Mechanical Systems and Signal Processing 162 (2022) 108035

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Generative adversarial network for geological prediction based on TBM operational data

Chao Zhang [a,b], Minming Liang [c], Xueguan Song [d,*], Lixue Liu [a], Hao Wang [a], Wensheng Li [a], Maolin Shi [e]

[a] *School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China*
[b] *Key Laboratory for Computational Mathematics and Data Intelligence of Liaoning Province, Dalian University of Technology, Dalian 116024, China*
[c] *SenseTime, Shenzhen, Guangdong 518063, China*
[d] *School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China*
[e] *School of Agricultural Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China*

ARTICLE INFO

Communicated by Jerome Antoni

Keywords:
Geological prediction
Generative adversarial network
Geological condition
Feature extraction
Tunnel boring machine

ABSTRACT

The prediction of tunnel geological conditions plays an important role in underground engineering, such as the tunnel construction and tunnel dynamic design. However, due to the invisibility of underground geological conditions, there remain many challenges in the design of geological prediction models. In this paper, we propose a generative adversarial network for geological prediction (GAN-GP) to accurately estimate the thickness of each rock-soil type in a tunnel boring machine (TBM) construction tunnel based on operational data collected from sensors equipped on the TBM. The generator of the GAN-GP contains feature-extraction (FE) and feature-integration (FI) modules. The former extracts the important features from the TBM operational data, and the latter produces the geological condition prediction, which estimates the thickness of each rock-soil type at a location. The discriminator of the GAN-GP determines whether the FI module's outputs are true geological data. After adversarial training, if the trained discriminator fails to distinguish them, the outputs of the FI module will accurately approximate the true geological condition. Experimental results support the effectiveness of the proposed GAN-GP model for geological prediction, and show that it outperforms the state-of-the-art models including support vector regression (SVR), feed-forward neural network (FNN) and random forest (RF) models.

1. Introduction

In recent decades, there has been a large demand for tunnel construction in transportation and public traffic projects. Evidently, the conventional tunneling technique (*i.e.,* drilling and blasting) has been unable to meet these increasing requirements due to its low efficiency and high environmental damage. In contrast, the tunnel boring machine (TBM), a representative mechanical tunneling equipment, has been widely used in various kinds of underground construction, such as the mine development [1], soil movement [2] and water conveyance projects [3], because of the advantages of low labor intensity, low environmental damage, rapid excavation

[*] Corresponding author.
E-mail address: sxg@dlut.edu.cn (X. Song).

https://doi.org/10.1016/j.ymssp.2021.108035
Received 30 March 2020; Received in revised form 25 April 2021; Accepted 8 May 2021
Available online 24 May 2021

speed and high safety level. In addition, TBM performance prediction [4,5] and TBM design optimization [6–8] have attracted much research interest.

However, complex geological conditions (*e.g.,* water leakage, weak regions or voids) not only seriously delay TBM construction schedules with large extra costs and causing instrument damage but also bring a high risk to the excavation process with additional hazards and even tremendous casualties. Therefore, one main challenge of the current TBM tunneling technique is to accurately and efficiently foresee unexpected geological condition changes [9]. Extensive research works have studied tunnel geological condition prediction. In general, the prediction tasks are mainly addressed through two different strategies: one strategy is based on geological analysis methods, which directly analyze the rock-soil samples taken from tunnel sections [10,11], and the other strategy is based on geophysical methods, which explore the geological characteristics of tunnel sections by measuring their physical characteristics [12–16]. These direct detection strategies for predicting tunnel geological conditions lack generalizability due to two aspects: 1) the prediction length is restricted by the performance of the sampling equipment, and 2) their usage is strongly dependent on the specific construction situations. These facts restrict the applicability of the two strategies in practical TBM construction, which usually requires that some specific rock-soil types be accurately and promptly predicted in front of a few (or tens of) meters from the current TBM location.

1.1. *Background and motivation*

To overcome the limitations of direct detection strategies, machine learning techniques have been widely used to build models for predicting tunnel geological conditions, where geological prediction is treated as a supervised learning task that explores the mapping relationship between observable data and geological conditions. There are two key points to this strategy: to acquire the observable data that are strongly related to the tunnel geological condition, and to develop the learning models based on relatively rare labeled data and massive unlabeled data.

One common method is to develop predictors with outputs that are specific indicators associated with rock-soil types. For example, Zhuang et al. [17] used support vector regression (SVR) to predict several mechanical indicators of surrounding rock masses, including gravity, Poisson’s ratio, elastic modulus cohesion, cohesion, friction angle and layer thickness. After inputting the mechanical parameters of the previous *twenty* locations, the SVR model provides the mechanical parameters of the next *five* locations. Alimoradi *et al.* [18] proposed an artificial neural network (ANN) to classify the rock mass rating, and the six input attributes of the network were the P-wave velocity, S-wave velocity, orientation, magnitude, wave type and x-coordinate, which were collected from the tunnel sensors. Von and Ismail [19] developed another ANN model for the prediction of rock grade points by setting the P-wave velocity, the S-wave velocity and the wave type as the input attributes. However, these models were designed only for the prediction task of a limited number of rock-soil types, and thus are unsuitable (at least they cannot be directly applied) to the prediction of complex geological conditions.

The TBM operational data, collected by the sensors equipped on a TBM, recode the continuous changes of the TBM’s running state, which is strongly related to the geological condition of the construction tunnel. Therefore, many research works use TBM operational data to develop prediction models for tunnel geological conditions, as illustrated in Fig. 1. Zhang et al. [20] adopted support vector machine (SVM) for classifying the rock-soil types appearing in the TBM construction tunnels. Liu et al. [21] combined SVR with stacked single-target technology to establish a predictor for rock mass parameters. Jung *et al.* [22] proposed an ANN model for

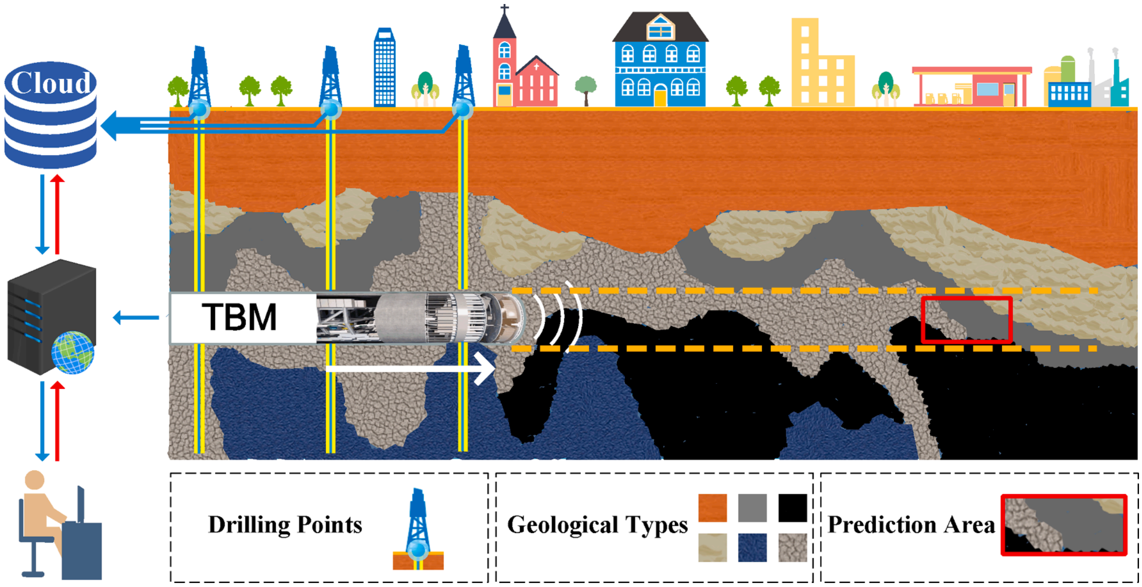


Fig. 1. Illustration of tunnel geological condition prediction based on TBM operational data.

predicting the ground condition ahead of the tunnel face by using TBM operational parameters, including the penetration rate, thrust force, cutter torque and all pairs of parameters. In [23,24], an ensemble of long short-term memory networks with additional one-dimensional convolutional layers was designed to describe the rock mass behavior based on TBM operational data. Shi et al. [25] made a comparison among several kinds of machine learning models for the geological prediction task. Liu et al. [26] employed a backpropagation (BP) neural network to predict the rock mass parameters in a TBM tunnel. Zhao et al. [27] designed a four-layer feedforward neural network (FNN) to predict the thickness of the specific rock-soil types, where the *seven* nodes of the 2nd hidden layer correspond to the *seven* kinds of physical–mechanical indicators associated with the TBM tunneling process. Moreover, Li et al. [28] designed a diagonal recurrent neural network for the predictive control of slurry pressure balance in the TBM tunneling process. However, it is still challenging to address the imbalance between the relatively few labeled data and massive unlabeled data. It is important to find a learning model that has a powerful mapping capability with a low demand on the size of labeled training samples. Subsequently, we will show that a neural network equipped with a generative adversarial structure is capable of meeting the demand better than conventional machine learning models.

