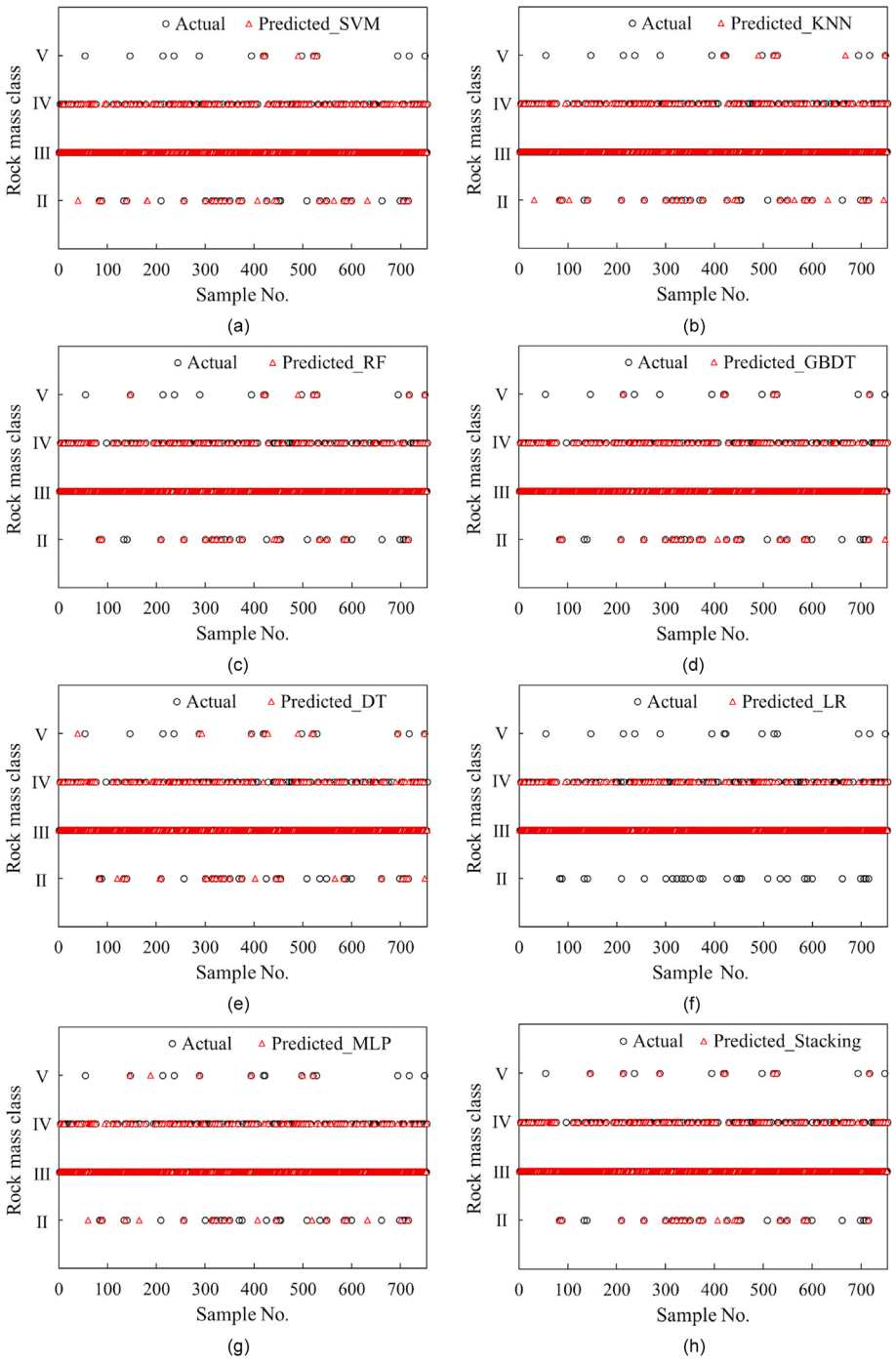

Fig. 11. Prediction results of different classifiers on test set: (a) SVM, (b) KNN, (c) RF, (d) GBDT, (e) DT, (f) LR, (g) MLP, and (h) Stacking.

In order to quantitatively analyse the prediction results and evaluate the performance of different classifiers, the evaluation metrics proposed in Section 2.4 are used to evaluate each classifier's performance, and the comparison between stacking ensemble classifier and individual classifiers is also analysed. Table 8 lists the evaluation metrics of each classifier. Fig. 12 shows the confusion matrix based on the *REC* of each classifier. It can be seen from Table 8 and Fig. 12 that:

(1) The stacking ensemble classifier has the best prediction performance with the four highest evaluation metrics; the values of $ACC_{\_Total}$, *Kappa*, $PRC_{\_Total}$, $REC_{\_Total}$ and F_{1_Total} are 93.1%, 0.823, 0.93, 0.931 and 0.928, respectively. The prediction performance of the stacking ensemble classifier on different rock mass classes is also the best compared to other individual classifiers. Taking *REC* as an example, the *REC* values of stacking ensemble classifier for classes II, III, IV and V are 70% (30/754), 98.4% (557/754), 81.7% (153/754) and 57.1% (14/754), respectively. The *REC* of stacking ensemble classifier is higher than the other seven individual classifiers. Especially for the class V, the *REC* values of the seven individual classifiers are all less than 50%. However, the prediction performance of the stacking ensemble classifier for class V is greatly improved with *REC* up to 57.1%. Also, the relative

Table 8
Evaluation metrics of each classifier.

Classifier	Rock mass class	Evaluation metrics					
		ACC (%)	*Kappa*	*PRC*	*REC*	F_1	Support
SVM	II			0.724	0.7	0.712	30
	III			0.921	0.948	0.935	557
	IV			0.803	0.771	0.787	153
	V			0.8	0.286	0.421	14
	Total	89	0.723	0.887	0.89	0.886	754
KNN	II			0.714	0.667	0.69	30
	III			0.906	0.948	0.926	557
	IV			0.796	0.706	0.747	153
	V			0.714	0.357	0.476	14
	Total	87.7	0.682	0.872	0.877	0.872	754
RF	II			0.9	0.6	0.72	30
	III			0.919	0.975	0.946	557
	IV			0.859	0.758	0.806	153
	V			0.875	0.5	0.636	14
	Total	90.7	0.756	0.905	0.907	0.903	754
GBDT	II			0.87	0.667	0.755	30
	III			0.933	0.982	0.957	557
	IV			0.885	0.804	0.842	153
	V			1	0.429	0.6	14
	Total	92.3	0.801	0.922	0.923	0.919	754
DT	II			0.593	0.533	0.561	30
	III			0.907	0.946	0.926	557
	IV			0.8	0.706	0.75	153
	V			0.545	0.429	0.48	14
	Total	87.1	0.672	0.866	0.871	0.868	754
LR	II			0	0	0	30
	III			0.823	0.95	0.882	557
	IV			0.595	0.431	0.5	153
	V			0	0	0	14
	Total	78.9	0.38	0.728	0.789	0.753	754
MLP	II			0.55	0.367	0.440	30
	III			0.855	0.923	0.888	557
	IV			0.661	0.549	0.6	153
	V			0.667	0.286	0.4	14
	Total	81.3	0.502	0.8	0.813	0.802	754
Stacking	II			0.875	0.7	0.778	30
	III			0.94	0.984	0.961	557
	IV			0.899	0.817	0.856	153
	V			1	0.571	0.727	14
	Total	93.1	0.823	0.93	0.931	0.928	754

relationship of the other two evaluation metrics (i.e. *PRC* and F_1) among different classifiers is similar to *REC*. The above analysis shows that the proposed stacking ensemble classifier has a powerful generalisation ability.

