



Research Paper

Multi-fidelity knowledge inheritance with active querying for data-driven clogging prediction during mechanized tunneling

Xiao Yuan^a, Shuying Wang^{a,b,*}, Tongming Qu^c, Huanhuan Feng^{a,d},
Pengfei Liu^e, Junhao Zeng^a

^a School of Civil Engineering, Central South University, Changsha 410075, China

^b College of Civil and Transportation Engineering, Shenzhen University, Shenzhen 518000, China

^c Institute of Water Engineering Sciences, Wuhan University, Wuhan 430072, China

^d China Railway Tunnel Group Co., Ltd., Guangzhou 511458, China

^e CCCC Second Harbor Engineering Co., Ltd., Wuhan 430012, China

Received 2 December 2024; received in revised form 1 April 2025; accepted 9 April 2025

Available online 26 August 2025

Abstract

Muck clogging during shield tunneling often leads to reduced construction efficiency, increased costs and potential safety hazards. Traditional methods for predicting muck clogging primarily rely on the operator's experience and conventional risk maps, but have limitations in dealing with complex construction conditions. To address these issues, this study presents a Monte-Carlo dropout (MCD)-assisted multi-fidelity neural network (MFNN) framework for effective prediction of muck clogging risk. First, a low-fidelity model is trained based on synthesized data using clogging risk maps. Subsequently, in-situ tunneling data are used as high-fidelity data to train multi-fidelity models. MCD serves to evaluate the uncertainty of the MFNN's inference, combined with an active learning strategy to refine the low-fidelity model via iterative training of the high-fidelity model. Experimental results show that the MCD-assisted MFNN framework captures clogging features more effectively than traditional machine learning models that use only single-fidelity data, especially in scenarios with imbalanced data. This study provides a viable solution for complex problems in shield tunneling by fully utilizing both experiential knowledge accumulated in engineering practice and field monitoring data, demonstrating the potential of integrating knowledge and data in tackling some challenges that were previously unresolved.

Keywords: Machine learning; Muck clogging; Monte-Carlo dropout; Active learning; Multi-fidelity neural networks

1 Introduction

Shield tunneling technology, as an advanced tunneling construction method, has played a crucial role in urban infrastructure development in recent years (Chen et al., 2023; Jiang et al., 2023). By conducting full-face excavation, shield tunneling can efficiently undertake underground construction in various hydrogeological

conditions (S. Wang et al., 2024a), reducing the impact on the surface environment while improving construction efficiency and safety (Zeng et al., 2025). A common and significant problem during shield tunneling is the formation of “muck clogging” (Wang et al., 2020). Muck clogging refers to the accumulation of hardened soil in front of the cutterhead during tunneling. This phenomenon can increase thrust and torque while reducing construction efficiency. In severe cases, it can block the cutterhead, accelerate cutter wear, and force the shield machine to halt for specialized maintenance and clog removal.

The origin of muck clogging is rooted in the complex interplay of various factors. First, differences in ground

* Corresponding author at: School of Civil Engineering, Central South University, Changsha 410075, China.

E-mail address: swang24@szu.edu.cn (S. Wang).

Peer review under the responsibility of Tongji University

conditions, such as soil cohesion, particle size, moisture content, and mineral composition, directly influence clogging formation. Second, the thrust applied during tunneling also affects the characteristics and formation of muck clogging. Due to these variations, the occurrence and severity of muck clogging can differ significantly. Current research on muck clogging mainly relies on experimental methods (S. Wang et al., 2024c). However, these experimental methods have limitations. While laboratory studies can effectively reveal the characteristics of muck clogging and provide clogging countermeasures, the complexity of clogging formation in actual tunneling conditions often makes it difficult to fully guide real-time prediction in the field (S. Wang et al., 2024c). Furthermore, laboratory tests are time-consuming and costly, and results are often delayed compared to project schedules. Quick response and immediate feedback are critical during construction, and the lag in experimental results may hinder timely prediction in practical situations (Wang et al., 2022).

In practice, traditional methods for predicting muck clogging rely on two main approaches. The first method depends on the operator's experience. Operators typically estimate clogging formation based on increased thrust and decreased tunneling speed. When the machine becomes noticeably harder to advance, it often indicates that muck clogging has already formed and hardened in front of the cutterhead. Unfortunately, by the time this is detected, it is often too late, and the operator must halt the machine to remove the clogging. The second method involves engineers using a risk map (Fig. 1) to make preliminary judgments (Thewes & Hollmann, 2016). This risk map is based on soil liquid limit, plastic limit, and moisture content, and its limitations are also evident, as it cannot fully capture the complexity of clogging formation, leading to inaccurate predictions in practice.

Given the limited effectiveness of traditional methods, machine learning has been increasingly resorted to address complex engineering challenges (Li et al., 2023; D. Zhang et al., 2023). Currently, machine learning is applied in various aspects of shield tunneling, including thrust prediction (Kong et al., 2022; W. Zhang et al., 2023), cutterhead torque prediction (Cachim & Bezuijen, 2019; X. Huang et al., 2022), soil conditioning agent prediction (S. Wang et al., 2024b), attitude prediction during tunneling (H. Huang et al., 2022; L. Wang et al., 2024), tunneling speed prediction (Xu et al., 2023; Zendaki et al., 2024), surface deformation prediction (Chen et al., 2019), and tunnel defect prediction (Basnet et al., 2023; Li et al., 2022).

Although machine learning has demonstrated its potential in shield tunneling, its data-driven nature also presents challenges. Machine learning model performance often depends on data quantity and quality. High-quality, diverse data are necessary to ensure model's effectiveness and generalization (Tran et al., 2022). In shield tunneling, data typically come from machine sensors, which may be subject to noise during acquisition. Furthermore, the data sets are often imbalanced depending on the issues to be addressed, such as the ratio of clogging to non-clogging cases or tunnel collapse to non-collapse incidents. Models trained on imbalanced data tend to favor common events while performing poorly at predicting rare ones (Hou et al., 2022). The root of this problem lies in the inability of standard loss functions to adequately reflect the importance of rare events in imbalanced data scenarios, causing models to overfit the common classes (Rezvani & Wang, 2023).

In recent years, the fusion of multi-source data in machine learning has gained increasing attention. Techniques such as hierarchical Bayesian modeling (Bozorgzadeh et al., 2019), clustering algorithms

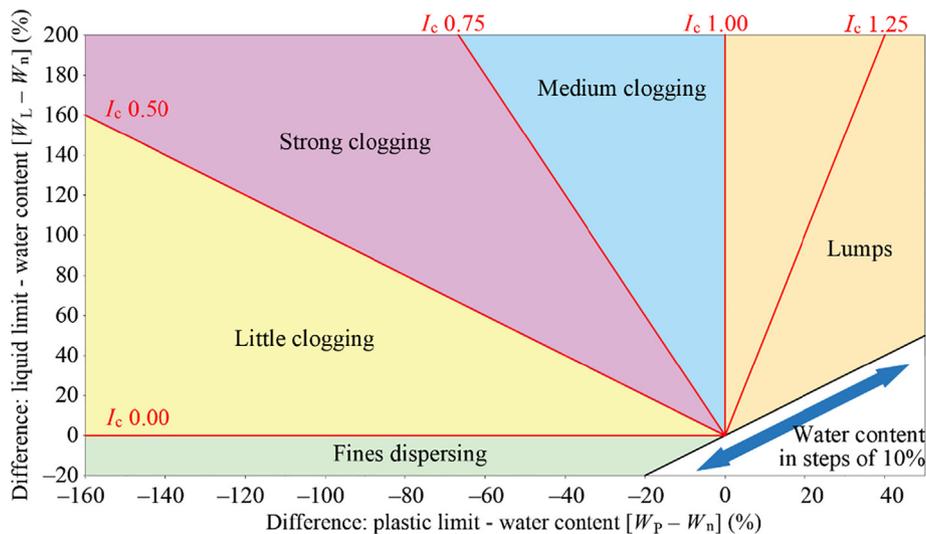


Fig. 1. Muck clogging risk map. (Note: W_L and W_P denote the liquid limit and plastic limits, respectively, and W_n is the natural water content (all in %); the horizontal axis plots $W_P - W_n$, the vertical axis $W_L - W_n$; I_c is the consistency index, defined as $I_c = (W_L - W_n) / (W_P - W_n)$).

(Cai et al., 2024), stacking algorithms (Liu et al., 2024; Luo et al., 2023), and transfer learning (Li et al., 2024; Qu et al., 2023b) can effectively integrate data from different sources, reducing dependence on specific datasets and improving model adaptability and flexibility. In particular, multi-fidelity neural networks (MFNNs) have gained widespread attention in recent years (Su et al., 2023; Zhang et al., 2022). This approach first utilizes a large amount of low-fidelity data to establish a baseline model that captures the basic features of the problem. Subsequently, scarce high-fidelity data are introduced to correct and optimize the model, resulting in more accurate fitting of actual engineering data. This method can significantly enhance the model's ability to recognize rare events when samples are limited.

In current machine learning practices, most learning follows a “passive learning” pipeline, where measurement data are used directly without a rational selection process for training data. This pipeline fails to maximize both data utilization efficiency and the model's performance. Without strategic sample selection, “passive learning” often misses important features in the data. In contrast, active learning uses adaptive sampling strategies (Qu et al., 2023a) to identify and select the most informative samples. This not only improves the efficiency of information acquisition but also significantly enhances the model's predictive performance when dealing with imbalanced data. In this study, a novel strategy that combines MFNN with Monte-Carlo dropout (MCD)-based active learning is proposed to tackle the challenge of predicting mud-cogging issues during shield tunneling.

The structure of this study is organized as follows: Section 2 introduces the principles of MFNN, the MCD uncertainty identification strategy, and the MCD-assisted MFNN framework. Section 3 describes the specific methods for muck clogging prediction, including the construction of low-fidelity data, acquisition of high-fidelity

engineering data, and model training and evaluation strategies for the MFNN. The effectiveness of multi-fidelity fusion is demonstrated through a comparative analysis of model performance. Section 4 discusses the significance of cross-fidelity data fusion in detail, along with a comparative study of two uncertainty identification methods based on MCD and committee models.