1.2. Generative adversarial networks (GANs)

Generative adversarial networks (GANs), originally proposed by Goodfellow [29], refer to a class of neural networks that are composed of two networks: a generator and a discriminator, and are trained via the contest between them in a min–max game. Given an input, which could be a real or a random noise datum in the different learning tasks, the generator will produce a new instance and the discriminator will determine whether the instance belongs to the actual training dataset or not. The essential idea of GANs is to let the two networks compete with each other during training until they nearly achieve Nash equilibrium. In particular, when the discriminator is trained to be incapable of distinguishing an output of the generator from a real instance, the distribution of the instance provided by the generator will approximate that of the real instance as much as possible. Such an adversarial training strategy makes the generator produce an accurate approximation of the real data with a relatively lower demand on the size of the training samples than traditional learning models. Therefore, GANs and their variants [30–34] have been widely used in many practical applications, *e.g.,* computer vision [35,36], natural language processing [37,38] and three-dimensional object modeling [39].

1.3. Overview of main results

In this paper, we propose a GAN for geological prediction (GAN-GP) of a TBM construction tunnel based on operational data collected from sensors equipped on the TBM. The generator of the GAN-GP produces the estimation of the thickness of each rock-soil type appearing at a location of the tunnel section, and the discriminator of the GAN-GP determines whether the outputs of the generator are the real geological data or not.

In particular, the generator is composed of two modules: a feature extraction (FE) module that extracts the important features from the original operational data and a feature integration (FI) module that integrates the outputs of the FE module to estimate the thickness of each rock-soil type appearing at the current location. The input of the discriminator of the GAN-GP is either a real geological datum, labeled as 'true', or the output of the generator, labeled as 'fake'. When the discriminator is trained to be incapable of determining whether its input is the real geological datum or not, the generator will produce an accuracy approximation of the true geological condition of the tunnel section. To improve the training efficiency and stability, we also introduce two extra tricks: the pre-training trick and the teacher-loss trick. The former minimizes the teacher loss of the generator to find reasonable initial generator weights for the subsequent adversarial training, and the latter balances the individual training progresses of the generator and the discriminator during adversarial training. The numerical experiments validate the proposed GAN-GP with the two tricks for the tunnel geological condition prediction task.

The rest of this paper is organized as follows. In Section 2, we preprocess the raw TBM operational data. In Section 3, we introduce the structure and the adversarial training strategy of the proposed GAN-GP. In Section 4, we conduct a numerical experiment to validate the GAN-GP, and the last section concludes the paper.

2. Data description and preprocessing

In this section, we first introduce the background of the research issue and then show preprocessing steps of the raw TBM operational data.

2.1. Data description

The tunnel geological condition prediction task of interest is associated with an urban subway construction project in Shenzhen, a city in China. The project was processed by using an earth pressure balance shield TBM, which consists of a cutter-head, a chamber, a screw conveyor, tail skin and auxiliaries. The TBM has a diameter of 6.2 m and a total mass of over 500 t, and the cutterhead features an opening percentage of 30% and 120 cutters with a frequency of 1 Hz.

The tunnel is about 2000 m long and 6.3 m wide. The engineering route is divided into 1365 ring sections and each section is approximately 1.5 m long. The range of the ground surface elevation is $0.2 \sim 5.8$ m, and the depth of the tunnel floor from the ground surface is within $11.8 \sim 25.4$ m. The stratum can be divided into five layers in terms of the rock-soil types, and each layer can be further divided into $2 \sim 7$ sublayers according to physical–mechanical indicators. In general, there are approximately *twenty* rock-soil types,

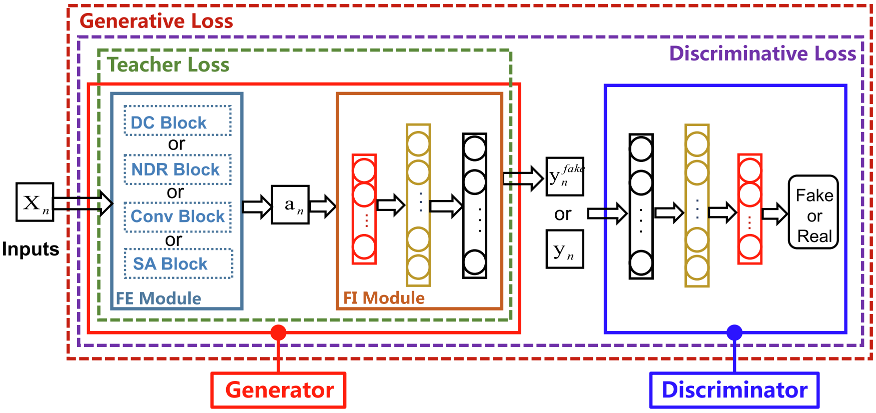


Fig. 2. The structure of the GAN-GP, where the hidden layers in the same color have identical structures.

which are identified by using different values of *seven* common physical–mechanical indicators: the natural severity (Y), internal friction angle (quick direct shear test) (FI), deformation modulus (EM), Poisson's ratio (P), coefficient of lateral pressure (SITA), permeability coefficient (K), and cohesive strength between the rock mass and anchors (FRB) (*cf.* Table A.3).[1] However, only *eleven* out of the *twenty* rock-soil types appear in the geological samples, so we omit the rest (*cf.* Table A.4).

The TBM operational data are collected by sensors equipped on the TBM. In total, approximately 4.6 million operational data points were collected, and each datum has $I = 69$ attributes corresponding to the different operational parameters, such as torque, thrust, tunneling speed and fuel tank temperature (*cf.* Table A.5). After data cleaning, 3.18 million operational data points remain.

The geological condition samples are drawn from 88 points from the entire construction tunnel by the drilling method, and the operational data within 0.3 m around a drilling point can be considered to have the same geological condition as the drilling samples. In this manner, we finally obtain 61,788 operational data labeled with the thickness of each rock-soil type.

2.2. Data Preprocessing

The raw TBM operational data are indexed by the discontinuous operation time because the construction will be intermittent in some cases, *e.g.,* ceasing for equipment maintenance or rest. To develop a continuous index system for the operational data, we use the operation parameter 'advance rate' to locate the position of each datum based on the integral, *i.e.,* the TBM's relative displacement at the j-th time point of the i-th ring section can be expressed as

$$s_{i,j} = \sum_{k=0}^{i-1} L_k + \int_{t_{i,0}}^{t_{i,j}} v_i(t)dt, \quad i = 1,\ldots,88. \tag{1}$$

where L_k stands for the length of the k-th ring section, $v_i(t)$ is the function of the 'advance rate' in the i-th ring section, and $t_{i,0}$ and $t_{i,j}$ are the initial time point and the j-th time point of the i-th ring section, respectively. Then, the sequence $\{s_{i,j}\}$ provides a continuous index system of the TBM operational data, and we denote the set of operational data as $\{\mathbf{x}_s\}_{s>0} \subset \mathbb{R}^I$ if no confusion arises. To exploit the sequential characteristics of the TBM operational data with the continuous index system, we concatenate τ-length continuous operational data $\mathbf{x}_{s_{n\tau+1}}, \mathbf{x}_{s_{n\tau+2}}, \cdots, \mathbf{x}_{s_{n\tau+\tau}}$ to generate a sample:

$$\mathbf{X}_n = \left[\mathbf{x}_{s_{n\tau+1}}, \mathbf{x}_{s_{n\tau+2}}, \cdots, \mathbf{x}_{s_{n\tau+\tau}}\right] \in \mathbb{R}^{69\times\tau},$$

where τ is the step size parameter. Then, we form the sample set $\{(\mathbf{X}_n, \mathbf{y}_n)\}_{n=1}^{N_\tau}$ with $\mathbf{y}_n = \left(y_n^{(1)}, \cdots, y_n^{(11)}\right)^T \in [0,1]^{11}$, where N_τ is the sample size associated with the step size τ, and $y_n^{(j)}$ stands for the normalized thickness of the j-th rock-soil type ($j = 1,2,\cdots,11$). Note that there exists a tradeoff in the choice of τ: a large τ can magnify the dissimilarity among these samples, but the sample size will accordingly become too small to guarantee the training quality. In subsequent experiments, we will empirically explore the relationship between the choice of τ and the prediction performance and find that a small τ is suitable for training the GAN-GP.