(2) Among the individual classifiers, GBDT and RF show relatively good performance and generalisation ability than other individual classifiers. In fact, the GBDT and RF also belong to the ensemble learning classifiers, which combining multiple DTs in different ways. While in this study, GBDT and RF are used as the base classifiers of stacking ensemble classifier, thus they are regarded as the individual classifiers.

(3) The prediction performance of SVM, KNN, DT and MLP are relatively poor, which have more misclassified samples than GBDT, RF and stacking ensemble classifiers. Taking the prediction of class II as an example, the *REC* values of GBDT, RF and stacking ensemble classifiers are 0.982, 0.975 and 0.984, respectively. However, the *REC* values of SVM, KNN, DT and MLP are as low as 0.923–0.948, with more samples belonging to class III misclassified as class IV. Thus, among the four classifiers (i.e. SVM, KNN, DT and MLP), the performance of the first three classifiers is relatively good with $ACC_{\_Total}$ of 87.1%–89%, while the performance of the MLP classifier is relatively poor with $ACC_{\_Total}$ of 81.3%.

(4) The LR classifier has the worst prediction performance on rock mass classification, and its evaluation metrics are all the lowest among all established classifiers. Additionally, it can be seen from Fig. 12f that the LR classifier cannot predict classes II and V. As a result, the samples belonging to class II are all misclassified as classes III (83.3%) and IV (16.7%). Moreover, the samples belonging to class V are also misclassified as classes III (14.3%) and IV (85.7%). The above analysis shows that the generalisation ability of the LR classifier is inferior.

(5) As for the *Kappa* metric, the *Kappa* value of the stacking ensemble classifier is 0.823, which means the strength of agreement is almost perfect according to Table 1. The strength of agreement of the LR classifier is inferior, with the *Kappa* value of 0.38. The strength of agreement of the MLP classifier is moderate, with the *Kappa* value of 0.502. In contrast, the strength of agreement of the other five classifiers are all good, with the *Kappa* value of 0.61–0.8. Additionally, it can be seen from Table 8 that there is a positive correlation between *ACC* and *Kappa*, and the value of *Kappa* is smaller than *ACC*. The *ACC* and *Kappa* of stacking ensemble classifier are all greater than those of the other seven individual classifiers, showing that the stacking technique can effectively improve the model performance.

Fig. 13 shows the error histogram of different classifiers. The error values of classification problems are discrete. Since the rock mass classes have four levels of II, III, IV and V in our study, the error of the established classifiers is within the range of {−3, −2, −1, 0, 1, 2, 3}. The value of error represents the level difference between the actual and predicted classes. Among them, $error = 0$ means that the predicted class is the same as the actual class, the positive error means that the level of the predicted class is higher than that of the actual class, and the negative error means that the level of the predicted class is lower than that of the actual class. As can be seen from Fig. 13, the error value of each classifier is less than 3. Moreover, the error values of the most misclassified samples are −1 and 1. For RF, GBDT and stacking ensemble classifiers, there is only one sample with the absolute value of error reaching 2, and the errors of the rest of the misclassified samples are −1 and 1. However, the frequency with $|error| = 1$ of stacking ensemble classifier is less than that of RF and GBDT classifiers. For other individual classifiers, the frequency of samples with $|error| = 2$ is more. It can be seen that the proposed stacking ensemble classifier is more reasonable in the classification of rock mass, in which the misclassified samples are generally incorrectly predicted as the adjacent classes.

The ROC curves are also implemented to evaluate the prediction performance in rock mass classification. Fig. 14 shows the ROC curves and corresponding *AUC* value of different classifiers. Table 9 lists the micro-average and macro-average *AUC* values of different classifiers. It can be seen that: (1) The micro-average and macro-average *AUC* values of the stacking ensemble classifier are 0.989 and 0.98, respectively, which shows the best prediction performance among all the classifiers. It is followed by GBDT classifier (micro-average $AUC = 0.985$ and macro-average $AUC = 0.969$) and RF classifier (micro-average $AUC = 0.982$ and macro-average $AUC = 0.968$). The SVM classifier (micro-average $AUC = 0.97$ and macro-average $AUC = 0.911$) and KNN classifier (micro-average $AUC = 0.960$ and macro-average $AUC = 0.903$) can be also considered as the good classifiers. The prediction performance of DT classifier (micro-average $AUC = 0.949$ and macro-average $AUC = 0.892$) and MLP classifier (micro-average $AUC = 0.938$ and macro-average $AUC = 0.853$) are relatively poor. The LR classifier has the worst accuracy with micro-average $AUC = 0.929$ and macro-average $AUC = 0.812$. (2) Because the macro-average *AUC* is more influenced by the minority class samples than macro-average *AUC*, the difference between these two metrics can reflect the