2 MCD-MFNN framework

2.1 Multi-fidelity neural network

The MFNN is a machine learning framework that effectively integrates data of different fidelities to improve prediction accuracy and generalization capability. The core idea is to establish a residual relationship between low-fidelity and high-fidelity models, enabling stepwise data fusion and correction to achieve efficient and reliable prediction results. This approach not only reduces the training time and computational cost of the high-fidelity model but also significantly enhances the reliability and accuracy of the results.

As shown in Fig. 2, the network first uses low-fidelity data $\mathbf{D}_L = \{(\mathbf{X}_L, \mathbf{Y}_L)\} = \{[\mathbf{x}_{L,i}, y_L(\mathbf{x}_{L,i})]\}_{i=1}^{N_L}$ to train a low-fidelity model \mathbf{M}_L . Subsequently, high-fidelity data $\mathbf{D}_H = \{(\mathbf{X}_H, \mathbf{Y}_H)\} = \{[\mathbf{x}_{H,j}, y_H(\mathbf{x}_{H,j})]\}_{j=1}^{N_H}$ are inputted into the trained low-fidelity model \mathbf{M}_L to obtain the corresponding low-fidelity predictions $\hat{\mathbf{Y}}_{L,H} = \{\hat{y}_L(\mathbf{x}_{H,j})\}_{j=1}^{N_H}$ for the high-fidelity data. These predictions are treated as additional feature information and combined with the original high-fidelity data \mathbf{X}_H to form new extended input data $\mathbf{X}_{\text{new}} = (\mathbf{X}_H, \hat{\mathbf{Y}}_{L,H})$.

Based on the extended new input data $\mathbf{X}_{\text{new}} = (\mathbf{X}_H, \hat{\mathbf{Y}}_{L,H})$, a high-fidelity residual network is trained using the residual $\mathbf{E}_{H-L,H} = \mathbf{Y}_H - \hat{\mathbf{Y}}_{L,H}$ between

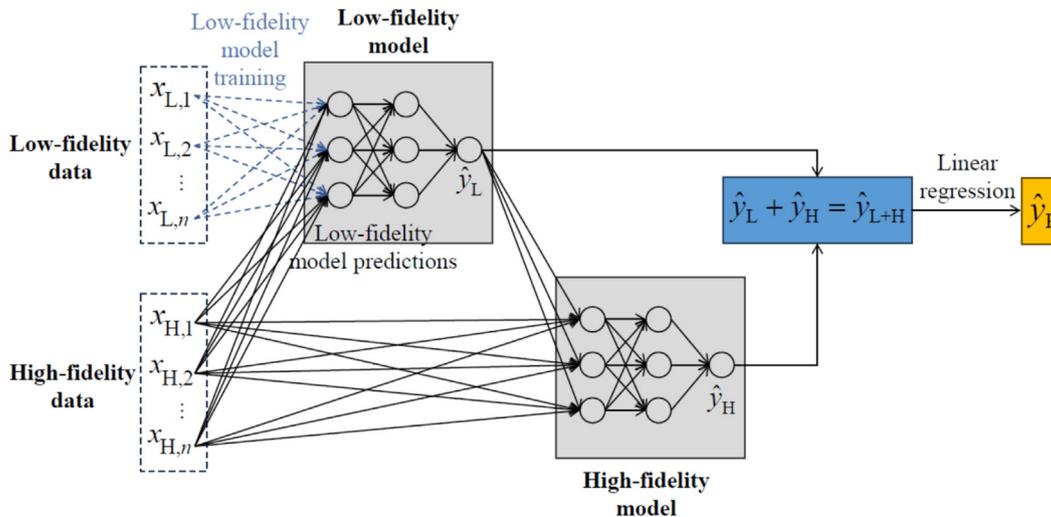


Fig. 2. Schematic diagram of a multi-fidelity residual network. (Note: \hat{y}_L means low-fidelity model output; \hat{y}_H means high-fidelity residual output; \hat{y}_{L+H} means the combined low-fidelity and high-fidelity output; \hat{y}_R means the final combined prediction.)

the original high-fidelity data \mathbf{Y}_H and the low-fidelity predictions $\hat{\mathbf{Y}}_{L,H}$ as labels. This residual component aims to capture the difference between the low-fidelity predictions and the actual high-fidelity data, effectively compensating for the shortcomings of the low-fidelity model. Therefore, the multi-fidelity predictions are obtained by adding the low-fidelity predictions $\hat{\mathbf{Y}}_{L,H}$ and the high-fidelity residual predictions $\hat{\mathbf{E}}_{H-L,H}$:

$$\hat{\mathbf{Y}} = \hat{\mathbf{Y}}_{L,H} + \hat{\mathbf{E}}_{H-L,H}. \quad (1)$$

To further improve prediction accuracy, a linear regression model is used to fit the multi-fidelity predictions and establish their relationship with the actual high-fidelity data:

$$\hat{\mathbf{Y}}_R = \beta_0(\hat{\mathbf{Y}}_{L,H} + \hat{\mathbf{E}}_{H-L,H}) + \beta_1, \quad (2)$$

where $\hat{\mathbf{Y}}_R$ is the final multi-fidelity prediction after linear regression adjustment, β_0 and β_1 are the linear model coefficients.

In summary, the MFNN first uses low-fidelity data to train a low-fidelity model and then uses high-fidelity data to obtain preliminary low-fidelity predictions through the low-fidelity model. These predictions are combined with high-fidelity data to form extended inputs, which are used to construct a residual relationship model to learn the prediction error. Finally, linear fitting is performed to further correct the predictions. This method effectively leverages the low acquisition costs of low-fidelity data and the accuracy of high-fidelity data, maximizing information utilization and making the prediction results more reliable.

2.2 Monte-Carlo dropouts uncertainty identification strategy

The MCD is a technique used to quantify the uncertainty of neural networks (Gal & Ghahramani, 2016). Unlike standard inference, where all neurons in the network remain active, MCD applies dropout even during inference. This means that each time the model makes a prediction, some neurons are randomly turned off with a certain probability, while the remaining neurons continue to function as usual.

Because different neurons are dropped each time, the network behaves slightly differently for the same input, leading to different prediction results. Mathematically, if we perform T separate predictions on the same input \mathbf{X} , we obtain multiple outputs:

$$\hat{y}_t = f(\mathbf{X}; \boldsymbol{\theta}_t), t = 1, 2, \dots, T, \quad (3)$$

where $\boldsymbol{\theta}_t$ represents the network parameters after some neurons are randomly deactivated in the t th prediction. By averaging these predictions, we obtain a more stable estimate of the output:

$$E[y] \approx \frac{1}{T} \sum_{t=1}^T \hat{y}_t. \quad (4)$$

The uncertainty associated with the prediction is quantified using the variance of these outputs:

$$\text{Var}(y) \approx \frac{1}{T} \sum_{t=1}^T \hat{y}_t^2 - \left(\frac{1}{T} \sum_{t=1}^T \hat{y}_t \right)^2. \quad (5)$$

A higher variance indicates greater uncertainty in the model's predictions, suggesting that the model lacks confidence in this particular case. This information can be used in active learning to prioritise selecting the most informative datasets for additional training, ensuring that the model improves its learning on complex or underrepresented cases, ultimately enhancing overall generalization. As illustrated in Fig. 3, the repeated application of dropout generates multiple network configurations, each providing slightly different outputs for the same input. These variations capture the intrinsic uncertainty of the model, facilitating targeted learning and enhancing generalization performance.

2.3 Combined framework

MFNN effectively integrates low-fidelity and high-fidelity data to enhance prediction accuracy, but its training strategy follows the same loss function design as traditional neural networks, which optimizes network parameters (weights and biases) to minimize the mean squared error (MSE) or cross-entropy loss. As a result, in balanced datasets, MFNN improves model performance. However, in imbalanced data scenarios — such as the classification of “muck clogging” versus “non-clogging” cases in this study — the optimization objective remains dominated by the majority class, leading the model to primarily learn majority class patterns while neglecting key minority class information. This bias ultimately limits MFNN's ability to accurately predict rare clogging events.

To address this issue, by integrating the multi-fidelity model introduced in Section 2.1, the MCD-based uncertainty identification strategy described in Section 2.2, and the concept of active learning, this study proposes an MCD-assisted multi-fidelity neural network framework. The core of this framework lies in fully utilizing the complementary characteristics of data with different fidelities, combined with uncertainty information to dynamically select representative samples for training, thereby continuously improving the model training process to achieve higher generalization.