[1] In fact, there are additional kinds of frequently used rock-soil physical–mechanical indicators, such as the natural moisture content, pore ratio, cohesive force (quick direct shear test), internal friction angle (consolidation quick direct test), compression modulus, coefficient of subgrade reaction (vertical & horizontal) and uniaxial compressive strength of rock (saturation & natural). Since these indicators are not shared by all rock-soil types, we omit them here.

3. Generative adversarial network for geological prediction (GAN-GP)

In this section, we will present the structure of the GAN-GP, including the generator and the discriminator (*cf.* Fig. 2). Moreover, we introduce the specific loss functions and the corresponding adversarial training strategy for the GAN-GP.

3.1. Generator of the GAN-GP

Since the geological information is encoded in the operational data around the point, the design of GAN-GP needs to exhaustively explore the sequential characteristics of the operational data, and thus the generator, denoted by G, contains two modules: the FE module and the FI module. The following are the details of the two modules.

3.1.1. Feature extraction (FE) module

Instead of human-made feature extraction, the FE module aims to self-learn the relatedness among the different samples and then extracts the important features as the inputs of the subsequent FI module. Here, we consider four kinds of FE methods (or network structures) to implement feature extraction: data concatenation (DC), nonlinear dimensionality reduction (NDR), convolution operation (Conv) and self-attention (SA).

For clarity, we first introduce matrix functions to mathematically formalize the operations of matrix vectorization, average pooling, convolution and nonlinear matrix products. Consider two matrices $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\tau] \in \mathbb{R}^{I\times\tau}$ with column vectors $\mathbf{x}_t = (x_{1t}, x_{2t}, \cdots, x_{It})^T$ ($t \in \{1,2,\cdots,\tau\}$) and $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_K] \in \mathbb{R}^{I\times K}$ with column vectors $\mathbf{w}_k = (w_{1k}, w_{2k}, \cdots, w_{Ik})^T$ ($k \in \{1,2,\cdots,K\}$).

(1) The vectorization of $\mathbf{X}$ is expressed as

$$\mathrm{vec}(\mathbf{X}) = \left(\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_\tau^T\right)^T \in \mathbb{R}^{\tau I}.$$

(2) The columnwise average pooling of $\mathbf{X}$ is expressed as

$$\mathrm{avp}_c(\mathbf{X}) = (\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_\tau)^T \in \mathbb{R}^\tau \tag{2}$$

with $\overline{x}_t = \frac{1}{I}\sum_{t=1}^{I} x_{it} (1\leqslant t\leqslant\tau)$.

(3) The convolution operation of $\mathbf{X}$ based on the convolution kernel $\mathbf{C} \in \mathbb{R}^{\omega\times I}$ with width ω and stride λ ($\omega, \lambda < \tau$) is expressed as

$$\mathbf{C}_\lambda(\mathbf{X}) = \left(\mathrm{tr}\left([\mathbf{X}]_{\mathscr{I}_0} \cdot \mathbf{C}\right), \mathrm{tr}\left([\mathbf{X}]_{\mathscr{I}_1} \cdot \mathbf{C}\right), \cdots, \mathrm{tr}\left([\mathbf{X}]_{\mathscr{I}_L} \cdot \mathbf{C}\right)\right)^T, \tag{3}$$

where $\mathscr{I}_l = \{1 + l\lambda, 2 + l\lambda, \cdots, \omega + l\lambda\}$ is the index set composed of t continuous indexes and $[\mathbf{X}]_{\mathscr{I}_l}$ stands for the matrix composed of the columns taken from matrix $\mathbf{X}$ w.r.t. the index set $\mathscr{I}_l$.

(4) The rectified linear unit (ReLU) function is denoted by $\mathrm{r}(x) := \max\{0, x\}$ ($x \in \mathbb{R}$), and the matrix-valued ReLU function w.r.t. the matrix product $\mathbf{X}^T\mathbf{W}$ is defined as

$$\mathbf{r}\left(\mathbf{X}^T\mathbf{W}\right) := \begin{bmatrix} \mathrm{r}\left(\mathbf{x}_1^T\mathbf{w}_1\right) & \mathrm{r}\left(\mathbf{x}_1^T\mathbf{w}_2\right) & \cdots & \mathrm{r}\left(\mathbf{x}_1^T\mathbf{w}_K\right) \\ \mathrm{r}\left(\mathbf{x}_2^T\mathbf{w}_1\right) & \mathrm{r}\left(\mathbf{x}_2^T\mathbf{w}_2\right) & \cdots & \mathrm{r}\left(\mathbf{x}_2^T\mathbf{w}_K\right) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{r}\left(\mathbf{x}_\tau^T\mathbf{w}_1\right)\mathrm{r}\left(\mathbf{x}_\tau^T\mathbf{w}_2\right)\cdots\mathrm{r}\left(\mathbf{x}_\tau^T\mathbf{w}_K\right) & & & \end{bmatrix}_{\tau\times K}.$$

(5) Given a vector $\mathbf{x} = (x_1, x_2, \cdots, x_N)^T \in \mathbb{R}^N$, the vector-valued softmax function $\overrightarrow{\sigma} : \mathbb{R}^N \to \mathbb{R}^N$ is defined as

$$\overrightarrow{\sigma}(\mathbf{x}) := (\sigma_1(\mathbf{x}), \sigma_2(\mathbf{x}), \cdots, \sigma_N(\mathbf{x}))^T,$$

where

$$\sigma_n(\mathbf{x}) = \frac{\mathrm{e}^{x_n}}{\sum\limits_{n=1}^{N} \mathrm{e}^{x_n}}.$$

Furthermore, given a matrix $\mathbf{X} = [x_{mn}]_{M\times N}$, the matrix-valued softmax function is defined as

$$\boldsymbol{\sigma}(\mathbf{X}) := \begin{bmatrix} \left(\sigma\left(\left(\mathbf{x}^{(1)}\right)^T\right)\right)^T \\ \left(\sigma\left(\left(\mathbf{x}^{(2)}\right)^T\right)\right)^T \\ \vdots \\ \left(\sigma\left(\left(\mathbf{x}^{(M)}\right)^T\right)\right)^T \end{bmatrix}_{M\times N},$$

where $\mathbf{x}^{(m)}$ is the m-th row vector of matrix $\mathbf{X}$.