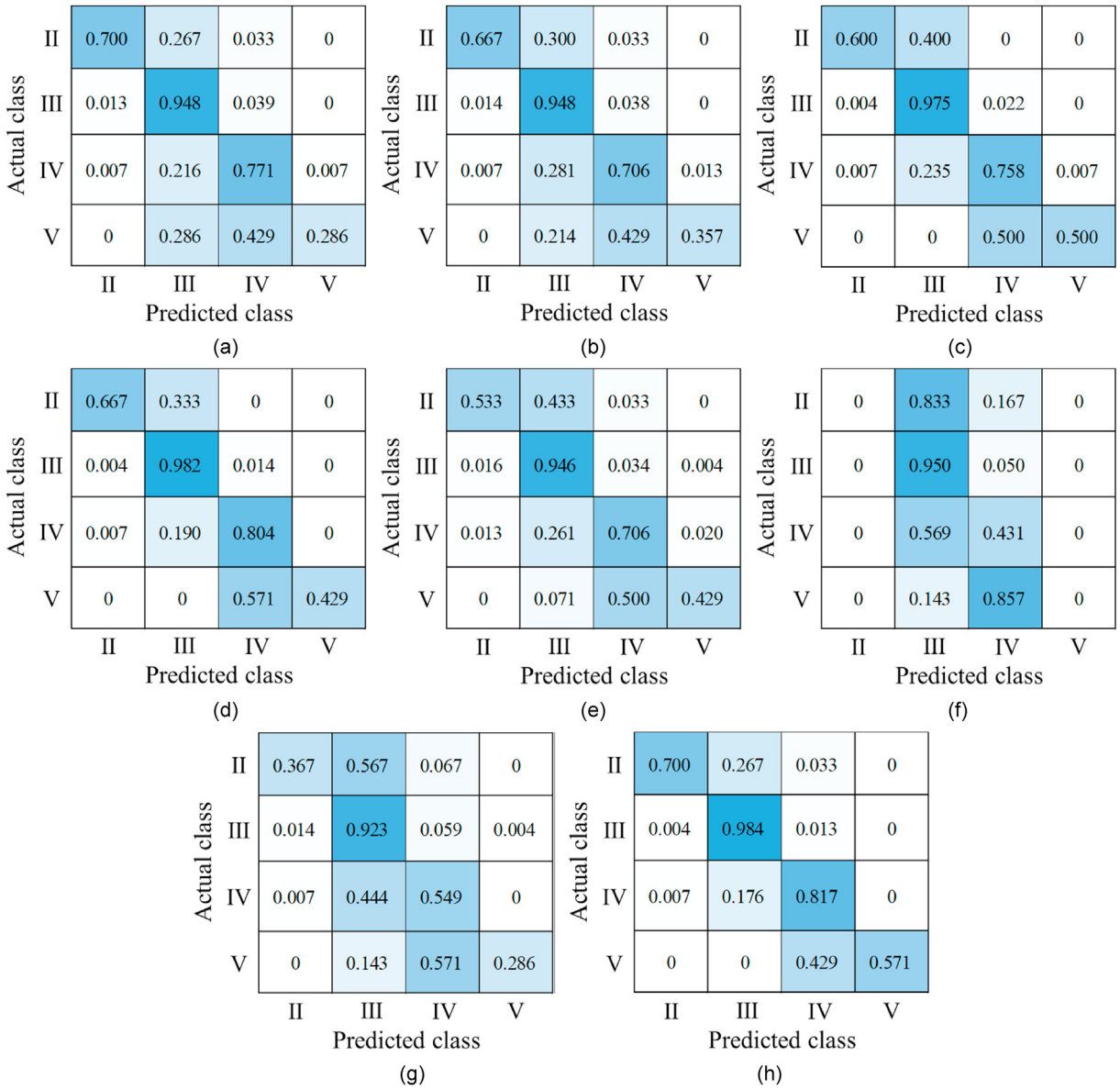

Fig. 12. Confusion matrix based on *REC* of each classifier: (a) SVM, (b) KNN, (c) RF, (d) GBDT, (e) DT, (f) LR, (g) MLP, and (h) Stacking.

learning ability of minority to imbalanced data. As can be seen from Table 9, the difference value of stacking ensemble classifier is the smallest, which shows better learning ability and improvement of the performance than individual classifiers.

4.4. Analysis of fitting effect of different classifiers

Over-fitting is a common problem in the training process, which means the prediction performance on the training set is much higher than that on the test set (Cawley and Talbot, 2010). It also indicates that the generalisation ability of the classifier is poor. Fig. 15 shows the relationship between the prediction accuracy of each classifier on the training and test sets. It can be seen that all the data points fall above the line $x = y$, which represents that the prediction accuracy on the training set is higher than that on the test set. Generally, if the difference between prediction accuracy on the training set and the test set is too large (i.e. the data point in Fig. 15 is far from the line $x = y$), over-fitting may occur in the training process. At present, there is no clear rule about how much difference of prediction accuracy between the training and test sets belongs to over-fitting. For most established classifiers, the difference between prediction accuracy on the training and test sets is relatively small. The data point of the KNN classifier falls above the line $x = 0.9y$, while the data point of the other seven classifiers all fall below the line $x = 0.9y$. More specifically, Table 10 lists the statistics of the prediction accuracy on the training set and the test set. In the first place, the difference of prediction accuracy on the training and test sets for the KNN classifier is 11.6%, while those of other classifiers are all less than 10%. In the second place, the ratio of prediction accuracy on the training set and the test set for the KNN classifier is 0.88, while the ratios of other classifiers are all less than 0.9. Additionally, for the LR classifier, the prediction accuracy on the training set is close to that of the test set, and the data point in Fig. 15 is close to line $x = y$. However, the prediction accuracy of

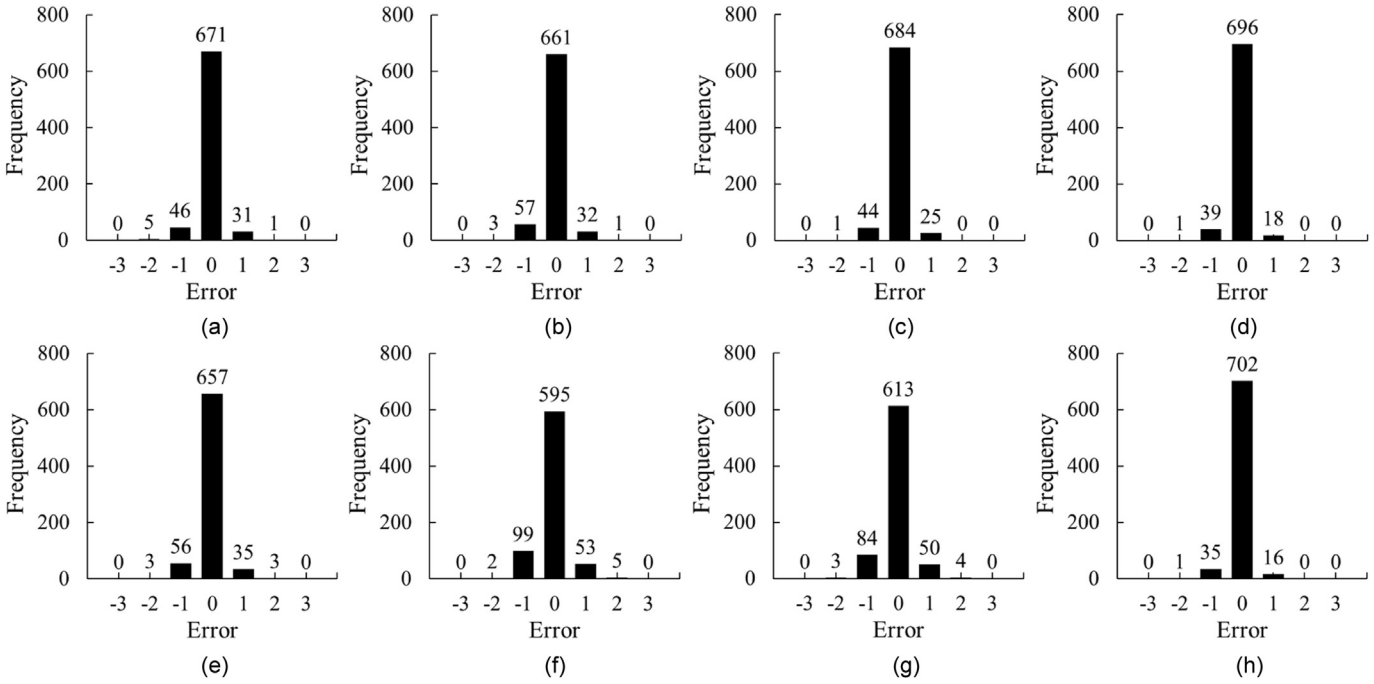

Fig. 13. Error histogram of different classifiers: (a) SVM, (b) KNN, (c) RF, (d) GBDT, (e) DT, (f) LR, (g) MLP, and (h) Stacking.

the LR classifier is low and cannot identify classes II and V. Therefore, the fitting effect of the LR classifier for the training set is also poor. The above analysis shows that, except for KNN and LR classifiers, the fitting effect of the other six classifiers can be considered as good.

In general, through the analysis of various aspects, it can be concluded that the proposed stacking ensemble classifier has good prediction performance and strong generalisation ability for rock mass classification. Therefore, it can be used to real-time and accurately predict the rock mass classes, which can help guide the adaptive adjustment for TBM in the tunnelling process.

5. Discussion

5.1. Influence of sample imbalance on classifier performance

In this section, the influence of the sample imbalance on the prediction effect is discussed. According to the above analysis, we can see that there are significant differences in the number of samples with different rock mass classes, as shown in Fig. 16. The overall ratio of the training and test sets is 6784/754 (i.e. 9/1), and the proportion of different classes in the training and test sets is also about 9/1. In the training set, the number of samples with different rock mass classes varies greatly, among which the number of samples of classes II, III, IV and V are 377, 5004, 1293 and 111, respectively. This may lead to a better fitting effect for samples of classes III and IV (or even over-fitting to a certain extent), and a worse fitting effect for the samples of classes II and V. Fig. 17 shows the relationship between the *REC* value of each classifier and the number of training samples. It can be seen that:

(1) The *REC* values of different classifiers on the test set positively correlate with the number of training samples. The more the samples in a certain class, the higher the *REC* value on the test set.