Building upon the multi-fidelity model, this framework introduces an active learning strategy during the training phase of the high-fidelity model and uses MCD to identify the uncertainty of samples in a high-fidelity data pool, achieving iterative optimization of the training process. By prioritizing high-uncertainty samples, this approach mitigates the bias introduced by imbalanced data, ensuring that the model effectively learns from hard-to-learn samples. Specifically, as shown in Fig. 4, the high-fidelity dataset is first divided into an initial training set, an unseen

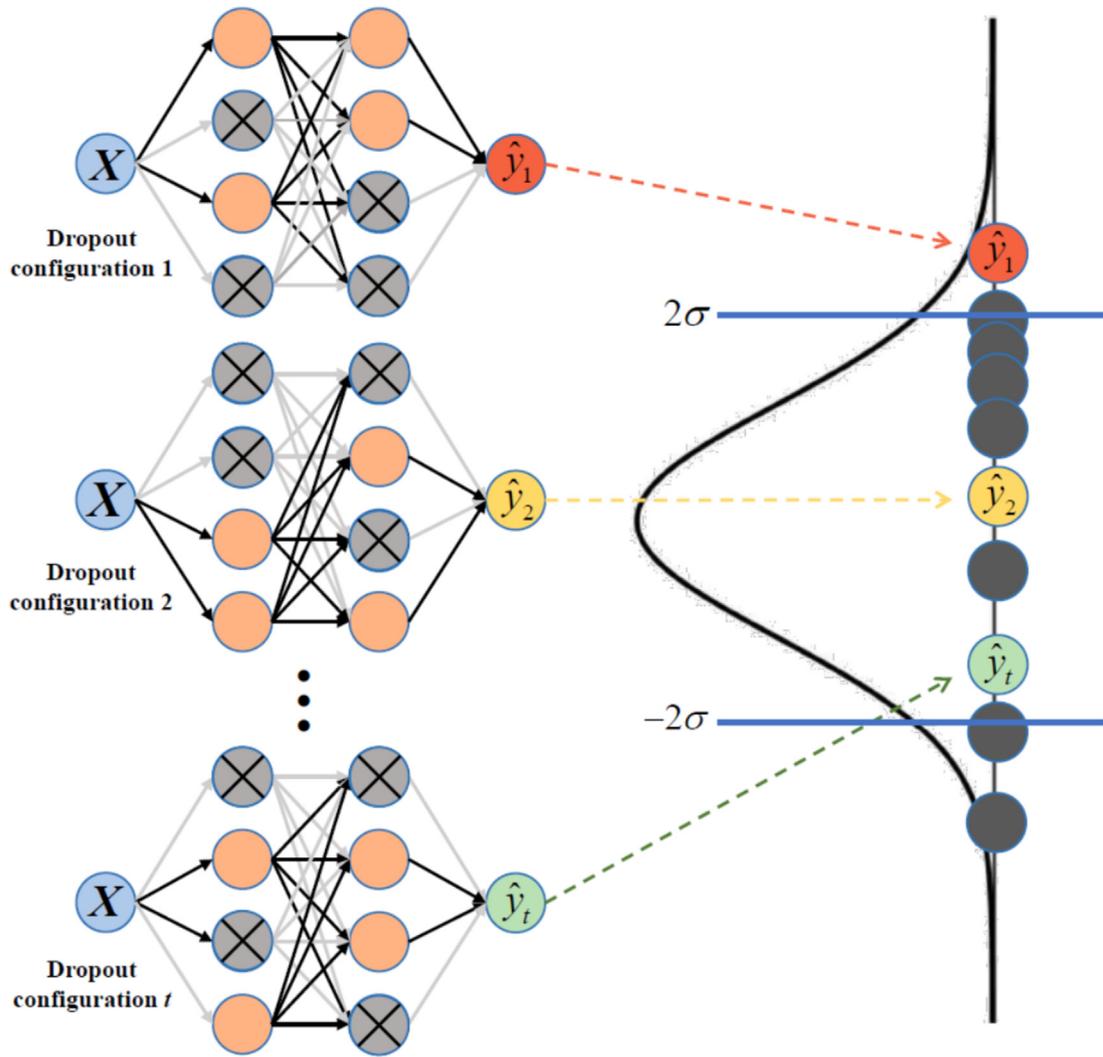


Fig. 3. Schematic of MCD-based uncertainty analysis. (Note: The gray neurons represent deactivated units (due to dropout), while the black neurons in the distribution plot represent the outputs of other dropout configurations; σ is the standard deviation of these predictions.).

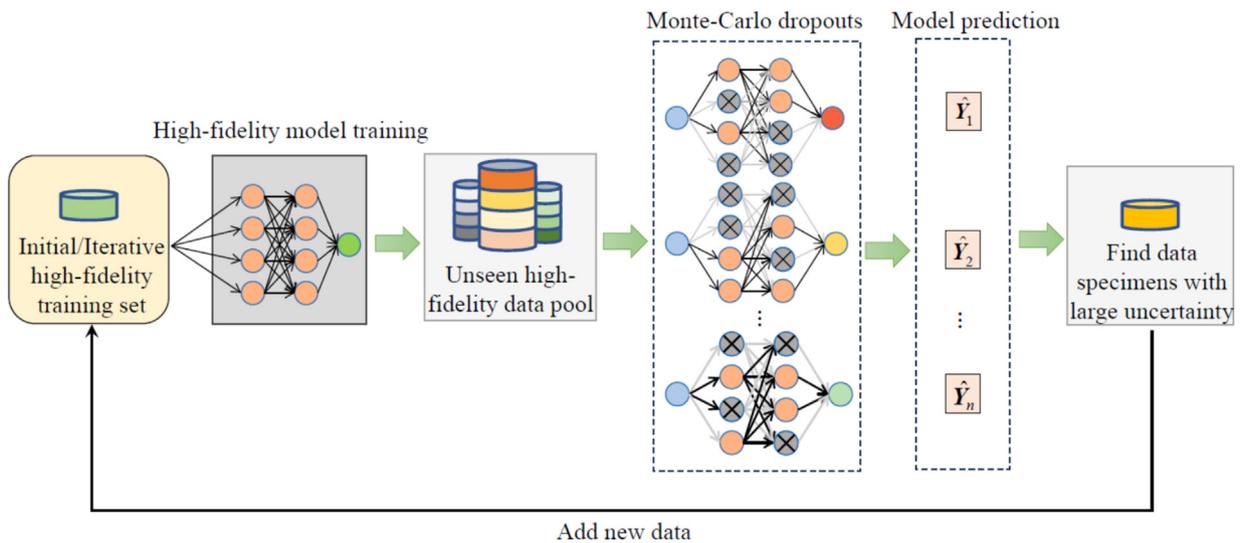


Fig. 4. Schematic diagram of the active learning strategy based on MCD uncertainty identification.

high-fidelity data pool (note that the term “unseen” here refers to being unseen initially; during an iterative process, a sampling with replacement strategy will be applied to add selected samples from the data pool to the training set), and an unseen test set. Initially, a model is pre-trained using the low-fidelity dataset. Subsequently, based on the pre-trained model, MCD is applied to make multiple predictions for each sample in the data pool, generating a set of predicted values. By calculating the variance of these predicted values, the uncertainty of the samples is quantified. A higher variance indicates that the model’s prediction for that sample is more unstable and has a higher level of uncertainty. Based on this evaluation, samples with high uncertainty are iteratively added to the initial training set, which is then updated to form an iterative training set. This allows the model to focus on learning complex, difficult-to-predict samples (Yuan et al., 2024).

This active learning process is repeated iteratively, gradually enhancing the model’s ability to handle uncertain samples, thereby improving its overall generalization performance. Finally, after sufficient training of the high-fidelity model, the outputs from the trained low-fidelity and high-fidelity models are combined, followed by a linear regression fit against the actual results, ultimately forming the complete MCD-assisted MFNN framework. As illustrated in Fig. 5, the basic procedures of the framework are as follows:

Step 1: Train a low-fidelity model M_L using low-fidelity data $D_L = \{(X_L, Y_L)\} = \{\{x_{L,i}, y_L(x_{L,i})\}_{i=1}^{N_L}\}$.

Step 2: Input high-fidelity data $D_H = \{(X_H, Y_H)\} = \{\{x_{H,j}, y_H(x_{H,j})\}_{j=1}^{N_H}\}$ into the trained low-fidelity model M_L to obtain the corresponding low-fidelity predictions $\hat{Y}_{L-H} = \{\hat{y}_L(x_{H,j})\}_{j=1}^{N_H}$ for the high-fidelity data.

Step 3: $X_{new} = (X_H, \hat{Y}_{L-H})$ and $E_{H-L-H} = Y_H - \hat{Y}_{L-H}$ are combined to construct the high-fidelity model training dataset $D_E = \{(X_{new}, E_{H-L-H})\}$.

Step 4: Then, split the dataset D_E into the initial training D_{E-T} and the high-fidelity data pool D_{E-V} in a certain ratio. Train the high-fidelity model using the training set D_{E-T} . Then, use the high-fidelity model to predict the D_{E-V} and apply Monte-Carlo dropouts to assess the uncertainty of the samples in the data pool D_{E-V} .

Step 5: Use a sampling with replacement strategy to extract a certain proportion of highly uncertain samples from the data pool D_{E-V} and add them to the training set, updating the training set D_{E-T} .

Step 6: Repeat Steps 4 and 5 to iteratively train the high-fidelity model M_H multiple times.

Step 7: Finally, input D_E into the trained models M_L and M_H , sum the outputs of each model, and fit the multi-fidelity predictions using a linear regression model.

In this combined framework, the model can efficiently utilize data of different fidelities and uncertainty information to better adapt to complex features, thereby gradually improving prediction accuracy and generalization capability.

3 Muck clogging prediction via the MCD-MFNN framework

Muck clogging is a critical and common issue in shield tunneling, and its accurate prediction is essential to ensure safe and efficient tunneling. However, muck-clogging data are often scarce, posing challenges for traditional machine learning model training, resulting in low prediction accuracy. To address this issue, this study employs the MCD-MFNN framework to integrate multi-source data related

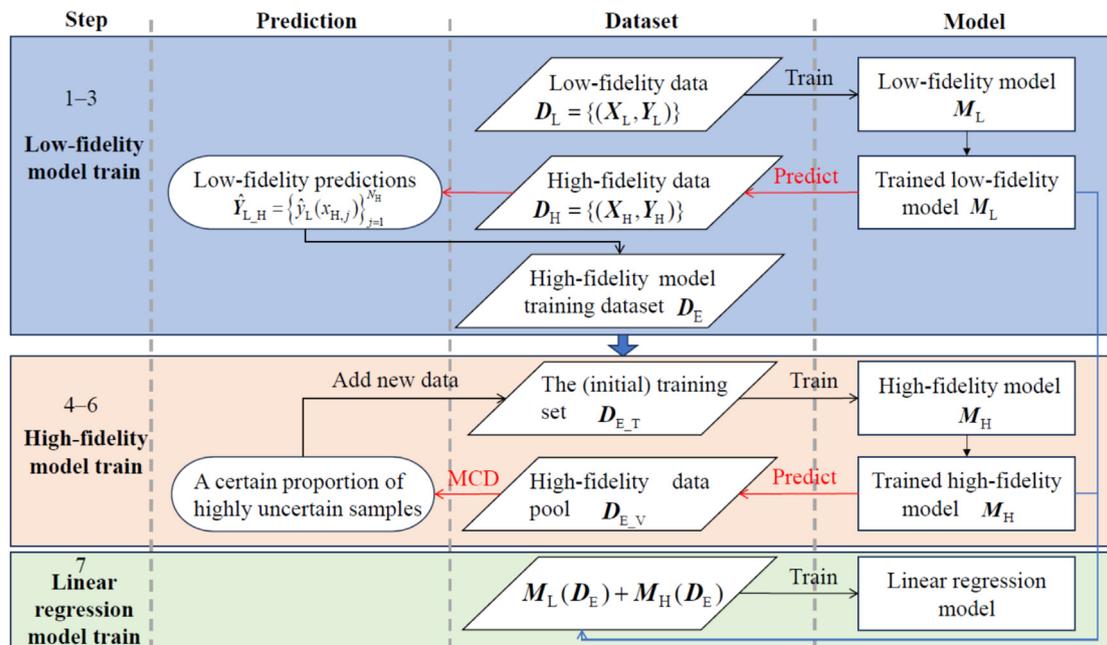


Fig. 5. Schematic illustration of the proposed MCD-assisted MFNN training process (inspired by Zhang et al., 2024b).

to muck clogging prediction, fully leveraging the complementary characteristics of low-fidelity and high-fidelity data to enhance model stability and generalization capability.