(6) The sigmoid function is defined as $\mathrm{sig}(x) := \frac{1}{1+\mathrm{e}^{-x}}$ $(x \in \mathbb{R})$. Given a matrix $\mathbf{X} = [x_{mn}]_{M\times N}$, the matrix-valued sigmoid function is written as

$$\mathbf{sig}(\mathbf{X}) := \begin{bmatrix} \mathrm{sig}(\mathbf{x}_{11}) & \mathrm{sig}(\mathbf{x}_{12}) & \cdots & \mathrm{sig}(\mathbf{x}_{1n}) \\ \mathrm{sig}(\mathbf{x}_{21}) & \mathrm{sig}(\mathbf{x}_{22}) & \cdots & \mathrm{sig}(\mathbf{x}_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{sig}(\mathbf{x}_{m1})\mathrm{sig}(\mathbf{x}_{m2})\cdots\mathrm{sig}(\mathbf{x}_{mn}) & & & \end{bmatrix}_{M\times N}.$$

Based on these notations, we then formalize the four kinds of FE methods (or network structures) adopted in the FE module. Given a sample $\mathbf{X}_n$, let $\mathbf{a}_n := \mathbf{a}(\mathbf{X}_n)$ be the corresponding output of the FE module,

- *Data Concatenation* (DC): This method directly concatenates the columns of the matrix $\mathbf{X}_n$ into one vector, which is also treated as an output of the FE module; *i.e.,* $\mathbf{a}_n = \mathrm{vec}(\mathbf{X}_n)$.
- *Nonlinear Dimensionality Reduction* (NDR): Given an input sample $\mathbf{X}_n$, we first introduce the weight matrix $\mathbf{W}$ and the matrix-valued ReLU function $\mathbf{r}(\cdot)$ to implement a nonlinear matrix product and then use columnwise average pooling to achieve a τ-dimensional output $\mathbf{a}_n$ $(1\leqslant n\leqslant N)$; *i.e.,*

$$\mathbf{a}_n = \mathrm{avp}_c\left(\mathbf{r}\left(\mathbf{X}^T\mathbf{W}\right)\right).$$

- *Convolution* (Conv): Given K convolution kernels $\mathbf{C}^{(k)} \in \mathbb{R}^{\omega\times I}$ with ω-width $(1\leqslant k\leqslant K)$, we can use the convolution operation to convert the sample $\mathbf{X}_n$ into a K-dimensional output of the FE module; *i.e.,*

$$\mathbf{a}_n = \mathrm{avp}_c\left(\left[\mathbf{C}^{(1)}_\lambda(\mathbf{X}_n), \cdots, \mathbf{C}^{(K)}_\lambda(\mathbf{X}_n)\right]\right).$$

- *Self-attention* (SA) [40]: Given a sample $\mathbf{X}_n$, set $\mathbf{K}_n = \mathbf{Q}_n = \mathbf{V}_n = \mathbf{X}_n$. By introducing the weight matrices $\mathbf{W}^{(h)}_K, \mathbf{W}^{(h)}_Q, \mathbf{W}^{(h)}_V \in \mathbb{R}^{I\times K}$ $(1\leqslant h\leqslant H)$, we compute

$$\widehat{\mathbf{K}}^{(h)}_n = \mathbf{ReLU}\left(\mathbf{X}^T_n\mathbf{W}^{(h)}_K\right) \in \mathbb{R}^{\tau\times K}; \widehat{\mathbf{Q}}^{(h)}_n = \mathbf{ReLU}\left(\mathbf{X}^T_n\mathbf{W}^{(h)}_Q\right) \in \mathbb{R}^{\tau\times K}; \widehat{\mathbf{V}}^{(h)}_n = \mathbf{ReLU}\left(\mathbf{X}^T_n\mathbf{W}^{(h)}_V\right) \in \mathbb{R}^{\tau\times K},$$

and then obtain H attention matrices in the following way:

$$\mathbf{Z}^{(h)}_n := \boldsymbol{\sigma}\left(\frac{\widehat{\mathbf{Q}}^{(h)}_n \cdot \left(\widehat{\mathbf{K}}^{(h)}_n\right)^T}{\sqrt{d_k}}\right) \cdot \widehat{\mathbf{V}}^{(h)}_n \in \mathbb{R}^{\tau\times K}, \quad 1\leqslant h\leqslant H.$$

Subsequently, we form the attention tensor

$$\mathbf{Z}_n := \left[\mathbf{Z}^{(1)}_n, \mathbf{Z}^{(2)}_n, \cdots, \mathbf{Z}^{(H)}_n\right]_{\tau\times K\times H}$$

and then concatenate the 2nd and the 3rd dimensions of the tensor $\mathbf{Z}_n$ to form a matrix

$$\widehat{\mathbf{Z}}_n := \left[\widehat{\mathbf{z}}^T_1, \widehat{\mathbf{z}}^T_2, \cdots, \widehat{\mathbf{z}}^T_\tau\right]^T \in \mathbb{R}^{\tau\times KH}$$

with $\widehat{\mathbf{z}}_t = \mathrm{vec}([\widehat{\mathbf{Z}}_n]_{t,:,:})$, where $[\widehat{\mathbf{Z}}_n]_{t,:,:}$ stands for the matrix corresponding to the t-th row of the tensor $\widehat{\mathbf{Z}}_n$. Finally, the output of the FE module is expressed as

$$\mathbf{a}_n = \mathrm{avp}_c\left(\widehat{\mathbf{Z}}_n\right) \in \mathbb{R}^\tau.$$

3.1.2. Feature integration (FI) module

The FI module is a four-layer fully-connected neural network, where the input layer is actually the output layer of the FE module; and the last three layers are of the 20-7-11 structure, which is in accordance with the findings in [27]. Specifically, the 7 nodes in the second hidden layer correspond to the 7 kinds of physical–mechanical indicators that are related to the TBM's excavation process, and the 11 nodes in the output layer are squashed by the softmax function to signify the thickness of each of 11 kinds of rock-soil types.

Given an input $\mathbf{X}_n$, the corresponding output of the FI module is denoted by $\mathbf{s}_n := \mathrm{G}(\mathbf{X}_n)$, which is also the output of the generator G. Moreover, the weights of the generator G are denoted by $\mathbf{W}_\mathrm{G}$.

3.2. Discriminator of the GAN-GP

The discriminator, denoted by D, aims to determine whether its inputs are real geological data. If the discriminator is trained to be incapable of distinguishing them correctly after adversarial training, the output of the generator is deemed an accurate estimate of the true geological condition. Similar to the original GAN, the discriminator, which partially inherits the structural characteristics of the FI module, is also a four-layer fully-connected neural network with an 11–7-20–1 structure, where the input layer has 11 nodes that correspond to the thicknesses of the 11 rock-soil types and the single output node indicates whether the input is a real geological datum or an output of the generator. The activation functions between the input layer and hidden layers are ReLU functions, and the output node is squashed by the '0–1' sigmoid function. We note that the similarity between the structure of the discriminator and that of the FI module can improve the training stability of the proposed GAN-GP. Moreover, the weights of the generator D are denoted by $\mathbf{W}_\mathrm{D}$.

3.3. Adversarial training strategy of the GAN-GP

There are two kinds of loss functions associated with the generator of the GAN-GP: the generator loss and the teacher loss. As suggested by the ordinary GAN, the generator loss should be defined as

$$\mathscr{L}'_\mathrm{G} := \mathbb{E}_\mathbf{y}\{\log(\mathrm{D}(\mathbf{y}))\} + \mathbb{E}_\mathbf{X}\{\log(1-\mathrm{D}(\mathrm{G}(\mathbf{X})))\}, \tag{4}$$

where the expectation $\mathbb{E}_\mathbf{y}$ (resp. $\mathbb{E}_\mathbf{X}$) is taken on the real geological (resp. TBM operational) data. Since the generator loss is minimized to update the generator weights $\mathbf{W}_G$ rather than the discriminator weights $\mathbf{W}_\mathrm{D}$, the generator loss (4) can be simplified as

$$\mathscr{L}_\mathrm{G} := \mathbb{E}_\mathbf{X}\{\log(1-\mathrm{D}(\mathrm{G}(\mathbf{X})))\}. \tag{5}$$

The discriminator loss is expressed as

$$\mathscr{L}_\mathrm{D} := \mathbb{E}_\mathbf{y}\{\log(1-\mathrm{D}(\mathbf{y}))\} + \mathbb{E}_\mathbf{X}\{\log(\mathrm{D}(\mathrm{G}(\mathbf{X})))\}. \tag{6}$$

The minimization of (6) makes the generator provide fake geological data that are identical to the real geological data so that the discriminator is cheated.

The teacher loss is expressed as

$$\mathscr{L}_\mathrm{T} := \mathbb{E}_{(\mathbf{X},\mathbf{y})}\left\{[\mathrm{G}(\mathbf{X})-\mathbf{y}]^2\right\}, \tag{7}$$

where $\mathbb{E}_{(\mathbf{X},\mathbf{y})}$ stands for the expectation taken on the pair of TBM operational and real geological data. The teacher loss $\mathscr{L}_\mathrm{T}$ will be used to pre-train the generator with the following purposes: 1) model adjustment - to avoid the phenomenon that the outputs of the generator match the distribution of real geological data, while the distribution of corresponding inputs is far away from that of real operational data, and 2) model stability - to avoid the corruption of the model and accelerate the training speed.