(2) With the increasing number of samples, the *REC* difference of different classifiers are gradually decreased. For the number of samples less than 500 (i.e. classes V and II), the *REC* differences of different classifiers are relatively significant. For the number of samples between 1000 and 1500 (i.e. class IV), the *REC* differences of different classifiers are decreased to a certain extent. When the number of samples increases to about 5000 (i.e. class V), the *REC* differences of different classifiers become relatively small.

(3) For different rock mass classes, the prediction performance of the stacking ensemble classifier is the best, which shows that the ensemble learning model has a more robust learning ability and generalisation ability than single classifiers for small and imbalanced samples. Additionally, the relevant analysis in Section 4.4 also shows that the imbalance of samples impacts the classifier's prediction effect.

In this section, the SMOTE is used to process the imbalanced samples, making the number of samples of classes V and II increased to 1000, while keeping the number of samples of other classes unchanged. After oversampling the samples of classes V and II, a relatively balanced training set is obtained. The numbers of samples of classes II, III, IV and V in the relatively balanced training set are 1000, 4998, 1286 and 1000, respectively. The test set remains the same as before. Table 11 lists the statistics of the original imbalanced training set, the relativey balanced training and test sets. It can be seen that the sample proportion of different rock masses becomes more balanced after the SMOTE processing. Especially, the sample proportions of classes II and V are increased to 12.07% from 5.73% to 1.64%, respectively.

In this section, all the established classifiers are trained and tested based on the dataset shown in Table 11. Table 12 presents the prediction accuracy of different classifiers on the test set with the imbalanced training set and relatively balanced training set. Table 13 presents the difference of prediction accuracy between the training and test sets of different classifiers with imbalanced training set and relatively balanced training set. After the SMOTE

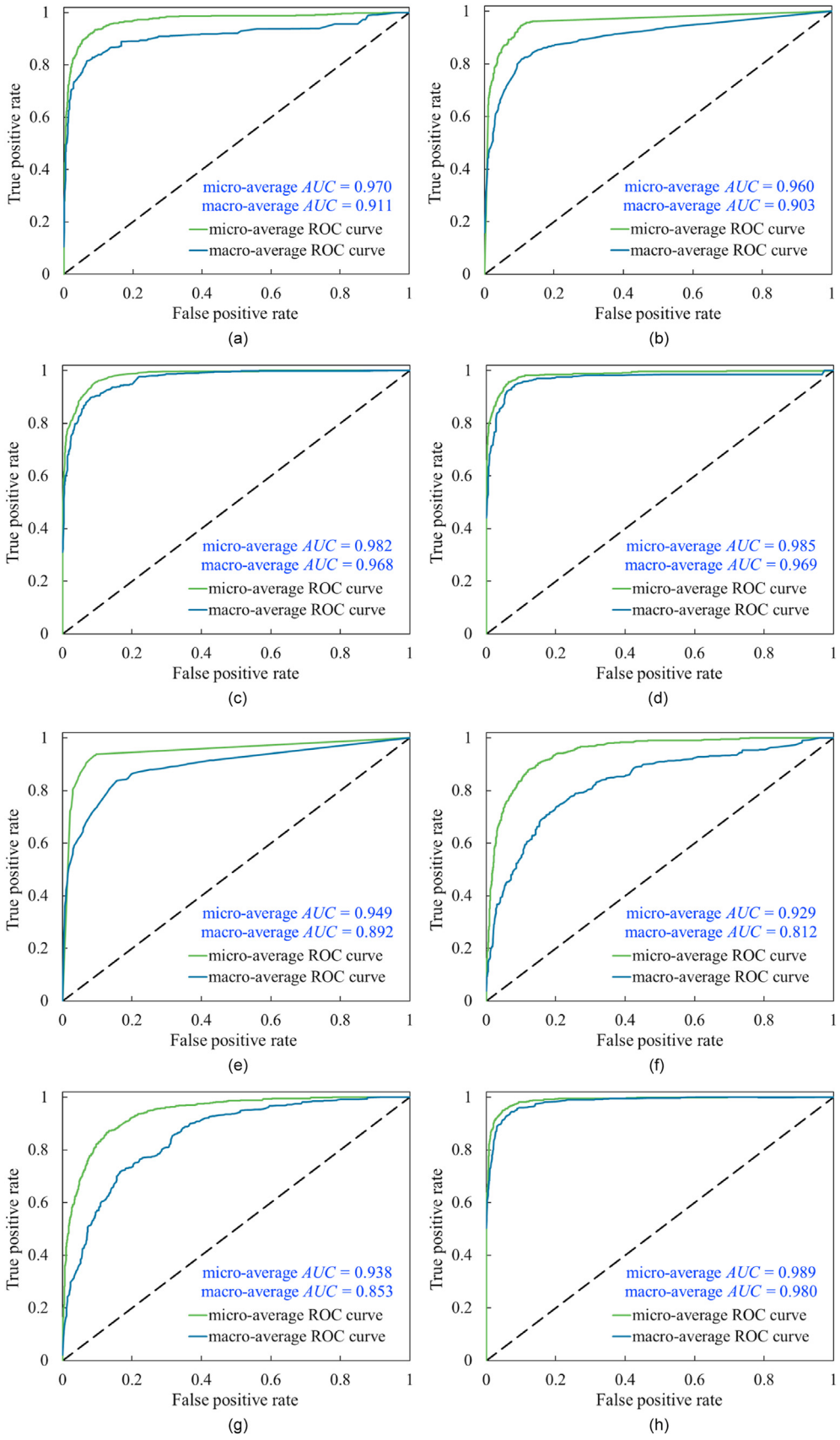

Fig. 14. ROC curves and corresponding *AUC* values of different classifiers: (a) SVM, (b) KNN, (c) RF, (d) GBDT, (e) DT, (f) LR, (g) MLP, and (h) Stacking.

Table 9
Micro-average and macro-average *AUC* values of different classifiers.

Classifier	Micro-average *AUC*	Macro-average *AUC*	Difference
SVM	0.97	0.911	0.059
KNN	0.96	0.903	0.057
RF	0.982	0.968	0.014
GBDT	0.985	0.969	0.016
DT	0.949	0.892	0.057
LR	0.929	0.812	0.117
MLP	0.938	0.853	0.085
Stacking	0.989	0.98	0.009

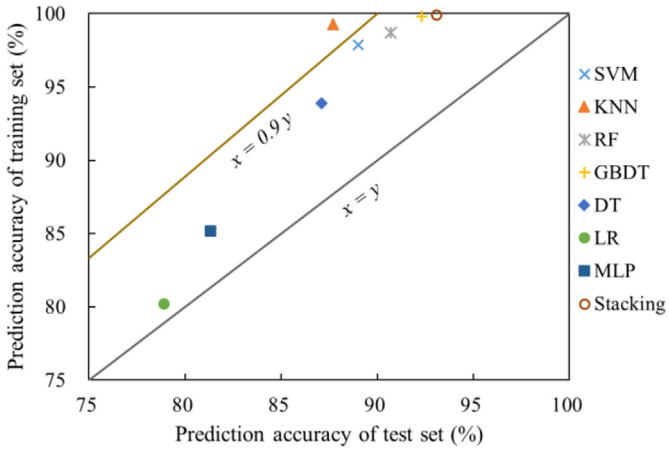

Fig. 15. Corresponding relationship between the prediction accuracy of each classifier on training and test sets.

Table 10
Statistics of the prediction accuracy on training and test sets.