3.1 Low-fidelity data

During shield tunneling, researchers have conducted extensive studies, summarizing numerous rules presented in various forms, such as analytical expressions, quantitative descriptions, and diagrams. These research findings contain valuable engineering knowledge that can be used to construct corresponding datasets, thereby generating large amounts of low-fidelity data. Notably, for predicting the formation of muck clogging during tunneling, as shown in Fig. 1, Thewes and Hollmann (2016) developed a muck clogging risk assessment chart based on experiments and field experience, which has been widely adopted in the shield tunneling field as an empirical tool for preliminary risk assessment. Specifically, the chart uses plastic limit-moisture content as the x -axis and liquid limit-moisture content as the y -axis, and categorizes risk regions into multiple levels using the consistency index (I_c). According to the risk map, the risk of muck clogging can be divided into strong clogging, medium clogging, little clogging, fines dispersing, and lumps regions. This risk assessment chart represents an expert-derived knowledge system based on engineering experience, forming the foundation for the low-fidelity dataset. On this basis, a random sampling approach (a method that selects data points randomly within predefined risk regions to ensure balanced coverage of different clogging risk levels) is used to generate low-fidelity samples using the risk map, and the muck clogging label is assigned based on their location, ensuring that the dataset explicitly reflects the prior knowledge embedded in the chart.

However, muck clogging formation is not only influenced by the three geological parameters of liquid limit, plastic limit, and moisture content, but also related to other factors such as void ratio, cutterhead rotation speed, cutterhead torque, total thrust, and penetration rate (Yuan et al., 2024). These parameters provide crucial insights into clogging conditions: when muck accumulates on the cutterhead, maintaining a constant rotation speed requires a significant increase in cutterhead torque, making the relationship between these two parameters an essential indicator of clogging severity. Additionally, total thrust reflects the resistance encountered during excavation, which increases when clogging restricts soil movement, requiring greater force to maintain tunnel advancement. Penetration rate, representing the efficiency of soil cutting, further aids in identifying clogging conditions as reduced excavation efficiency often correlates with excessive adhesion and clogging. Therefore, relying solely on these three geological parameters may not fully capture the actual complexity of clogging formation. Additionally, since the risk map only considers these three geological parameters,

the generated sample labels remain unchanged regardless of the combination of other features.

To construct a set of low-fidelity data that reflects both the knowledge of the risk map, and the features related to muck clogging formation, this study uses kernel density estimation (KDE) to generate data. Mimicking the actual distribution of engineering data, low-fidelity samples are generated with features such as void ratio, cutterhead rotation speed, cutterhead torque, total thrust, and penetration rate. These samples are combined with those generated based on the risk map to construct a more comprehensive set of low-fidelity data, ultimately generating 10 000 samples, of which 4500 are labelled as muck clogging and 5500 as non-clogging. This hybrid approach ensures that the dataset integrates both domain knowledge and statistical representation of real-world tunneling conditions, balancing knowledge-driven insights with data-driven variability.

3.2 High-fidelity engineering data

The in-situ monitoring data from Changchun Metro Line 5 in Jilin, China are utilized as high-fidelity data to validate the proposed MCD-assisted MFNN strategy. This project used a 6.48 m diameter earth pressure balance (EPB) cutterhead with a 40% opening ratio to deal with complex geological conditions. The cutterhead was equipped with various tools, including 4 central twin disc cutters, 32 single-edged disc cutters, 8 edge scrapers, 36 chisels, and 8 gauge cutters.

During shield tunneling, various construction parameters such as thrust and tunneling speed were recorded in real-time. In this case, the full-section shield crossing of the weathered mudstone stratum (rings 420–670) was taken as an example (highlighted in red in Fig. 6), with muck clogging occurring in rings 628–630. The physical statistical parameters for this stratum are presented in Table 1. To predict muck clogging, the study used both tunneling parameters and geological data from this section for model training. Since borehole geological data are discrete, Kriging interpolation was employed to generate a continuous spatial distribution of geological parameters along the tunnel alignment. A spherical variogram model was selected, with the sill set as 0.8 and the nugget effect as 0.1, ensuring a balance between spatial variability and measurement noise. These interpolated values were then matched to their respective tunneling rings based on chainage positions, ensuring that each ring's dataset comprehensively incorporated both geological and excavation parameters. This approach provided a more precise representation of real-world conditions, enhancing the accuracy of the predictive model.

The dataset contained 25 360 samples after excluding machine shutdown and abnormal tunneling phases. Only 1445 samples represented muck-clogging events, while 23 915 were labelled as non-clogging, resulting in a 1:16 data imbalance. This imbalance makes it difficult for traditional

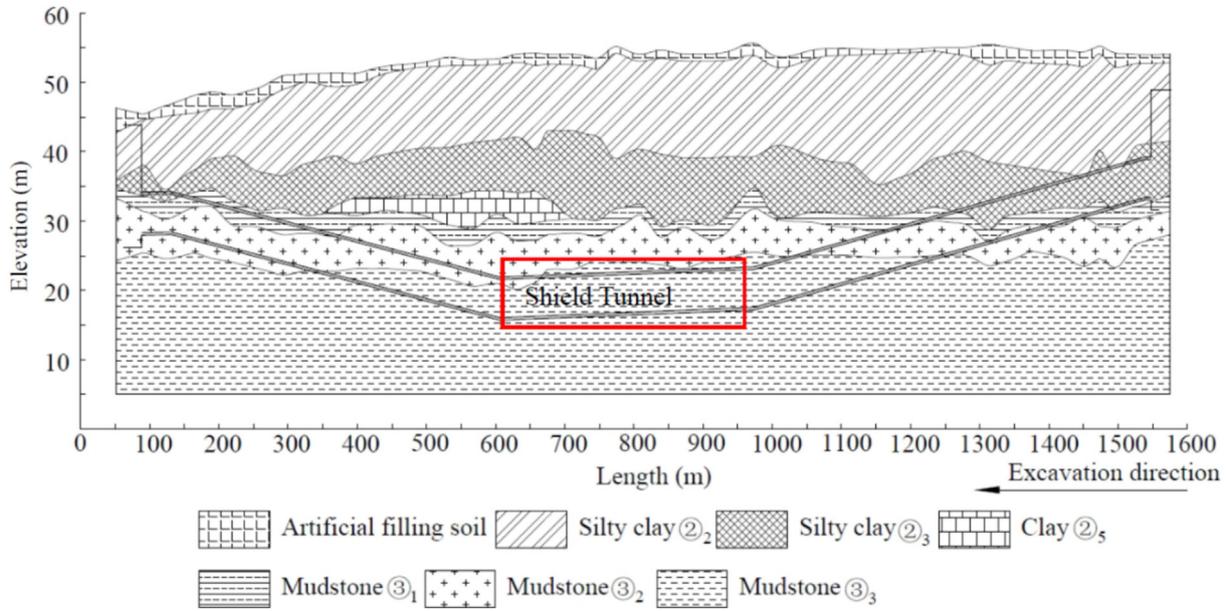


Fig. 6. Geological cross-section of the shield tunnel.

Table 1
Physical parameters of weathered mudstone stratum.

Statistical properties	Water content (%)	Liquid limit (%)	Plastic limit (%)	Void ratio
Maximum value	29.41	43.41	27.12	0.84
Minimum value	23.13	37.81	21.61	0.70
Average value	25.27	39.22	22.66	0.75

machine learning models to effectively detect clogging events.

3.3 Model training

Muck clogging risk prediction is essentially a classification problem, where the final layer of the neural network uses a Softmax activation function to convert the network output into a class probability distribution to obtain the final class label. Therefore, if the Softmax activation function is removed and the output remains as a probability value, the classification problem can be transformed into a regression problem with a single output node producing values between 0 and 1. Values above 0.5 indicate muck clogging, while a value of 0.5 or below indicates normal operation. Based on this, the MCD-assisted multi-fidelity neural network proposed in Section 2.3 is adopted in this section for muck clogging risk prediction. The trained model is expected to produce an output close to 1 for muck-clogging samples and near 0 for non-clogging samples.

During model training, the optimal hyperparameter combination for the low-fidelity model is first determined using a trial-and-error method, as shown in Table 2. The low-fidelity model is fully trained using the low-fidelity

dataset constructed in Section 3.1 to ensure that it captures the main features of the low-fidelity data. The high-fidelity dataset is divided into an initial training set, a high-fidelity data pool for validation and re-sampling, and a testing set in a ratio of 1:1:8. The higher proportion of the test set is to evaluate the model's generalization ability when facing data imbalance and limited training set size. To further enhance the stability and reliability of the model, a five-fold cross-validation (5-fold CV) strategy is employed. Specifically, in each fold, four-fifths of the dataset are assigned as the testing set, while the remaining one-fifth is further divided into an initial training set and a high-fidelity data pool, maintaining the original 1:1 ratio between them. The final model output is obtained by averaging the results from five independently trained models, each derived from a different fold. This ensemble approach mitigates potential biases caused by specific training configurations and increases the robustness and generalization of the predictions.