In Alg. 1, we sketch the workflow of the adversarial training strategy for GAN-GP. Different from the standard GAN's adversarial training strategy, the training strategy of the GAN-GP contains two particular tricks: one is the pre-training trick, which uses the training samples to pre-train the generator before the adversarial training; and the other is the teacher-loss trick, which minimizes the teacher loss $\mathscr{L}_T$ to refine the generator weights after minimizing the generator loss $\mathscr{L}_G$ and the discriminator loss $\mathscr{L}_D$ during each iteration of the adversarial training.

Algorithm 1: Adversarial Learning Strategy for the GAN-GP

```
Input: ${(\mathbf{X}_n,\mathbf{y}_n)}_{n=1}^N$, $\mathbf{W}_\mathrm{G}^{(0,0)}$, $\mathbf{W}_\mathrm{D}^{(1)}$, $J$, $K$, $\eta_0$, $\eta_\mathrm{G}$, $\eta_\mathrm{D}$, $\eta_\mathrm{T}$;
Ouput: $\mathbf{W}_\mathrm{G}$, $\mathbf{W}_\mathrm{D}$;
forall $j = 1,2,\cdots,J$; do
    (Pre-training) Update $\mathbf{W}_\mathrm{G}$ by minimizing the teacher loss $\mathscr{L}_\mathrm{T}$: $\mathbf{W}_\mathrm{G}^{(0,j)} = \mathbf{W}_\mathrm{G}^{(0,j-1)} - \eta_0 \cdot \frac{\partial\mathscr{L}_\mathrm{T}}{\partial\mathbf{W}_\mathrm{G}}$;
end for
Set $\mathbf{W}_\mathrm{G}^{(1)} = \mathbf{W}_\mathrm{G}^{(0,J)}$;
for all $k = 1,2,\cdots,K$; do
    Update $\mathbf{W}_\mathrm{G}$ and $\mathbf{W}_\mathrm{D}$ by minimizing the discriminator loss $\mathscr{L}_\mathrm{D}$: $\mathbf{W}_\mathrm{G}^{(k,1)} = \mathbf{W}_\mathrm{G}^{(k)} - \eta_\mathrm{G} \cdot \frac{\partial\mathscr{L}_\mathrm{D}}{\partial\mathbf{W}_\mathrm{HF}}$ and $\mathbf{W}_\mathrm{D}^{(k,1)} = \mathbf{W}_\mathrm{D}^{(k)} - \eta_\mathrm{D} \cdot \frac{\partial\mathscr{L}_\mathrm{D}}{\partial\mathbf{W}_\mathrm{D}}$;
```

(continued on next page)

(*continued*)

Algorithm 1: Adversarial Learning Strategy for the GAN-GP

```
8:          Update $\mathbf{W}_{\mathrm{G}}$ and $\mathbf{W}_{\mathrm{D}}$ by minimizing the generator loss $\mathscr{L}_{\mathrm{G}}$: $\mathbf{W}_{\mathrm{G}}^{(k,3)} = \mathbf{W}_{\mathrm{G}}^{(k,2)} - \eta_{\mathrm{G}} \cdot \frac{\partial \mathscr{L}_{\mathrm{G}}}{\partial \mathbf{W}_{\mathrm{G}}}$ and $\mathbf{W}_{\mathrm{D}}^{(k+1)} = \mathbf{W}_{\mathrm{D}}^{(k,1)} - \eta_{\mathrm{D}} \cdot \frac{\partial \mathscr{L}_{\mathrm{G}}}{\partial \mathbf{W}_{\mathrm{D}}}$;
            Update $\mathbf{W}_{\mathrm{G}}$ by minimizing the teacher loss $\mathscr{L}_{\mathrm{T}}$: $\mathbf{W}_{\mathrm{G}}^{(k+1)} = \mathbf{W}_{\mathrm{G}}^{(k,3)} - \eta_{\mathrm{T}} \cdot \frac{\partial \mathscr{L}_{\mathrm{T}}}{\partial \mathbf{W}_{\mathrm{G}}}$;
9:      end for
10:     Set $\mathbf{W}_{\mathrm{G}} = \mathbf{W}_{\mathrm{G}}^{(K+1)}$ and $\mathbf{W}_{\mathrm{D}} = \mathbf{W}_{\mathrm{D}}^{(K+1)}$.
```

4. Numerical experiments

In this section, we conduct numerical experiments to explore the following concerns: 1) the performance of the GAN-GP in the geological prediction task; 2) the influence on the prediction performance caused by different FE methods adopted in the FE module; 3) the influence on the prediction performance caused by different choices of step size τ; 4) the feasibility of the proposed adversarial training strategy; 5) the effectiveness of the pre-training trick; and 6) the effectiveness of the teacher-loss trick. All the experiments were performed by using TensorFlow on a computer equipped with an Intel® i7-6700 K CPU at 4.0 GHz × 8, 64 GB RAM and two Nvidia® GTX-1080 graphic cards.

4.1. Experimental setting

In the experiments, we consider the varying step sizes $\tau \in \{5, 10, 15, 20, 25\}$ and the corresponding sizes N_τ of the samples labeled with the thickness of each rock-soil type (*cf.* Table 1). For each τ, we split the N_τ samples into two parts: 80% of the samples are for training, and the rest of the samples are for testing. We note that the two parts are not randomly selected from the original samples because this manner will damage the sequential information encoded in them. The prediction performance is measured by using the smallest (top-5) test loss out of five repeated experiments:

$$\text{test loss} := \frac{1}{J}\sum_{j=1}^{J}\left|\mathbf{y}_j - \widehat{\mathbf{y}}_j(\mathbf{X}_j)\right|,$$

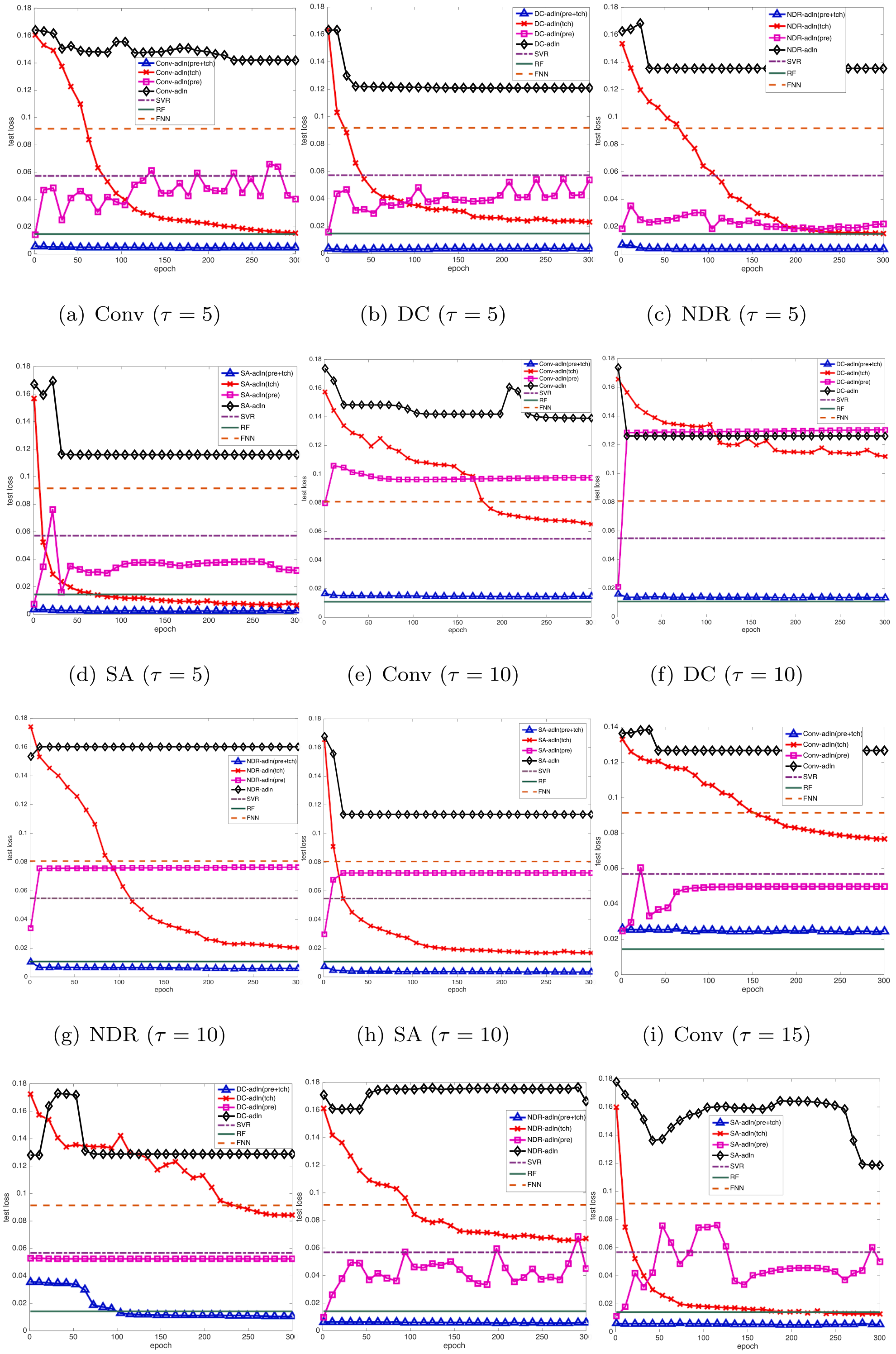
where $\mathbf{y}_j$ and $\widehat{\mathbf{y}}_j(\mathbf{X}_j)$ are the real output and the geological prediction corresponding to the input $\mathbf{X}_j$, respectively.