Classifier	Prediction accuracy on training set, A_{train} (%)	Prediction accuracy on test set, A_{test} (%)	$A_{train}-A_{test}$	A_{test}/A_{train}
SVM	97.9	89	8.9	0.91
KNN	99.3	87.7	11.6	0.88
RF	98.7	90.7	8	0.92
GBDT	99.8	92.3	7.5	0.92
DT	93.9	87.1	6.8	0.93
LR	80.2	78.9	1.3	0.98
MLP	85.2	81.3	3.9	0.95
Stacking	99.9	93.1	6.8	0.93

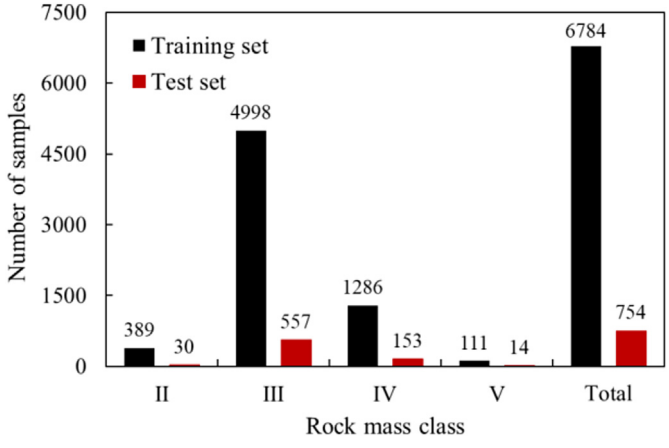

Fig. 16. Number of different rock mass classes in training set and test set.

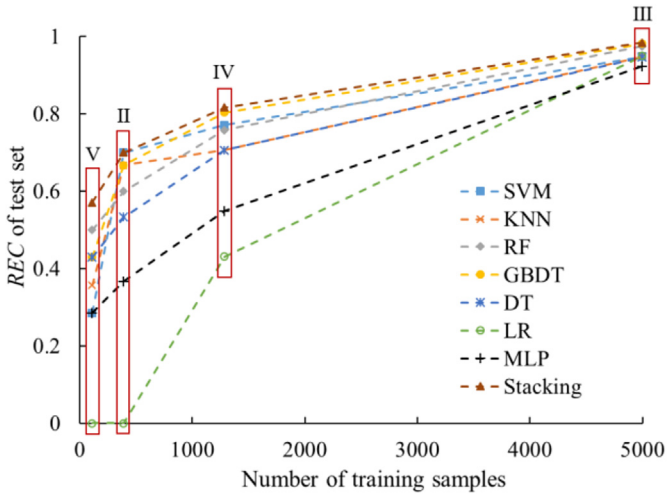

Fig. 17. Relationship between the *REC* value of each classifier and the number of training samples.

Table 11
Statistics of original imbalanced training set, relatively balanced training set and test set.

Rock mass class	Imbalanced training set		Relatively balanced training set		Test set	
	Number	Proportion (%)	Number	Proportion (%)	Number	Proportion (%)
II	389	5.73	1000	12.07	42	5.57
III	4998	73.67	4998	60.33	551	73.08
IV	1286	18.96	1286	15.52	146	19.36
V	111	1.64	1000	12.07	15	1.99
Total	6784	100	4786	100	754	100

Table 12
Prediction accuracy of difference classifiers on test set with imbalanced training set and relatively balanced training set.

Classifier	Prediction accuracy (%)		Variation (%)
	Imbalanced training set	Relatively balanced training set	
SVM	89	89.4	0.4
KNN	87.7	89.4	1.7
RF	90.7	91.9	1.2
GBDT	92.3	93.6	1.3
DT	87.1	87.5	0.4
LR	78.9	79.5	0.6
MLP	81.3	82.6	1.3
Stacking	93.1	94.2	1.1

Table 13
Difference of prediction accuracy between training and test sets of difference classifiers with imbalanced training set and relatively balanced training set.

Classifier	Difference of prediction accuracy (%)		Variation (%)
	Imbalanced training set	Relatively balanced training set	
SVM	8.9	8.8	0.1
KNN	11.6	10.6	1
RF	8	6.8	1.2
GBDT	7.5	6.4	1.1
DT	6.8	6.4	0.4
LR	1.3	1.1	0.2
MLP	3.9	3.3	0.6
Stacking	6.8	5.8	1

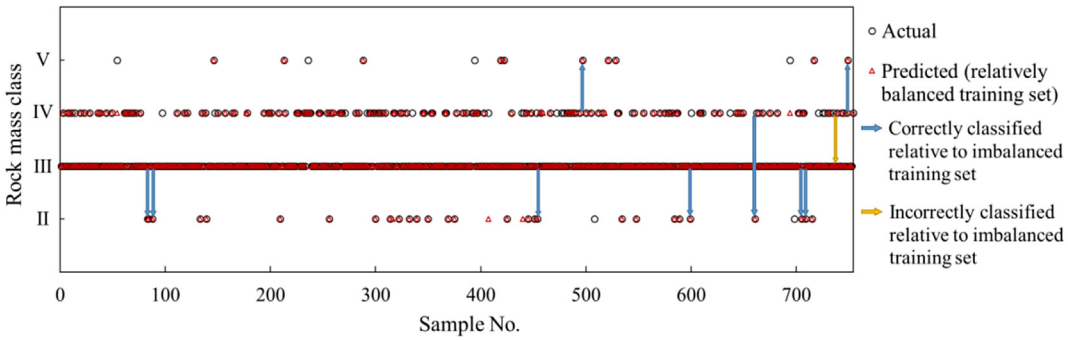

Fig. 18. Prediction results of stacking ensemble classifier on the test set based on learning of the relatively balanced training set.

oversampling, the prediction accuracy of each classifier is improved to a certain extent. Furthermore, the difference in prediction accuracy between the training and test sets is decreased. The results show that a more balanced training set is beneficial for the learning process of classifiers. Generally, the classifier with a relatively balanced training set is prone to have the better fitting effect and prediction performance.

Taking stacking ensemble classifier as an example, the influence of sample imbalance on the prediction performance of samples with different rock mass classes is analysed. Fig. 18 shows the prediction results of the stacking ensemble classifier on the test set based on learning of the relatively balanced training set. It can be seen that after learning the relatively balanced training set, although one sample belonging to class IV is incorrectly classified as class V, the prediction effects for minority class samples (i.e. classes II and V) are improved, and more samples of these two classes are correctly classified than before. More specifically, Fig. 19 shows the evaluation metrics comparison of the stacking ensemble classifier with the imbalanced training set and relatively balanced training set. After the process of SMOTE oversampling for the training set, the *REC* value of class IV is slightly decreased from 0.817 to 0.81, and the *PRC* value of class IV remains unchanged. However, the values of *REC*, *PRC* and F_1 for rock mass classes in other cases are improved to some extent. Especially, the values of *REC* and F_1 for minority class samples are increased significantly, in which the *REC* values of classes II and V are increased from 0.7 to 0.933, and from 0.571 to 0.714, respectively. Also the F_1 values of classes II and V are increased from 0.778 to 0.918, and from 0.727 to 0.833, respectively. The above analysis shows that the relatively balanced training set can effectively improve the prediction performance of the stacking ensemble classifier to a certain extent. To sum up, for the machine models, the more balanced training set is more favorable.