In MCD-based uncertainty quantification, the dropout probability is a key hyperparameter that influences the trade-off between stability and uncertainty representation. In this study, the dropout probability was tested within a range of 0.1–0.5, and 0.2 was selected as the optimal value. A lower dropout probability (e.g., 0.1) resulted in insuffi-

Table 2
Hyperparameter configuration for low-fidelity and high-fidelity models.

Parameter	Range	Value	
		Low-fidelity model	High-fidelity model
Model	MLP	MLP	MLP
Number of neurons in the input layer	Integer [6, 12]	8	9
Number of neurons in the output layer	Categorical {1, 2}	1	1
Number of hidden layers	Integer [1, 5]	2	3
Regularization	Categorical {None, L1, L2, L1 + L2}	L2	L2
Loss function	Categorical {MSE, MAE}	MSE	MSE
Epoch	Integer [20, 200]	50	50
Optimizer	Categorical {SGD, Adam, RMSprop}	Adam	Adam
Activation	Categorical {ReLU, Tanh, Sigmoid}	ReLU	ReLU
Learning rate	Continuous [0.0001, 0.1]	0.001	0.01

Note: {} represents a discrete set of predefined values from which the model selects one; [] denotes an interval within which a parameter is selected. If the parameter is an integer, the selection is limited to integer values within the range.

cient uncertainty representation, while a higher probability (e.g., 0.5) led to excessive information loss. The final selection of 0.2 ensured a balance between model stability and effective uncertainty quantification.

Similarly, the optimal hyperparameter combination for the high-fidelity residual network is determined using a trial-and-error method, as shown in Table 2. For the final stage of the model, the linear regression model uses the least square method to solve for regression coefficients and intercepts, with the hyperparameter configuration shown in Table 3.

Figure 7 shows the training loss curves of the high-fidelity residual model across different training epochs, comparing the use of MCD with the baseline model without MCD. From the figure, it can be observed that the baseline model without MCD exhibits higher loss values. In contrast, the model employing the MCD strategy shows significantly reduced loss values, and as the number of training iterations increases, there is a trend of further loss reduction. This indicates that during model training, the iterative application of the MCD strategy enables deeper learning of the samples, thereby gradually enhancing the model's predictive capability.

3.4 Model evaluation indicators

The choice of model evaluation indicators is crucial for comprehensively understanding the performance of the MCD-MFNN framework. Since multiple regression models are involved in this framework (e.g., low-fidelity model, high-fidelity residual model, and linear regression model),

Table 3
Hyperparameter configuration for the linear regression model.

Parameter	Value	
	Without MCD	MCD
Regression coefficient	1.3349	1.0908
Intercept	-0.0357	-0.0109

various regression evaluation metrics are required for performance analysis. First, mean absolute error (MAE) and the coefficient of determination (R^2) are used as the main regression evaluation metrics.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (6)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (7)$$

where n represents the total number of samples; y_i and \hat{y}_i are the true value and predicted value of the i th sample, respectively; \bar{y} is the average of all true values. The closer the R^2 value is to 1, the better the model's fitting performance.

However, relying solely on regression evaluation metrics does not provide a complete assessment of model performance; classification evaluation metrics are also needed to evaluate the model's accuracy in identifying muck clogging. Specifically, Recall is used to measure the model's ability to identify specific categories, combined with the F_1 -score for a comprehensive evaluation:

$$\text{Recall}_{\text{category}} = \frac{\text{TP}_{\text{category}}}{\text{TP}_{\text{category}} + \text{FN}_{\text{category}}}, \quad (8)$$

$$F_1\text{-score} = 2 \times \frac{\text{Precision}_{\text{overall}} \times \text{Recall}_{\text{overall}}}{\text{Precision}_{\text{overall}} + \text{Recall}_{\text{overall}}}, \quad (9)$$

where $\text{TP}_{\text{category}}$ represents the number of samples correctly identified as belonging to that category by the model, while $\text{FN}_{\text{category}}$ represents the number of samples belonging to that category that were not identified by the model. $\text{Precision}_{\text{overall}}$ represents the proportion of samples predicted to belong to that category that do belong to it.

3.5 Performance comparison

To evaluate the performance of the proposed MCD-assisted multi-fidelity neural network framework, four sets

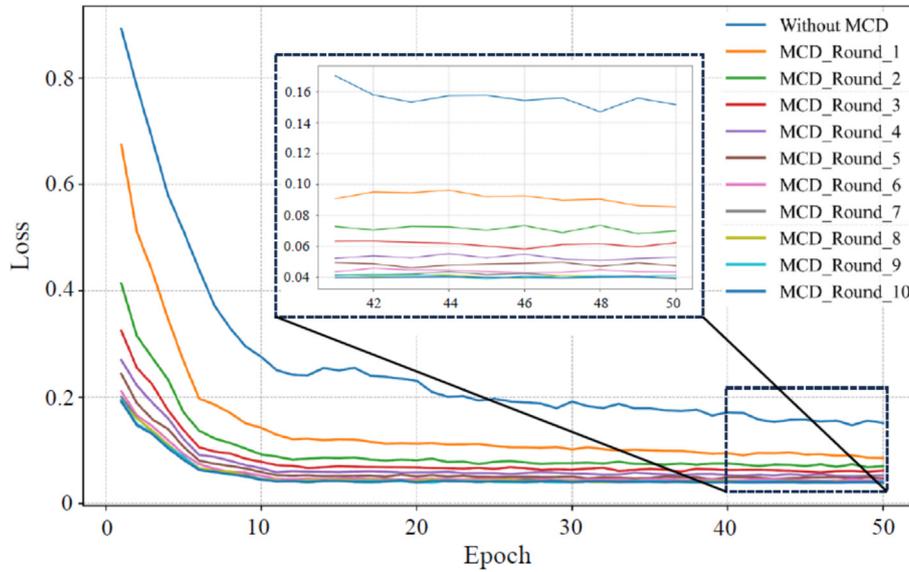


Fig. 7. Learning curve of the active learning strategy.

of experiments were designed. These experiments serve as an ablation study to quantify the impact of different components on model performance. The experimental configurations are as follows:

- (1) Multi-layer perceptron (MLP) model: a baseline model trained exclusively on high-fidelity data without incorporating multi-fidelity learning or MCD-based uncertainty quantification. This experiment assesses whether high-fidelity data alone are sufficient for accurate prediction and evaluates the extent to which class imbalance affects the model's performance.
- (2) Low-fidelity model prediction: a model trained solely on low-fidelity data to evaluate its predictive capability. By assessing the generalization ability of a knowledge-driven approach, this experiment examines whether low-fidelity data alone can effectively support real-world high-fidelity predictions.
- (3) MFNN model: a model trained using the partitioned initial training set and an unseen data pool, with the low-fidelity model output serving as auxiliary information for the high-fidelity component.
- (4) MCD-assisted MFNN: The MFNN was trained using the partitioned training set and unseen data pool, incorporating the MCD strategy during training. In each iteration, 5 % of the high-uncertainty samples from the data pool are selected using a sampling with replacement strategy and added to the training set. This experiment examines the effectiveness of MCD in mitigating class imbalance by prioritizing uncertain samples.

The regression results in Table 4 show varying model performance. The traditional MLP and low-fidelity models performed poorly, with MAEs of 0.54 and 0.81, respec-

tively. The MFNN model, by combining low-fidelity and high-fidelity data, reduces the MAE to 0.33, while the MCD-assisted MFNN model further reduces the MAE to 0.06, demonstrating superior predictive performance. In terms of the R^2 , the value (0.13) for the low-fidelity model indicates its inability to effectively capture the variations in the test set data. In contrast, the MFNN model achieves an R^2 of 0.51, which is much higher than that of the traditional MLP (0.24) and low-fidelity models, while the MCD-assisted MFNN model achieves an R^2 of 0.78, indicating optimal predictive accuracy.

While regression evaluation metrics provide insights into the model's predictive accuracy for clogging probability, they are insufficient for assessing classification performance. Due to threshold effects, improvements in MAE and R^2 do not always translate to higher classification accuracy. As a result, classification metrics such as Recall and F_1 -score are necessary to fully evaluate the model's ability to distinguish clogging and non-clogging cases. Table 5 shows the classification results of the models. The low-fidelity model showed strong bias, with a Recall of 0.88 for clogging events, but only 0.22 for non-clogging events, resulting in a poor F_1 -score of 0.28. The traditional MLP model performed slightly better with an F_1 -score of 0.59, achieving a Recall of 0.86 for non-clogging events, but only 0.43 for the clogging class. In contrast, the MFNN model significantly improved performance with an overall F_1 -score of 0.85, achieving a Recall of 0.99 for the non-clogging events and 0.69 for the clogging events. The MCD-assisted MFNN model performed best with an F_1 -score of 0.91, maintaining a high Recall of 0.98 for non-clogging while improving clogging detection to 0.81 Recall, demonstrating excellent performance in balancing prediction capabilities for both classes.

In summary, the MCD-assisted MFNN model excels in regression and classification tasks, significantly improving

Table 4
Performance comparison of different models for muck clogging regression prediction.

Error indicators	Traditional MLP	Low-fidelity model	MFNN		MCD-based MFNN	
	\hat{y}_T	\hat{y}_L	$\hat{y}_L + \hat{y}_H$	\hat{y}_R	$\hat{y}_L + \hat{y}_H$	\hat{y}_R
MAE	0.54	0.81	0.33	0.21	0.06	0.04
R^2	0.24	0.13	0.43	0.51	0.71	0.78

Note: \hat{y}_T denotes traditional MLP prediction; \hat{y}_L is the low-fidelity model prediction; $\hat{y}_L + \hat{y}_H$ is the MFNN output obtained by summing the low-fidelity prediction \hat{y}_L and the high-fidelity prediction \hat{y}_H ; and \hat{y}_R is the final prediction obtained by applying linear regression to $\hat{y}_L + \hat{y}_H$.