To validate the adversarial training strategy for the GAN-GP given in Alg. 1, denoted by 'adln(pre + tch)', we also consider three more strategies in the experiments: 1) the adversarial training without the pre-training trick, denoted by 'adln(tch)'; 2) the adversarial training without the teacher-loss trick, denoted by 'adln(pre)'; and 3) the adversarial training without the pre-training and the teacher-loss tricks, denoted by 'adln'.

In the pre-training phase of 'adln(pre + tch)' (steps 1 ~ 4 of Alg. 1, we minimize the teacher loss by using the Adam method with a learning rate of $\eta_0 = 0.005$ and a mini-batch size of 32. The training will stop after $J = 200$ iterations. In the adversarial phase of 'adln

Table 1
Top-5 test losses of the different prediction models with varying step sizes $\tau \in \{5, 10, 15, 20, 25\}$.

Model	FE Method	$\tau = 5$	$\tau = 10$	$\tau = 15$	$\tau = 20$	$\tau = 25$
		($N_\tau = 12325$)	($N_\tau = 6141$)	($N_\tau = 4080$)	($N_\tau = 3046$)	($N_\tau = 2431$)
adln	DC	**0.1210**	0.1261	0.1288	0.1325	0.1471
	Conv	0.1420	0.1390	**0.1267**	0.1641	0.1308
	NDR	0.1354	0.1602	0.1315	0.1425	**0.1190**
	SA	0.1160	**0.1134**	0.1186	0.1659	0.1147
adln(pre)	DC	0.0528	0.1302	**0.0526**	0.0932	0.0844
	Conv	**0.0404**	0.0974	0.0499	0.0851	0.0895
	NDR	**0.0217**	0.0764	0.0470	0.0925	0.0739
	SA	**0.0317**	0.0727	0.0485	0.0923	0.0776
adln(tch)	DC	**0.0239**	0.1117	0.0841	0.1029	0.1049
	Conv	**0.0154**	0.0649	0.0766	0.0789	0.1103
	NDR	**0.0148**	0.0204	0.0442	0.0726	0.0859
	SA	**0.0062**	0.0170	0.0135	0.0266	0.0180
adln(pre + tch)	DC	**0.0038**	0.0135	0.0110	0.0319	0.0526
	Conv	**0.0046**	0.0147	0.0246	0.0254	0.0950
	NDR	**0.0037**	0.0060	0.0062	0.0182	0.0188
	SA	0.0036	**0.0035**	0.0055	0.0105	0.0146
FNN	–	**0.0737**	0.0807	0.0915	0.1037	0.1160
SVR	–	**0.0526**	0.0548	0.0570	0.1482	0.1554
RF	–	**0.0074**	0.0108	0.0145	0.0182	0.0202



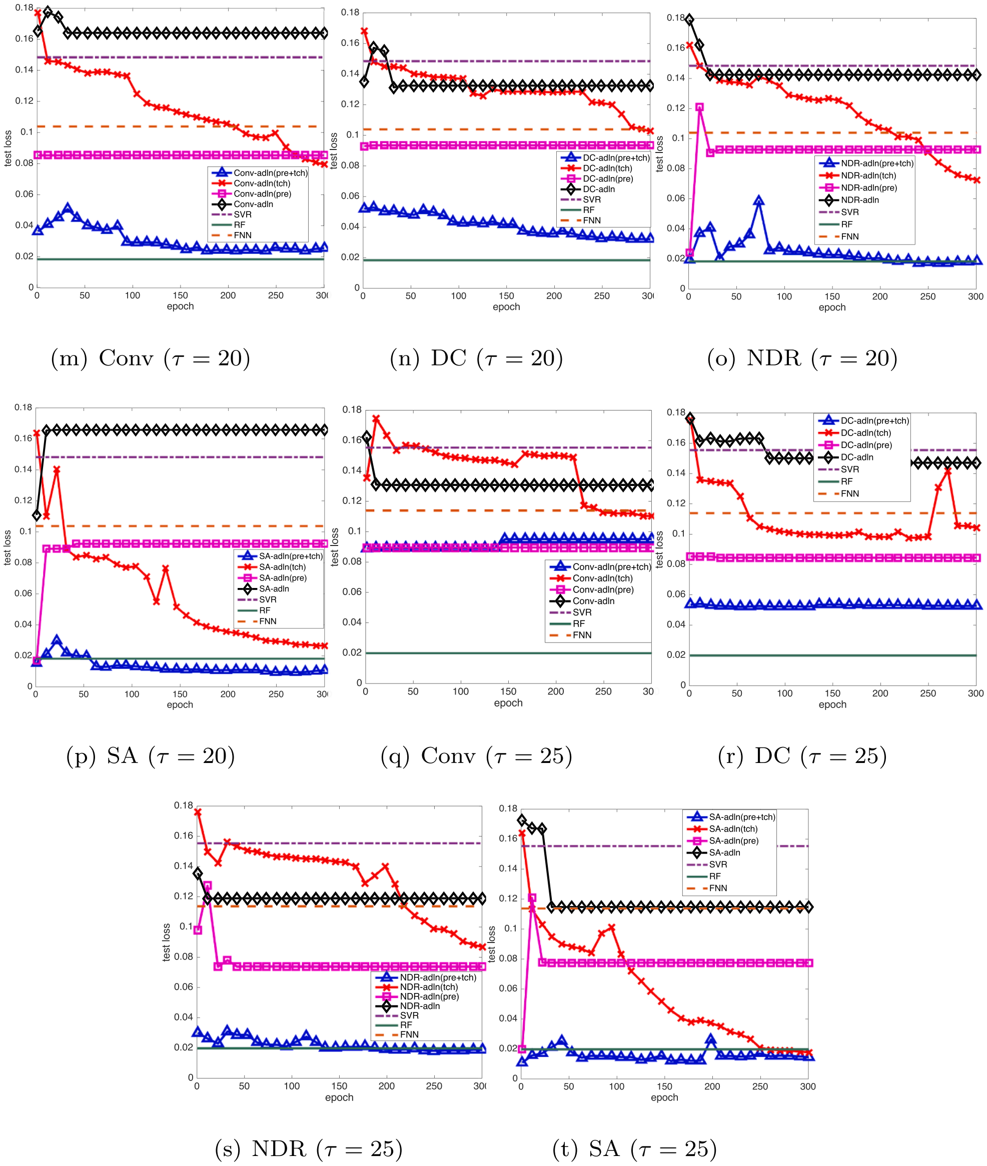
(a) Conv ($\tau = 5$) (b) DC ($\tau = 5$) (c) NDR ($\tau = 5$)

(d) SA ($\tau = 5$) (e) Conv ($\tau = 10$) (f) DC ($\tau = 10$)

(g) NDR ($\tau = 10$) (h) SA ($\tau = 10$) (i) Conv ($\tau = 15$)

(j) DC ($\tau = 15$) (k) NDR ($\tau = 15$) (l) SA ($\tau = 15$)

Fig. 3. The test losses of the different models for predicting the rock-soil types.