5.2. Limitations

In our study, the stacking technique of ensemble learning is utilised to establish the prediction model for rock mass classification, and the analysis results show that the stacking ensemble classifier has stronger robust and generalisation ability than individual classifiers. Moreover, through the machine learning algorithm, the mapping relationship between rock mass quality and critical operational parameters of TBM is established, which promotes the development of real-time prediction of rock mass classification during the TBM tunnelling process. Therefore, the methods of this study can be used in cases with similar construction conditions and TBM machine parameters. However, there are several limitations in our study, which can be summarised as follows:

(1) The inherent uncertainties of geological condition such as the joint/discontinuity properties, ground characteristics and localised stress states are not considered in the proposed models.
(2) The influence of the cutterhead wear is not considered in our models. The wear of the cutterhead will affect the state of rock breaking to a certain extent, which may influence the values of the TBM operational parameters under different rock mass classes.
(3) The machine learning models are established based on the assumption that the training and prediction samples are independent and identically distributed. However, different projects will have some differences in geology condition, TBM type, construction design requirements, etc., which will limit the applicability of the proposed classifiers to actual projects.

6. Conclusions

This paper presents the real-time prediction of rock mass class based on the stacking ensemble classifier and TBM operation big data. The stacking ensemble classifiers are constructed using SVM, KNN, GBDT and RF as the base classifiers and GBDT as the meta-classifier. Through data processing, 7538 TB M tunnelling cycles are obtained, and the mean value of the selected data without outliers in the stable phase is calculated as the input of classifiers. Based on the tree-based feature selection and removing the highly correlated features, 10 crucial features are selected to predict the rock mass classes. The dataset is divided into the training and test sets in the ratio of 9/1 using simple random sampling. Eight classifiers are established, including SVM, KNN, GBDT, RF, DT, MLP, LR and stacking ensemble classifiers. The grid search method is used to select the optimised hyper-parameters for each classifier. All the classifiers are trained by the training set, and the prediction performance of each classifier is tested by the test set. The comparison between the stacking ensemble classifier and other individual classifiers is analysed. Moreover, the influence of sample imbalance in the training set is briefly discussed. The specific conclusions are drawn as follows:

(1) Compared with the individual classifiers, the proposed stacking ensemble classifier has the better prediction performance, with the values of ACC_{Total}, *Kappa*, PRC_{Total}, REC_{Total}, F_{1_Total}, micro-average and macro-average AUC equal to 93.1%, 0.823, 0.93, 0.931, 0.928, 0.989 and 0.98, respectively. Also, the absolute error of the samples for stacking ensemble classifier are less than 2, among which the absolute error of most samples is 0, and only a few samples have the absolute error of 1. Furthermore, except for the KNN

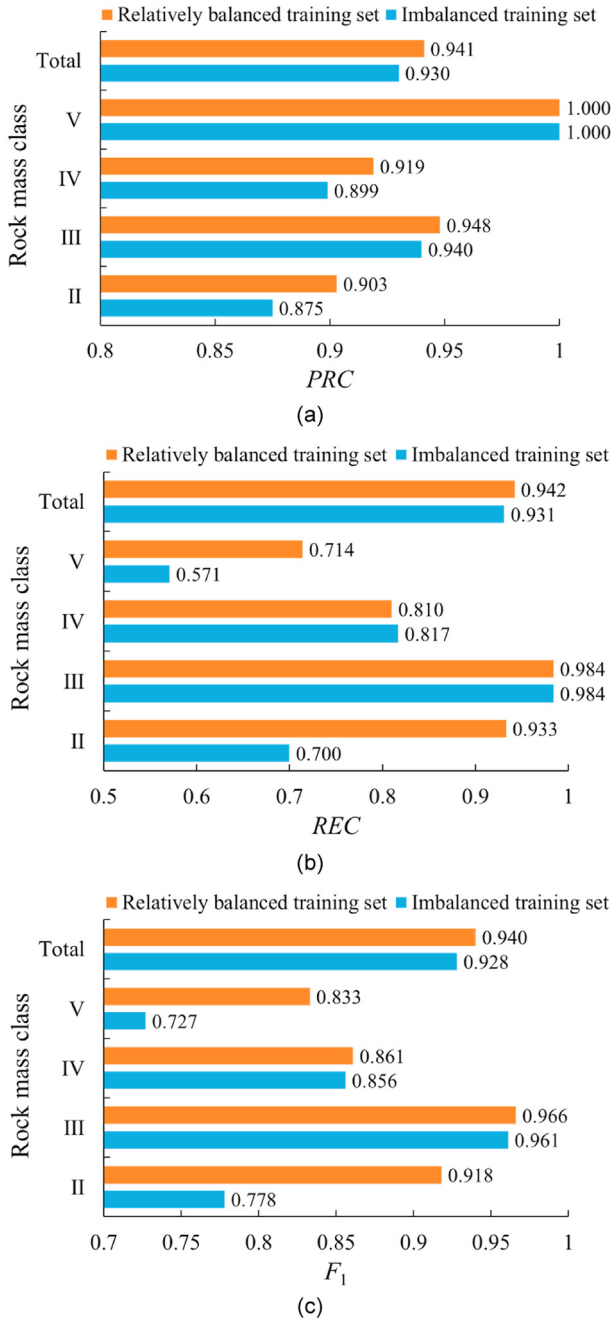

Fig. 19. Evaluation metrics comparison of stacking ensemble classifier with imbalanced training set and relatively balanced training set: (a) *PRC*, (b) *REC*, and (c) F_1.

and LR classifiers, the fitting effect of the other six classifiers can be considered as good.

(2) The stacking technique of ensemble learning can effectively improve the prediction performance of base classifiers on rock mass classification. Especially for the minority class, the prediction effects of stacking ensemble classifier are significantly higher than that of individual classifiers. Therefore, it shows that the ensemble learning model has a more powerful learning and generalisation ability than individual classifiers for small and imbalanced samples.

(3) The *REC* values of different classifiers on the test set positively correlate with the number of training samples. The more the samples in a certain class, the higher the *REC* value on the test set. With the increasing number of samples, the *REC* difference of different classifiers is gradually decreased.

(4) After the SMOTE oversampling for the minority class samples, the overall prediction effects of each classifier are improved to a certain extent. The difference in prediction accuracy between the training and test sets for each classifier is decreased. Taking stacking ensemble classifier as an example, after the SMOTE oversampling for the training set, although the *REC* value of class IV is slightly decreased from 0.817 to 0.81, the values of *REC* and F_1 for minority class samples are significantly increased. The results show that the classifier with a relatively balanced training set is prone to have the better-fitting effect and prediction performance.

To sum up, the proposed stacking ensemble classifier can be well used for the real-time prediction of rock mass classification. However, the sample imbalance is an existing problem, limiting the prediction effects on the minority class samples. Also, the influence of geological condition changes and the wear of TBM cutters are not fully considered. These unsolved problems and the model transfer learning methods will be further studied in our future work.

Declaration of competing interest

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.

Acknowledgments

This study was funded by the National Natural Science Foundation of China (Grant No. 41941019) and the State Key Laboratory of Hydroscience and Engineering (Grant No. 2019-KY-03). Additionally, we sincerely thank the data support from the National Program on Key Basic Research Project of China (973 Program) (Grant No. 2015CB058100), China Railway Engineering Equipment Group Corporation and the Survey and Design Institute of Water Conservancy of Jilin Province, China.