Table 5
Comparison of model classification performance.

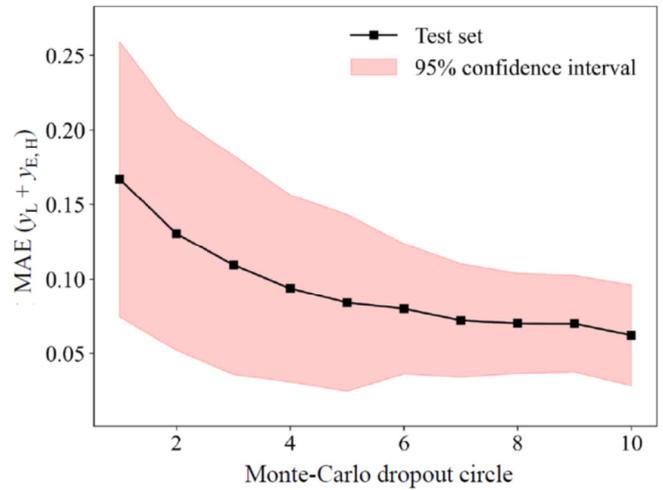
Method	Recall		F_1 -score
	Non-clogging	Clogging	
Traditional MLP	0.86	0.43	0.59
Low-fidelity model	0.22	0.88	0.28
MFNN	0.99	0.69	0.85
MCD-based MFNN	0.98	0.81	0.91

the accuracy and stability of muck-clogging predictions. By using the MCD strategy, the model is better equipped to handle high-uncertainty samples, resulting in improved generalization. In comparison, the MFNN model outperforms traditional MLP and low-fidelity models but still underperforms in contrast to the MCD-assisted MFNN.

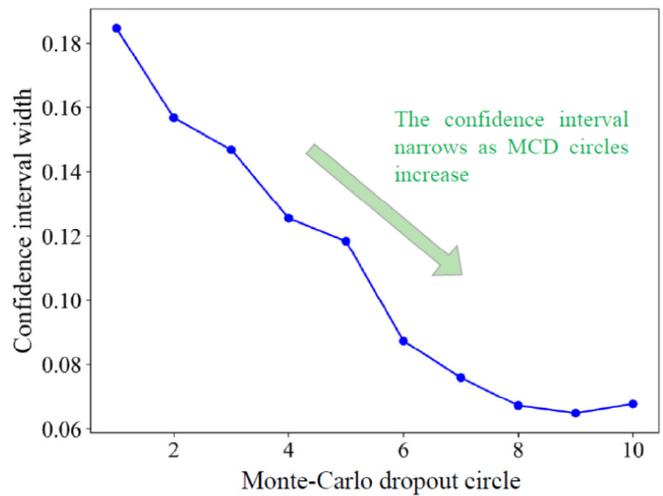
3.6 Performance analysis

Predicting muck clogging faces two main challenges: (1) data scarcity, as clogging events are rare but severe incidents during tunneling, and (2) data imbalance, with non-clogging events vastly outnumbering clogging events. The MCD-assisted MFNN model excels by combining MFNN’s ability to handle scarce data with MCD-assisted active learning to address data imbalance. Figure 8(a) shows that as the number of Monte-Carlo dropout circles increases, the predictive capability of the MCD-based MFNN model gradually improves. Meanwhile, the red-shaded 95% confidence interval narrows, indicating that the model’s predictive stability improves over successive iterations. Figure 8(b) further quantifies the reduction in uncertainty by plotting the width of the confidence interval across different Monte-Carlo dropout circles. The downward trend demonstrates that as the number of dropout circles increases, the predictive uncertainty systematically decreases. This confirms that the MCD-assisted MFNN framework effectively reduces uncertainty through iterative training, leading to improved model confidence in clogging predictions.

Further analysis was conducted on the samples selected from the data pool during each MCD iteration. Figure 9 shows the proportion of muck-clogging samples selected in each iteration. It can be observed that the proportion of clogging samples among the selected samples increases



(a)



(b)

Fig. 8. Loss values against MCD-assisted learning cycles in test samples. (a) Variation in MAE loss against MCD circles, and (b) variation in confidence interval width across MCD circles.

as the number of iterations grows, rising from less than 10% initially to nearly 60% by the end. This indicates that muck-clogging samples, due to their complexity and high uncertainty, are repeatedly selected for learning, making them the focus of model training. The complexity of clog-

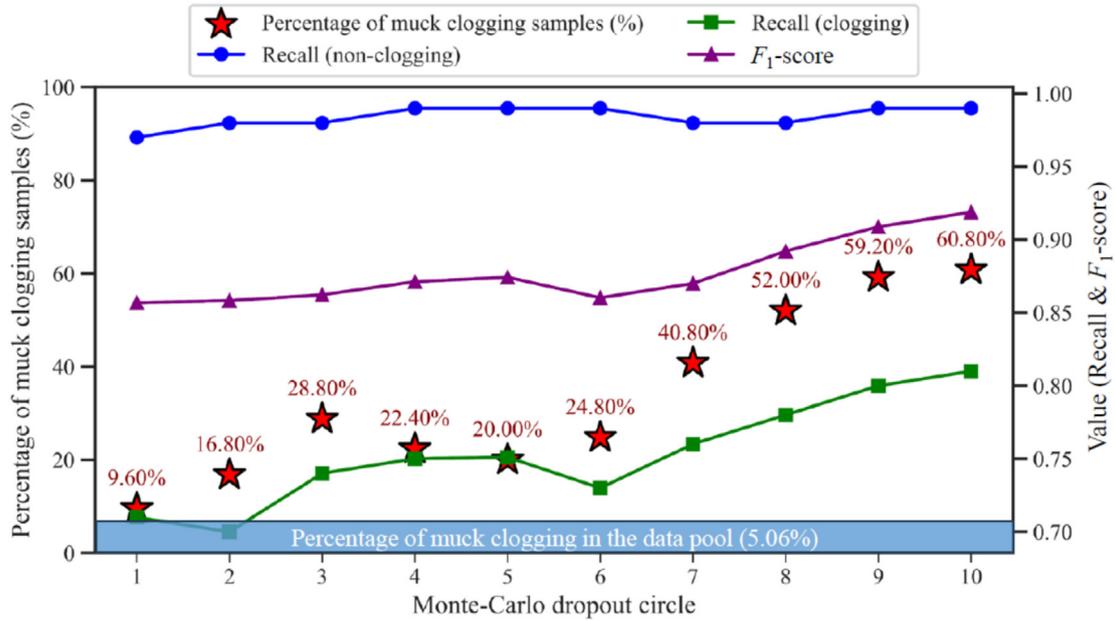


Fig. 9. Proportion of clogging samples selected in each iteration.

ging samples prevents the model from learning the underlying pattern effectively during the initial stages, necessitating repeated learning and adjustment in subsequent iterations. Additionally, Fig. 9 demonstrates how the model’s prediction performance evolves throughout the MCD iterations. As the number of iterations increases, Recall for clogging cases steadily improves, while Recall for non-clogging cases remains stable. The F₁-score also shows a consistent upward trend, indicating that the model benefits from progressively incorporating uncertain samples into the training process. These results confirm that MCD not only reinforces the learning of minority-class samples but also enhances its predictive performance, particularly in recognizing muck-clogging cases.

Figure 10 further corroborates the learning process by illustrating the proportion of muck-clogging samples selected during 10 rounds of active learning. Note that the x-axis represents the frequency of a sample being selected during ten iterations, rather than the iteration rounds, while the y-axis represents the proportion of muck clogging samples at different selection frequency ranges. For example, a point at (9, 50%) indicates that half of the samples selected nine times were muck-clogging events. The bubble size and the number in the bubble indicate the total number of samples at each selection frequency. Among the 299 samples selected only once, few represent clogging samples. However, as the selection frequency increases, the proportion of clogging samples increases

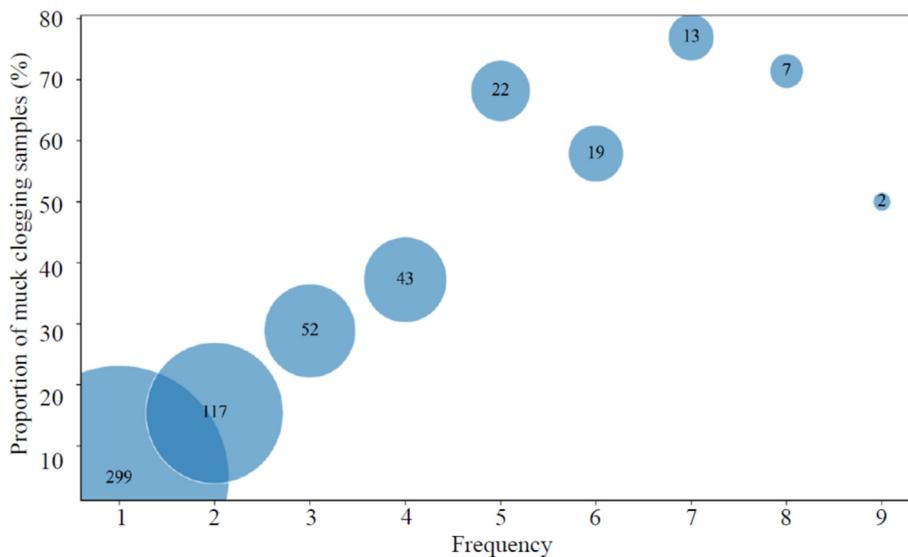


Fig. 10. Proportion of clogging samples across different MCD iteration frequencies.

significantly. For example, among samples selected four or more times, the proportion of clogging samples exceeds 40%, and in some high-frequency selected specimens, the percentage exceeds 60%. Since repeatedly selected samples are usually those that are challenging for the model to learn and have high uncertainty, this result indicates that muck-clogging samples are indeed among the most challenging samples.