(m) Conv ($\tau = 20$)

(n) DC ($\tau = 20$)

(o) NDR ($\tau = 20$)

(p) SA ($\tau = 20$)

(q) Conv ($\tau = 25$)

(r) DC ($\tau = 25$)

(s) NDR ($\tau = 25$)

(t) SA ($\tau = 25$)

Fig. 3. (*continued*).

(pre + tch)' (steps 5 $\sim$ 10 of Alg. 1, we minimize the generator loss, the discriminator loss and the teacher loss by using the Adam method with a mini-batch size of 88. The number of iterations K equals 300 and the learning rates are set to $\eta_G = 0.0001$, $\eta_D = 0.0005$ and $\eta_T = 0.001$. Following the convention in deep learning literature, these hyper-parameters of GAN-GP are set according to the empirical observations in the numerical experiments.

Remark 1. The following is the reason why the learning rate η_D for the discriminator loss is larger than the learning rates η_G and η_T for the generator loss and the teacher loss, respectively. In each iteration, the generator weights $\mathbf{W}_G$ will be updated twice (by minimizing the generator loss and the teacher loss, respectively) but the discriminator weights $\mathbf{W}_D$ can only be updated once. Therefore, setting a larger learning rate η_D for the discriminator loss aims to balance the behaviors of the generator and the discriminator. In this manner, we prevent the phenomenon that although the generator has been trained enough to produce an accuracy approximation to the real data, the current discriminator cannot distinguish the outputs of the generator from the real data, and the gradient of discriminator loss may vanish in advance.

As a comparison, we also apply three state-of-the-art models to handle the prediction task: support vector regression (SVR) with Gaussian kernels, feedforward neural network (FNN) and random forest (RF). The SVR, FNN and RF are the most representative learning models and have become the benchmark methods in many practical problems. Thus, setting them as references can make the performance examination more convincing.

- FNN, a representative connectionist method, models the connections among the neurons to imitate the complex functions of the human brain. It has played an essential part in the geological condition prediction works [19,22–24].
- SVR, a variant of support vector machine (SVM) that was originally designed for classification, approximates complicated nonlinear functions by using a linear combination of kernel functions that map the original data into the high-dimensional (or infinite-dimensional) feature space. SVR has successfully afforded geological condition prediction tasks in some specific settings [17,21]
- RF, a representative symbolism method, learns rules from the training data and then uses logical reasoning to produce the response of future inputs. Its superiority in geological condition prediction has been discussed in [41,42,25].

The three models are implemented by using the 'scikit-learn' toolbox in Python 3.7, and the following summarizes their main hyper-parameter settings.

- Based on 'sklearn.neural_network.MLPRegressor', the FNN has two hidden layers with 20 and 7 nodes, and all the activation functions are sigmoid functions except that the output nodes are squashed by using linear functions. This structure is inspired by the research findings of [27], where the 7 hidden nodes in the second hidden layer correspond to the 7 kinds of physical–mechanical indicators that are related to the TBM excavation process. Adam is chosen as the solver for weight optimization; the L_2 penalty parameter is 0.001; the learning rate is 0.001; the maximum number of iterations is 200; the exponential decay rate for the estimates of the first (resp. second) moment vector in Adam is 0.9 (resp. 0.999); the value for numerical stability in Adam is 10^{-8}; and the samples will be shuffled in each iteration.
- SVR with a Gaussian kernel is implemented by using 'sklearn.svm.SVR'. Through *five*-fold cross-validation with grid search, the kernel coefficient is 1/11, the regularization parameter is 1, and the tolerance for the stopping criterion is 0.0001.
- The RF is processed via 'sklearn.ensemble.RandomForestRegressor' and the function to measure the quality of a split is set as the mean squared error (MSE). Similarly, *five*-fold cross-validation with grid search leads to the following hyper-parameters: the maximum depth of the tree is 8; the minimum number of samples required to split an internal node is 2; the number of trees in the forest is 100; the minimum number of samples required to be at a leaf node is 50; and the number of features considered when looking for the best split is 11.

Moreover, we use principal component analysis (PCA) to reduce the dimension of training inputs to 34 while maintaining 95% variance.

4.2. Experimental results and discussion

As shown in Fig. 3 and Table 1, the GAN-GP with SA in the FE module achieves the best prediction performance among all the models. Moreover, the GAN-GP with NDR and the RF model also perform well but worse than the GAN-GP with SA in most cases. The teacher-loss trick plays an essential role in the adversarial training of the GAN-GP, and training without the teacher-loss trick fails regardless of what FE method is adopted in the FE module. Moreover, the pre-training trick provides reasonable initial weights for training the GAN-GP, and thus significantly accelerates the training process. In contrast, training without the pre-training trick requires a large number of epochs, making it easier for the model to be trapped by local minima. In addition, a small step size τ is more suitable to developing GAN-GPs for the prediction task than a large one, which implies that the sample size plays a more important role in GAN-GP training than the dissimilarity among samples. Namely, the GAN-GP training places a relatively high demand on the quantity of the samples.

The contest between the generator and the discriminator of the GAN-GP during adversarial training makes the most of the limited amount of labeled training data and yields a better performance than the conventional FNN with the same structure as the generator of the GAN-GP (*i.e.,* the GAN-GP trained only through the pre-training progress). As shown in Alg. 1, during each iteration of the adversarial training process, the minimization of the discriminator loss (6) simultaneously updates the discriminator and the generator weights to fool the discriminator, making it incapable of distinguishing a generator output or a real instance, and the subsequent minimization of the generator loss (5) refines the generator weight to make the generator outputs describe the geological condition information as exactly as possible. Such adversarial training updates the generator weights twice in one iteration from different standpoints: one is to make the distribution of the generator outputs coincide with the real distribution of rock-soil types in the tunnel; and the other is to find an accurate mapping relation between the TBM operational data and the thicknesses of the rock-soil types. In contrast, the traditional training for the FNN considers only the latter, which is the reason why the GAN-GP with an adversarial training strategy outperforms the conventional FNN. However, such an adversarial structure significantly increases the training difficulty of the GAN-GP. To improve the training efficiency and stability, we then introduce the pre-training trick and the teacher-loss trick: the former is used to find reasonable initial generator weights before adversarial training, and the latter is used to balance the individual training progress of the generator and the discriminator during adversarial training. The experimental results, listed in Table 1, demonstrate that the teacher-loss trick plays a crucial role in training the GAN-GP, and the introduction of the teacher-loss trick can significantly improve the GAN-GP's prediction performance.

Remark 2. Recalling (5) and (6), the minimization of the generator loss $\mathscr{L}_{\mathrm{G}}$ and the discriminator loss $\mathscr{L}_{\mathrm{D}}$ (called the adversarial training phase) requires the competition between the generator and the discriminator in each iteration, where the generator and the discriminator weights will be updated simultaneously. Specifically, the function of the teacher-loss trick lies in the following two

Table 2
Computational costs of the different predictors with varying step sizes $\tau \in \{5, 10, 15, 20, 25\}$.