References

Alimoradi, A., Moradzadeh, A., Naderi, R., Salehi, M.Z., Etemadi, A., 2008. Prediction of geological hazardous zones in front of a tunnel face using TSP-203 and artificial neural networks. Tunn. Undergr. Space Technol. 23 (6), 711–717.
Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. 46 (3), 175–185.
Barton, N., 2002. Some new Q-value correlations to assist in site characterization and tunnel design. Int. J. Rock Mech. Min. Sci. 39 (2), 185–216.
Barton, N., Lien, R., Lunde, J., 1974. Engineering classification of rock masses for the design of tunnel support. Rock Mech 6 (4), 189–236.
Bieniawski, Z.T., 1973. Engineering classification of jointed rock masses. Civ. Eng. S. Afr. 15 (12), 335–343.
Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30 (7), 1145–1159.
Breiman L. Random forests. Mach. Learn. 45 (1), 5–32.
Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.
Brun, A.L., Britto, A.S., Oliveira, L.S., Enembreck, F., Sabourin, R., 2018. A framework for dynamic classifier selection oriented by the classification problem difficulty. Pattern Recogn 76, 175–190.
Carter, N.J., Schwertman, N.C., Kiser, T.L., 2009. A comparison of two boxplot methods for detecting univariate outliers which adjust for sample size and asymmetry. Stat. Methodol. 6 (6), 604–621.
Cawley, G.C., Talbot, N.L., 2010. On over-fitting in model selection and subsequent selection bias in performance evaluation. J. Mach. Learn. Res. 11 (1), 2079–2107.
Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.
Chen, Z.Y., Zhang, Y.P., Li, J.B., Li, X., Jing, L.J., 2021. Diagnosing tunnel collapse sections based on TBM tunneling big data and deep learning: a case study on the Yinsong project, China. Tunn. Undergr. Space Technol. 108, 103700.
Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 20 (1), 37–46.

Díez-Pastor, J.F., Rodríguez, J.J., García-Osorio, C., Kuncheva, L.I., 2015. Random balance: ensembles of variable priors classifiers for imbalanced data. Knowledge-Based Syst. 85, 96–111.
Dong, L.J., Li, X.B., Peng, K., 2013. Prediction of rockburst classification using Random Forest. Trans. Nonferrous Met. Soc. China 23 (2), 472–477.
Durgesh, K.S., Lekha, B., 2010. Data classification using support vector machine. J. Theor. App. Inform. Technol. 12(1), 1–7.
Elrahman, S.M.A., Abraham, A., 2013. A review of class imbalance problem. J. Netw. Innov. Comput. 1, 332–340.
Fan, Q., Wang, Z., Li, D., Gao, D., Zha, H., 2017. Entropy-based fuzzy support vector machine for imbalanced datasets. Knowledge-Based Syst. 115, 87–99.
Feng, D.C., Liu, Z.T., Wang, X.D., Jiang, Z.M., Liang, S.X., 2020. Failure mode classification and bearing capacity prediction for reinforced concrete columns based on ensemble machine learning algorithm. Adv. Eng. Inform. 45, 101126.
Feurer, M., Hutter, F., 2019. Hyperparameter optimization. In: Automated Machine Learning. Springer, pp. 3–33.
Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 29 (5), 1189–1232.
Friedman, J.H., 2002. Stochastic gradient boosting. Comput. Stat. Data Anal. 38 (4), 367–378.
Ganganwar, V., 2012. An overview of classification algorithms for imbalanced datasets. Int. J. Emerg. Technol. Adv. Eng. 2 (4), 42–47.
GB/T50218–2014, 2014. Standard for Engineering Classification of Rock Masses. China Planning Press, Beijing, China (in Chinese).
GB50487-2008, 2008. Code for Engineering Geological Investigation of Water Resources and Hydropower. China Planning Press, Beijing, China (in Chinese).
Gholami, R., Rasouli, V., Alimoradi, A., 2013. Improved RMR rock mass classification using artificial intelligence algorithms. Rock Mech. Rock Eng. 46 (5), 1199–1209.
Gong, Q.M., Yin, L.J., Ma, H.S., Zhao, J., 2016. TBM tunnelling under adverse geological conditions: an overview. Tunn. Undergr. Space Technol. 57, 4–17.
Hamidi, J.K., Shahriar, K., Rezai, B., Rostami, J., Bejari, H., 2010. Risk assessment based selection of rock TBM for adverse geological conditions using Fuzzy-AHP. Bull. Eng. Geol. Environ. 69 (4), 523–532.
Hasanpour, R., Schmitt, J., Ozcelik, Y., Rostami, J., 2017. Examining the effect of adverse geological conditions on jamming of a single shielded TBM in Uluabat tunnel using numerical modeling. J. Rock Mech. Geotech. Eng. 9 (6), 1112–1122.
Hassanpour, J., Rostami, J., Zhao, J., 2011. A new hard rock TBM performance prediction model for project planning. Tunn. Undergr. Space Technol. 26 (5), 595–603.
Hoek, E., 1994. Strength of rock and rock masses. ISRM News J 2 (2), 4–16.
Huang, R., Huang, J., Ju, N., Li, Y., 2013. Automated tunnel rock classification using rock engineering systems. Eng. Geol. 156, 20–27.
Imandoust, S.B., Bolandraftar, M., 2013. Application of k-nearest neighbor (KNN) approach for predicting economic events: theoretical background. Int. J. Eng. Res. Appl. 3 (5), 605–610.
Jalalifar, H., Mojedifar, S., Sahebi, A.A., 2014. Prediction of rock mass rating using fuzzy logic and multi-variable RMR regression model. Int. J. Min. Sci. Technol. 24 (2), 237–244.
Jing, L.J., Li, J.B., Yang, C., Chen, S., Zhang, N., Peng, X.X., 2019. A case study of TBM performance prediction using field tunnelling tests in limestone strata. Tunn. Undergr. Space Technol. 83, 364–372.
Jung, J.H., Chung, H.Y., Kwon, Y.S., Lee, I.M., 2019. An ANN to predict ground condition ahead of tunnel face using TBM operational data. KSCE J. Civ. Eng. 23 (7), 3200–3206.
Kardani, N., Zhou, A., Nazem, M., Shen, S.L., 2020. Improved prediction of slope stability using a hybrid stacking ensemble method based on finite element analysis and field data. J. Rock Mech. Geotech. Eng. 13 (1), 188–201.
Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), San Francisco, USA, pp. 1137–1145.
Kuhn, M., Johnson, K., 2013. Classification trees and rule-based models. In: Applied Predictive Modeling. Springer, New York, USA, pp. 369–413.
Kumar, V., Minz, S., 2014. Feature selection: a literature review. Smart Comput. Rev. 4 (3), 211–229.
Landis, J., Koch, G., 1977. The measurement of observer agreement for categorical data. Biometrics 33 (1), 159–174.
Li, S., Liu, B., Xu, X., Nie, L., Liu, Z., Song, J., Fan, K., 2017. An overview of ahead geological prospecting in tunneling. Tunn. Undergr. Space Technol. 63, 69–94.
Li, C.Y., Hou, S.K., Liu, Y.R., Qin, P.X., Jin, F., Yang, Q., 2020. Analysis on the crown convergence deformation of surrounding rock for double-shield TBM tunnel based on advance borehole monitoring and inversion analysis. Tunn. Undergr. Space Technol. 103, 103513.
Liao, Y., Vemuri, V.R., 2002. Use of k-nearest neighbor classifier for intrusion detection. Comput. Secur. 21 (5), 439–448.
Liu, Y.R., Hou, S.K., 2019. Rockburst prediction based on particle swarm optimization and machine learning algorithm. In: Proceedings of the 3rd International Conference on Information Technology in Geo-Engineering (ICITG). Springer, pp. 292–303.
Liu, Y.R., Hou, S.K., Li, C.Y., Zhou, H.W., Jin, F., Qin, P.X., Yang, Q., 2020a. Study on support time in double-shield TBM tunnel based on self-compacting concrete backfilling material. Tunn. Undergr. Space Technol. 96, 103212.
Liu, Q.S., Wang, X.Y., Huang, X., Yin, X., 2020b. Prediction model of rock mass class using classification and regression tree integrated AdaBoost algorithm based on TBM driving data. Tunn. Undergr. Space Technol. 106, 103595.
Liu, B., Wang, R., Zhao, G., Guo, X., Wang, Y., Li, J., Wang, S., 2020c. Prediction of rock mass parameters in the TBM tunnel based on BP neural network integrated simulated annealing algorithm. Tunn. Undergr. Space Technol. 95, 103103.
Liu, Z.B., Li, L., Fang, X.L., Qi, W.B., Shen, J.M., Zhou, H.Y., Zhang, Y.L., 2021. Hard-rock tunnel lithology prediction with TBM construction big data using a global-attention-mechanism-based LSTM network. Autom. Constr. 125, 103647.
Luque, A., Carrasco, A., Martín, A., de las Heras, A., 2019. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognit 91, 216–231.
Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A., Brown, S.D., 2004. An introduction to decision tree modeling. J. Chemom. 18 (6), 275–285.
Panda, M., 2017. Elephant search optimization combined with deep neural network for microarray data analysis. J. King Saud Univ. – Computer Inf. Sci. 32, 940–948.
Polikar, R., 2012. Ensemble learning. In: Ensemble Machine Learning. Springer, pp. 1–34.
Potdar, K., Pardawala, T.S., Pai, C.D., 2017. A comparative study of categorical variable encoding techniques for neural network classifiers. Int. J. Comput. Appl. Technol. 175 (4), 7–9.
Sainin, M.S., Alfred, R., Adnan, F., Ahmad, F., 2017. Combining sampling and ensemble classifier for multiclass imbalance data learning. In: Proceedings of International Conference on Computational Science and Technology. Springer, Singapore, pp. 262–272.
Salimi, A., Rostami, J., Moormann, C., 2017. Evaluating the suitability of existing rock mass classification systems for TBM performance prediction by using a regression tree. Procedia Eng 191, 299–309.
Salimi, A., Rostami, J., Moormann, C., Hassanpour, J., 2018. Examining feasibility of developing a rock mass classification for hard rock TBM application using nonlinear regression, regression tree and generic programming. Geotech. Geol. Eng. 36 (2), 1145–1159.
Salunkhe, U.R., Mali, S.N., 2016. Classifier ensemble design for imbalanced data classification: a hybrid approach. Procedia Comput. Sci. 85, 725–732.
Santos, A.E.M., Lana, M.S., Pereira, T.M., 2021. Rock mass classification by multivariate statistical techniques and artificial intelligence. Geotech. Geol. Eng. 39 (3), 2409–2430.
Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., Wang, Z., 2007. A novel feature selection algorithm for text categorization. Expert Syst. Appl. 33 (1), 1–5.
Shi, S.S., Li, S.C., Li, L.P., Zhou, Z.Q., Wang, J., 2014. Advance optimized classification and application of surrounding rock based on fuzzy analytic hierarchy process and Tunnel Seismic Prediction. Autom. Constr. 37, 217–222.
Srivastava, D.K., Bhambhu, L., 2010. Data classification using support vector machine. J. Theoret. Appl. Inf. Technol. 12 (1), 243–248.
Sun, W., Trevor, B., 2018. A stacking ensemble learning framework for annual river ice breakup dates. J. Hydrol. 561, 636–650.
Sun, S., Wang, S., Wei, Y., 2020. A new ensemble deep learning approach for exchange rates forecasting and trading. Adv. Eng. Inform. 46, 101160.
Vapnik, V., 2000. The Nature of Statistical Learning Theory. Springer, New York.
Viloria, A., Lezama, O.B.P., Mercado-Caruzo, N., 2020. Unbalanced data processing using oversampling: machine Learning. Procedia Comput. Sci. 175, 108–113.
Wang, S., Li, J., Wang, Y., Li, Y., 2016. Radar HRRP target recognition based on gradient boosting decision tree. In: International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Datong, China, pp. 1013–1017.
Wang, C., Gong, G.F., Yang, H.Y., Zhou, J.J., Zhang, Y.K., 2018. NSVR based predictive analysis of cutterhead torque for hard rock TBM. J. Zhejiang Uni. (Eng. Sci.) 52 (3), 479–486 (in Chinese).
Wei, R.M., Wang, J.Y., Jia, W., 2018. MultiROC: Calculating and Visualizing ROC and PR Curves across Multi-Class Classifications. R Package Version 1.1.1. R Core Development Team, Vienna. https://cran.r-project.org/web/packages/multiROC/index.html.
Wistuba, M., Schilling, N., Schmidt-Thieme, L., 2015. Learning hyperparameter optimization initializations. In: International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, pp. 1–10.
Wolpert, D.H., 1992. Stacked generalization. Neural Netw 5 (2), 241–259.
Yagiz, S., 2006. A model for the prediction of tunnel boring machine performance. In: Proceedings of 10th IAEG Congress, Nothingam, UK, pp. 1–10.
Yang, H.Q., Wang, H., Zhou, X.P., 2016. Analysis on the rock–cutter interaction mechanism during the TBM tunneling process. Rock Mech. Rock Eng. 49 (3), 1073–1090.
Zhang, Q., Liu, Z., Tan, J., 2019. Prediction of geological conditions for a tunnel boring machine using big operational data. Autom. Constr. 100, 73–83.
Zhao, J.H., Shi, M.L., Hu, G., Song, X.G., Zhang, C., Tao, D.C., Wu, W., 2019. A data-driven framework for tunnel geological-type prediction based on TBM operating data. IEEE Access 7, 66703–66713.
Zheng, Y.L., Zhang, Q.B., Zhao, J., 2016. Challenges and opportunities of using tunnel boring machines in mining. Tunn. Undergr. Space Technol. 57, 287–299.
Zheng, S., Jiang, A.N., Yang, X.R., Luo, G.C., 2020. A new reliability rock mass classification method based on least squares support vector machine optimized by bacterial foraging optimization algorithm. Adv. Civ. Eng. 1–13, 2020.
Zhou, J., Li, X.B., Mitri, H.S., 2015. Comparative performance of six supervised learning methods for the development of models of hard rock pillar stability prediction. Nat. Hazards 79 (1), 291–316.