Two key findings can be summarized from this analysis: (1) Clogging samples are harder for the model to learn than non-clogging samples. Their higher selection rate across iterations and frequent reselection indicates that the model struggles to grasp their patterns, requiring repeated training on these complex cases. (2) Sample quantity does not directly relate to learning difficulty. Although muck-clogging samples account for only 5.06% of the data pool, they are selected disproportionately often during training. Some non-clogging samples are selected multiple times, suggesting that these samples exhibit high predictive uncertainty. Therefore, simple resampling or undersampling strategies are insufficient for addressing data imbalance issues. More sophisticated training strategies are needed to handle these complex samples effectively.

4 Discussion

4.1 Cross-fidelity data fusion and utilization

In shield tunneling applications, data sources are often diverse and hierarchical. MFNN can effectively integrate data from different fidelity levels to address high-complexity and high-uncertainty tasks. Numerical simulations, empirical formulas, or lower-precision measurements can be regarded as low-fidelity data, while precise sensor information collected in the field is a type of high-fidelity data. Restricted to the high cost and difficulty of obtaining high-fidelity data, MFNN can be a potential scheme by utilizing high-fidelity and low-fidelity data for joint training, allowing the model to learn basic patterns from low-fidelity data and then refine them with high-fidelity data, forming an efficient multi-level learning strategy.

This cross-fidelity fusion not only enhances prediction accuracy but also improves generalization. The extensive coverage of low-fidelity data allows the model to acquire prior knowledge, facilitating adaptation to unseen conditions. Meanwhile, high-fidelity data act as a correction mechanism, ensuring more reliable predictions across varying geological and operational scenarios. Consequently, MFNN effectively generalizes across different shield tunneling environments, making it better suited for real-world engineering applications.

4.2 Uncertainty identification methods

When discussing uncertainty analysis methods, committee-based active learning is also commonly used to quantify predictive uncertainty for surrogate models.

In committee models, multiple models (usually with different initializations or architectures) are constructed to make predictions on the same input, and if these models exhibit significant disagreement in their predictions, it indicates that the sample has high uncertainty (Yuan et al., 2024).

However, in practical applications, committee models require multiple models to be trained in each round of active learning, resulting in high computational demands, especially with large datasets and numerous models. In contrast, MCD shows greater computational efficiency for uncertainty quantification by repeatedly applying dropout within a single model, avoiding the need for training multiple independent models, thereby significantly reducing computational costs. Therefore, in situations with high training demands, MCD has an efficiency advantage over committee model-based methods.

4.3 Comparison of traditional imbalance data processing methods

In classification tasks, common data imbalance handling techniques include oversampling, undersampling, weight adjustment method, and the synthetic minority oversampling technique (SMOTE). Oversampling increases the number of minority class samples by duplicating existing instances, whereas undersampling reduces the number of majority class samples to achieve balance. SMOTE generates synthetic samples through interpolation, thereby expanding the distribution range of the minority class to some extent. In contrast, the weight adjustment method assigns higher weights to the loss function of minority class samples, increasing the model's attention to this category.

In the weight adjustment method, assuming that the loss function of the model adopts the cross-entropy loss:

$$L = - \sum_{c=1}^C y_c \ln(\hat{y}_c), \quad (10)$$

where y_c represents the true class label, and \hat{y}_c represents the predicted probability of class c , and C is the total number of classes.

When dealing with class imbalance, a weight factor w_c can be introduced to adjust the contribution of different classes to the loss function:

$$L = - \sum_{c=1}^C w_c y_c \ln(\hat{y}_c), \quad (11)$$

$$w_c = \frac{1}{N_c}, \quad (12)$$

where N_c denotes the number of samples in class c .

To evaluate the effectiveness of these methods, this study adopts the same cross-validation approach as described in Section 3.3, applying different data imbalance handling techniques to the training set. The detailed parameters for each method are presented in Table 6, and their predictive performance is compared against the traditional MLP

Table 6
Parameter settings for traditional imbalanced data processing methods.

Method	Parameter	Range	Value
Undersampling	Sampling_strategy	Categorical (['float', 'str', 'dict', 'callable'])	Auto
	Replacement	Categorical (['True', 'False'])	False
Oversampling	Sampling_strategy	Categorical (['float', 'str', 'dict', 'callable'])	Auto
Weight adjustment method	Weight calculation	Categorical ($[1/N_c, 1/\sqrt{N_c}]$)	$1/N_c$
SMOTE	Sampling_strategy	Categorical (['float', 'str', 'dict', 'callable'])	Auto
	$K_neighbors$	Integer (≥ 1)	4

Table 7
Comparison of traditional data imbalance strategy performance.

Method	Recall		F_1 -score
	Non-clogging	Clogging	
Traditional MLP	0.86	0.43	0.59
MFNN	0.99	0.69	0.85
MCD-based MFNN	0.98	0.81	0.91
Undersampling	0.81	0.44	0.53
Oversampling	0.83	0.56	0.71
Weight adjustment method	0.81	0.54	0.68
SMOTE	0.86	0.55	0.70

model (trained solely on high-fidelity data), MFNN, and MCD-based MFNN. Experimental results (Table 7) indicate that, except for undersampling, which reduces predictive performance due to information loss, all other methods improve the model's ability to classify minority class samples to some extent. However, the improvement remains limited, and overall, these methods still underperform compared to the MFNN framework. In contrast, MFNN provides additional prior knowledge through low-fidelity data, enabling the model to acquire an initial understanding of the minority class. As a result, even when the number of minority class samples is small, the model can still maintain relatively high predictive performance.

Although traditional data imbalance handling methods improve minority class learning to a certain extent, they also present several limitations. Oversampling can increase the quantity of minority class samples but is prone to overfitting, as the model repeatedly learns from the same duplicated samples, which affects generalization performance. Undersampling balances the class distribution by reducing majority class samples, but this approach may cause information loss. When the dataset is small, it weakens the model's ability to learn majority class patterns, ultimately compromising overall performance. SMOTE expands the diversity of minority class samples by generating synthetic data through interpolation, but these synthetic samples may not fully align with real-world engineering data distributions, potentially leading the model to learn patterns that deviate from actual conditions. The weight adjustment method enhances the model's focus on minority class samples by assigning higher loss function weights. However, its effectiveness depends on appropriate weight selection, and

in real-world engineering applications, class proportions are often unknown or dynamically changing over time, making it challenging to determine an optimal weighting strategy.

4.4 Limitation

The proposed MCD-assisted MFNN framework demonstrates superior performance in addressing data imbalance in shield tunneling applications. However, its applicability remains constrained by the construction of low-fidelity data, which relies on engineering heuristics and expert-derived knowledge (Conti et al., 2023; He et al., 2024). For instance, the muck clogging risk assessment chart, derived from engineering experience and experimental studies, provides an initial classification capability. Nevertheless, in scenarios where reliable empirical rules are unavailable or difficult to quantify, the generation of low-fidelity data becomes challenging. This limitation is particularly evident in unconventional geological conditions or extreme tunneling scenarios (e.g., high groundwater pressure environments, highly heterogeneous soil compositions), where a lack of sufficient engineering experience or statistical feasibility makes it difficult to construct effective low-fidelity datasets, thereby affecting the training efficiency of the MFNN framework.

Furthermore, the reliance on empirical knowledge is a double-edged sword in terms of model generalization (Zhang et al., 2024a). When empirical rules cover a broad range of geological conditions, the generated low-fidelity data can represent diverse tunneling scenarios, enabling the MFNN model to achieve greater generalization and adaptability across different shield tunneling environments. However, if the empirical rule set is too narrow, the effectiveness of the low-fidelity data diminishes, potentially limiting the model's exposure to a sufficiently diverse set of conditions. As a result, the model may struggle to learn robust decision boundaries for minority class samples, ultimately impairing predictive performance.

5 Conclusions

This study proposed an MCD-assisted MFNN framework to predict muck clogging risk in shield tunneling. The muck-clogging risk chart knowledge, summarized

from experiments and engineering experience, was represented in data form to synthesize the low-fidelity dataset, which was combined with high-fidelity data collected from actual tunneling projects to train the model. The results demonstrate that the presented MCD-assisted MFNN framework shows superior predictive performance and generalization capabilities compared to traditional machine learning models when dealing with complex and imbalanced data, such as the prediction of muck clogging during shield tunneling. The specific conclusions are as follows:

- (1) The presented method utilizes a knowledge chart accumulated in practice to synthesize muck-clogging data for multi-fidelity learning. This approach not only digitizes engineering experience but also provides a new way to tackle the challenge of insufficient data in complex engineering problems such as shield tunneling.
- (2) The MFNN model used in this study effectively integrates low-fidelity and high-fidelity data. Through joint training, the model learns basic patterns from low-fidelity data and fine-tunes them on high-fidelity data, forming an efficient multi-level learning system. This multi-fidelity fusion method allows the model to utilize all available information for improved predictions.
- (3) The MCD-based active learning strategy effectively identifies samples with high uncertainty, and iteratively trains the high-fidelity model on these samples, significantly increasing the model's focus on high-uncertainty samples and enhancing the predictive capability of the high-fidelity model.
- (4) The use of uncertainty identification revealed that muck-clogging samples are generally harder to learn than samples from other categories. Additionally, in imbalanced datasets, the proportion of samples in different classes does not directly correlate with their learning difficulty. Classes with many samples can still contain difficult samples, while classes with fewer proportions of samples may have easier ones. This indicates that simple resampling or undersampling methods are not always effective in addressing the problem of imbalanced samples.

Data availability

The data and code used in this study are hosted on the Clear Data Bay platform (<https://www.cleardatabay.com/index.php?c=show&id=37>).

CRedit authorship contribution statement

Xiao Yuan: Methodology, Investigation, Writing – review & editing, Writing – original draft, Validation, Supervision, Conceptualization. **Shuying Wang:** Supervi-

sion, Project administration, Methodology, Funding acquisition, Conceptualization. **Tongming Qu:** Visualization, Validation, Software, Investigation, Data curation, Conceptualization. **Huanhuan Feng:** Resources, Investigation, Validation. **Pengfei Liu:** Validation, Supervision, Resources. **Junhao Zeng:** Software, Investigation.