Zhou, J., Li, X.B., Mitri, H.S., 2016. Classification of rockburst in underground projects: comparison of ten supervised learning methods. J. Comput. Civil. Eng. 30 (5), 04016003.
Zhou, J., Shi, X., Du, K., Qiu, X., Li, X., Mitri, H.S., 2017. Feasibility of random-forest approach for prediction of ground settlements induced by the construction of a shield-driven tunnel. Int. J. Geomech. 17 (6), 04016129.
Zhou, J., Li, E.M., Yang, S., Wang, M.Z., Shi, X.X., Yao, S., Mitri, H.S., 2019. Slope stability prediction for circular mode failure using gradient boosting machine approach based on an updated database of case histories. Saf. Sci. 118, 505–518.
Zhou, J., Qiu, Y.G., Armaghani, D.J., Zhang, W.G., Li, C.Q., Zhu, S.L., Tarinejad, R., 2021a. Predicting TBM penetration rate in hard rock condition: a comparative study among six XGB-based metaheuristic techniques. Geosci. Front. 12 (3), 101091.
Zhou, J., Qiu, Y.G., Zhu, S.L., Armaghani, D.J., Li, C.Q., Nguyen, H., Yagiz, S., 2021b. Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate. Eng. Appl. Artif. Intell. 97, 104015.

Yaoru Liu obtained his BSc and PhD degrees in Tsinghua University, China, in 1998 and 2004, respectively. He has been working in Department of Hydraulic Engineering, Tsinghua University since 2005. His research interests include (1) long-term stability analysis and dynamic failure of rock mass structures, and (2) application of information technology in safety evaluation of rock engineering (dam foundation, slope and tunnel). He has been participated in a large number of projects funded by National Natural Science Foundation of China and PowerChina.