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.

Acknowledgement

This study is supported by the National Natural Science Foundation of China (Grant No. 52022112).

References

- Basnet, P. M. S., Mahtab, S., & Jin, A. B. (2023). A comprehensive review of intelligent machine learning based predicting methods in long-term and short-term rock burst prediction. *Tunnelling and Underground Space Technology*, 142, 105434.
- Bozorgzadeh, N., Harrison, J. P., & Escobar, M. D. (2019). Hierarchical Bayesian modelling of geotechnical data: Application to rock strength. *Geotechnique*, 69(12), 1056–1070.
- Cachim, P., & Bezuijen, A. (2019). Modelling the torque with artificial neural networks on a tunnel boring machine. *KSCE Journal of Civil Engineering*, 23(10), 4529–4537.
- Cai, Y. M., Ching, J., & Phoon, K. K. (2024). Tailored clustering method to identify quasi-regional sites. *Engineering Geology*, 333, 107490.
- Chen, R. P., Zhang, P., Kang, X., Zhong, Z. Q., Liu, Y., & Wu, H. N. (2019). Prediction of maximum surface settlement caused by earth pressure balance (EPB) shield tunneling with ANN methods. *Soils and Foundations*, 59(2), 284–295.
- Chen, X. S., Shen, J., Bao, X. H., Wu, X. L., Tang, W. C., & Cui, H. Z. (2023). A review of seismic resilience of shield tunnels. *Tunnelling and Underground Space Technology*, 136, 105075.
- Conti, P., Guo, M. W., Manzoni, A., & Hesthaven, J. S. (2023). Multi-fidelity surrogate modeling using long short-term memory networks. *Computer Methods in Applied Mechanics and Engineering*, 404, 115811.
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning* (pp. 1050–1059). New York, USA.
- He, G. F., Zhang, P., & Yin, Z. Y. (2024). Active learning inspired multi-fidelity probabilistic modelling of geomaterial property. *Computer Methods in Applied Mechanics and Engineering*, 432, 117373.
- Hou, S. K., Liu, Y. R., & Yang, Q. (2022). Real-time prediction of rock mass classification based on TBM operation big data and stacking technique of ensemble learning. *Journal of Rock Mechanics and Geotechnical Engineering*, 14(1), 123–143.
- Huang, H. W., Chang, J. Q., Zhang, D. M., Zhang, J., Wu, H. M., & Li, G. (2022a). Machine learning-based automatic control of tunneling posture of shield machine. *Journal of Rock Mechanics and Geotechnical Engineering*, 14(4), 1153–1164.
- Huang, X., Zhang, Q. T., Liu, Q. S., Liu, X. W., Liu, B., Wang, J. J., & Yin, X. (2022b). A real-time prediction method for tunnel boring machine cutter-head torque using bidirectional long short-term memory networks optimized by multi-algorithm. *Journal of Rock Mechanics and Geotechnical Engineering*, 14(3), 798–812.
- Jiang, X., Zhu, H. H., Yan, Z. G., Zhang, F. S., Ye, F., Li, P. N., Zhang, X. H., Dai, Z. R., Bai, Y., & Huang, B. S. (2023). A state-of-art review on development and progress of backfill grouting materials for shield tunneling. *Developments in the Built Environment*, 16, 100250.
- Kong, X. X., Ling, X. Z., Tang, L., Tang, W. C., & Zhang, Y. F. (2022). Random forest-based predictors for driving forces of earth pressure

- balance (EPB) shield tunnel boring machine (TBM). *Tunnelling and Underground Space Technology*, 122, 104373.
- Li, D. Y., Liu, Z. D., Xiao, P., Zhou, J., & Armaghani, D. J. (2022). Intelligent rockburst prediction model with sample category balance using feedforward neural network and Bayesian optimization. *Underground Space*, 7(5), 833–846.
- Li, J. B., Chen, Z. Y., Li, X., Jing, L. J., Zhang, Y. P., Xiao, H. H., Wang, S. J., Yang, W. K., Wu, L. J., Li, P. Y., Li, H. B., Yao, M., & Fan, L. T. (2023). Feedback on a shared big dataset for intelligent TBM Part I: Feature extraction and machine learning methods. *Underground Space*, 11, 1–25.
- Li, J. H., Guo, D., Chen, Z. Y., Li, X., & Li, Z. F. (2024). Transfer learning for collapse warning in TBM tunneling using databases in China. *Computers and Geotechnics*, 166, 105968.
- Liu, D. J., Zhang, W. P., Dai, Q. Q., Chen, J. Y., Duan, K., & Li, M. Y. (2024). Safety evaluation method for operational shield tunnels based on semi-supervised learning and a stacking algorithm. *Tunnelling and Underground Space Technology*, 153, 106027.
- Luo, H., Fang, Y., Wang, J. F., Wang, Y. B., Liao, H., Yu, T., & Yao, Z. G. (2023). Combined prediction of rockburst based on multiple factors and stacking ensemble algorithm. *Underground Space*, 13, 241–261.
- Qu, T. M., Guan, S. H., Feng, Y. T., Ma, G., Zhou, W., & Zhao, J. D. (2023a). Deep active learning for constitutive modelling of granular materials: From representative volume elements to implicit finite element modelling. *International Journal of Plasticity*, 164, 103576.
- Qu, T. M., Zhao, J. D., Guan, S. H., & Feng, Y. T. (2023b). Data-driven multiscale modelling of granular materials via knowledge transfer and sharing. *International Journal of Plasticity*, 171, 103786.
- Rezvani, S., & Wang, X. Z. (2023). A broad review on class imbalance learning techniques. *Applied Soft Computing*, 143, 110415.
- Su, M. M., Guo, N., & Yang, Z. X. (2023). A multifidelity neural network (MFNN) for constitutive modeling of complex soil behaviors. *International Journal for Numerical and Analytical Methods in Geomechanics*, 47(18), 3269–3289.
- Thewes, M., & Hollmann, F. (2016). Assessment of clay soils and clay-rich rock for clogging of TBMs. *Tunnelling and Underground Space Technology*, 57, 122–128.
- Tran, N., Chen, H. H., Bhuyan, J., & Ding, J. H. (2022). Data curation and quality evaluation for machine learning-based cyber intrusion detection. *IEEE Access*, 10, 121900–121923.
- Wang, L., Pan, Q. J., & Wang, S. Y. (2024). Data-driven predictions of shield attitudes using Bayesian machine learning. *Computers and Geotechnics*, 166, 106002.
- Wang, S. Y., Liu, P. F., Hu, Q. X., & Zhong, J. Z. (2020). Effect of dispersant on the tangential adhesion strength between clay and metal for EPB shield tunnelling. *Tunnelling and Underground Space Technology*, 95, 103144.
- Wang, S. Y., Liu, T. Y., Zheng, X. C., Yang, J. S., & Yang, F. (2024a). Dynamic collapse characteristics of the tunnel face induced by the shutdown of earth pressure balance shields (EPB): A 3D material point method study. *Underground Space*, 16, 164–182.
- Wang, S. Y., Ni, Z. L., Qu, T. M., Wang, H. B., & Pan, Q. J. (2022). A novel index to evaluate the workability of conditioned coarse-grained soil for EPB shield tunnelling. *Journal of Construction Engineering and Management*, 148(6), 04022028.
- Wang, S. Y., Yuan, X., & Qu, T. M. (2024b). Machine learning-informed soil conditioning for mechanized shield tunneling. *Computer-Aided Civil and Infrastructure Engineering*, 39(17), 2662–2682.
- Wang, S. Y., Zhu, H. B., Liu, P. F., & Qu, T. M. (2024c). Soil slaking under the effect of dispersants: Characteristics and mechanism. *Environmental Earth Sciences*, 83, 397.
- Xu, Q. H., Huang, X., Zhang, B. G., Zhang, Z. X., Wang, J. H., & Wang, S. F. (2023). TBM performance prediction using LSTM-based hybrid neural network model: Case study of Baimang River tunnel project in Shenzhen, China. *Underground Space*, 11, 130–152.
- Yuan, X., Wang, S. Y., Qu, T. M., Feng, H. H., Liu, P. F., Zeng, J. H., & Chen, X. S. (2024). Learning the hard-to-learn: Active learning for imbalanced datasets in data-centric tunnel engineering. *Computers and Geotechnics*, 174, 106629.
- Zendaki, Y., Cao, B. T., Alsahly, A., Freitag, S., & Meschke, G. (2024). A simulation-based software to support the real-time operational parameters selection of tunnel boring machines. *Underground Space*, 14, 176–196.
- Zeng, R. K., Wang, S. Y., Zhang, Y., & Qu, T. M. (2025). CFD-DEM modeling of seepage in foam-conditioned soil. *Computers and Geotechnics*, 177, 106818.
- Zhang, D. M., Fu, L., Huang, H. W., Wu, H. M., & Li, G. (2023a). Deep learning-based automatic detection of muck types for earth pressure balance shield tunneling in soft ground. *Computer-Aided Civil and Infrastructure Engineering*, 38(7), 940–955.
- Zhang, P., Sheil, B., & Cheng, Q. (2024a). Multi-fidelity learned emulator for waves and porous coastal structures interaction modelling. *Computers and Geotechnics*, 176, 106718.
- Zhang, P., Yin, Z. Y., & Sheil, B. (2024b). Multifidelity constitutive modeling of stress-induced anisotropic behavior of clay. *Journal of Geotechnical and Geoenvironmental Engineering*, 150(3), 04024003.
- Zhang, P., Yin, Z. Y., Jin, Y. F., Yang, J., & Sheil, B. (2022). Physics-informed multifidelity residual neural networks for hydromechanical modeling of granular soils and foundation considering internal erosion. *Journal of Engineering Mechanics*, 148(4), 04022015.
- Zhang, W. S., Liu, X. Y., Zhang, L. Y., & Wang, Y. D. (2023b). Intelligent real-time prediction of multi-region thrust of EPB shield machine based on SSA-LSTM. *Engineering Research Express*, 5(3), 035013.