Model	FE Method	$\tau = 5$ ($N_\tau = 12325$)			$\tau = 10$ ($N_\tau = 6141$)			$\tau = 15$ ($N_\tau = 4080$)			$\tau = 20$ ($N_\tau = 3046$)			$\tau = 25$ ($N_\tau = 2431$)		
		Pre-tr (s)	Tr (s)	Te (s)	Pre-tr (s)	Tr (s)	Te (s)	Pre-tr (s)	Tr (s)	Te (s)	Pre-tr (s)	Tr (s)	Te (s)	Pre-tr (s)	Tr (s)	Te (s)
adln	DC	–	230.7370	0.0040	–	145.3394	0.0040	–	94.5552	0.0030	–	107.7807	0.0040	–	77.3644	0.0040
	Conv	–	252.3833	0.0030	–	166.1697	0.0040	–	113.8426	0.0030	–	121.8065	0.0050	–	83.1160	0.0040
	NDR	–	332.1682	0.0050	–	199.9912	0.0040	–	142.6855	0.0050	–	135.9410	0.0050	–	102.3887	0.0050
	SA	–	475.7299	0.0080	–	309.9502	0.0070	–	194.5308	0.0060	–	183.8108	0.0060	–	144.3189	0.0070
adln(pre)	DC	72.2368	137.8235	0.0030	37.9525	81.4931	0.0050	25.4509	58.4038	0.0040	26.2945	59.7040	0.0040	19.0421	47.1061	0.0030
	Conv	88.7587	152.3756	0.0040	46.5376	86.2992	0.0040	30.3857	64.2203	0.0040	35.8327	67.3398	0.0070	23.4009	51.6554	0.0040
	NDR	108.0501	200.9447	0.0040	54.9132	108.0501	0.0040	36.3179	77.6823	0.0050	37.2883	83.4429	0.0040	26.1578	60.4633	0.0040
	SA	179.5978	292.5856	0.0080	91.8683	158.4683	0.0060	61.0298	116.0856	0.0060	63.8103	119.2858	0.0060	43.1361	85.9866	0.0060
adln(tch)	DC	–	255.6933	0.0040	–	147.9329	0.0040	–	105.3682	0.0040	–	94.8773	0.0040	–	83.4488	0.0040
	Conv	–	298.1441	0.0040	–	169.2785	0.0040	–	120.9925	0.0040	–	115.5000	0.0050	–	89.3630	0.0040
	NDR	–	375.6394	0.0040	–	225.3659	0.0040	–	143.4494	0.0050	–	130.9231	0.0050	–	107.2096	0.0040
	SA	–	565.5986	0.0080	–	313.0442	0.0070	–	221.8777	0.0060	–	193.6105	0.0070	–	162.6489	0.0060
adln(pre + tch)	DC	71.3023	154.4949	0.0040	39.4786	90.2387	0.0040	26.4612	64.0637	0.0040	25.4315	76.0675	0.0050	18.0577	48.5013	0.0040
	Conv	85.1004	181.1055	0.0030	47.3286	101.0634	0.0040	32.4452	73.3628	0.0040	35.4921	71.4229	0.0040	21.8515	53.7931	0.0040
	NDR	106.3835	229.3912	0.0050	56.1418	123.9454	0.0040	37.5237	85.5323	0.0040	33.6604	95.5394	0.0050	26.0044	63.4054	0.0040
	SA	178.3351	346.7026	0.0080	93.1430	186.7497	0.0060	63.2080	132.9944	0.0060	65.7298	120.0100	0.0070	41.6775	96.3892	0.0060
FNN	–	–	0.8866	0.0010	–	0.7819	0.0010	–	0.8647	0.0010	–	0.9235	0.0010	–	0.6722	0.0010
SVR	–	–	1.0981	0.1237	–	1.8720	0.1875	–	1.2467	0.1277	–	0.8198	0.0858	–	0.5785	0.0636
RF	–	–	3.8308	0.0110	–	5.7721	0.0130	–	3.8527	0.0100	–	2.8723	0.0110	–	2.2251	0.0120

[1] 'Pre-tr', 'Tr' and 'Te' are the abbreviations for the words 'pre-training', 'training' and 'testing', respectively.

aspects:

- One is to overcome the lack of real geological data. In the adversarial training phase, the discriminator rectifies the generator by using the real geological data, but such a rectification actually has a high demand of the geological-data quality. Unfortunately, the current dataset only contains 88 geological sampling points, which may not be enough to support a desired rectification. Alternatively, after the adversarial training phase of each training iteration, the teacher-loss trick sets the updated generator weights as the initial weights, and employs the TBM operational data labeled with the thickness of each rock-soil type to minimize the teacher loss $\mathscr{L}_{\mathrm{T}}$ (*cf.* (7)). In this manner, the mapping relationship between the TBM operational data and the tunnel geological condition is used to improve the rectification results obtained in the adversarial training phase.
- The other is to stabilize the training process against the uncertainty arising from the Adam method with mini-batch, a variant of stochastic gradient method. In the adversarial training phase, there is a strong interaction between the generator and the discriminator weights. The uncertainty arising from the Adam method is likely to make the discriminator weights updated in a wrong direction, and then to have a negative influence on the update of generator weights. The teacher-loss trick is helpful to pull the generator weights back to a reasonable optimization direction in each training iteration.

In addition, we compare the computational costs of the GAN-GP with different training tricks and the state-of-the-art models (FNN, SVR and RF models). The experiment has the same parameter settings as above. As shown in Table 2, the training cost of the GAN-GP is significantly higher than that of the state-of-the-art models because of the complex network structure and training strategy, while its testing cost is comparable to that of the state-of-the-art models. Interestingly, we also find that the total pre-training cost and training cost in ‘adln (pre)’ (resp. ‘adln (pre + tch)’) is lower than the individual training cost in ‘adln’ (resp. ‘adln (tch)’). This experimental result demonstrates the important role of the pre-training trick in GAN-GP adversarial training.

In summary, the advantages of high prediction accuracy, low demand for labeled samples and reasonable test cost guarantee the high applicability of the proposed GAN-GP in practical tunnel geological condition prediction tasks.

5. Conclusion

In this paper, we propose a generative adversarial network for geological prediction (GAN-GP) to accurately estimate the thickness of each rock-soil type in a tunnel boring machine (TBM) construction tunnel based on the operational data collected from sensors on the TBMs. In contrast with massive time-continuous TBM operational data, only a small part of the data are labeled with geological information obtained at discrete locations by using the drilling method. To overcome the imbalance between the sizes of the labeled and unlabeled operational data, we first label the operational data within 0.3 m around a drilling point with the same geological condition as the drilling sample, and then concatenate every τ continuous operational data to form a new datum. In this manner, we increase the size of operational data labeled with geological information but introduce redundancy into the new dataset.

In this paper, we propose a generative adversarial network for geological prediction (GAN-GP) of a TBM construction tunnel based on operational data collected from sensors on a TBM. The proposed GAN-GP aims to accurately estimate the thickness of each rock-soil type at an arbitrary location in the TBM construction tunnel.

To deal with such a dataset, we design the generator of the GAN-GP composed of two modules: the feature extraction (FE) module extracts the important features from the operational data as the inputs of the FI module, and the feature integration (FI) module estimates the thickness of each rock-soil type at a location. We adopt four kinds of FE methods: data concatenation (DC), nonlinear dimensionality reduction (NDR), one-dimensional convolution operation (Conv) and self-attention (SA). The discriminator of the GAN-GP aims to determine whether the outputs of the FI module are real geological data. After adversarial training, if the trained discriminator fails, the outputs of the FI module will provide an accurate estimation of the geological condition. Moreover, we also introduce the pre-training and the teacher-loss tricks to enhance the stability and efficiency of adversarial training for the GAN-GP.

The experimental results validate the proposed GAN-GP, and show that 1) the GAN-GP with SA in the FE module outperforms the other models in the geological prediction task; 2) both the pre-training and the teacher-loss tricks play essential roles during the adversarial training, and the absence of either of them will cause the adversarial training to fail; 3) a small step size τ is more suitable for developing the models for geological prediction, which implies that the sample size is a key factor in the adversarial training of GAN-GP; and 4) the testing cost of the GAN-GP is comparable to those of the state-of-the-art models (*i.e.,* the FNN, SVR and RF models) despite a relatively higher training cost that is due to the complex network structure and training strategy.

Compared with the state-of-the-art predictors, the superiority of the GAN-GP benefits from its specific structure that fully takes into account the inherent characteristics of the geological condition prediction task. First, there is an imbalance between the relatively few TBM operational data labeled with geological information and massive unlabeled data. The adversarial training strategy implemented between the generator and the discriminator of the GAN-GP aims to exhaustively exploit the rock-soil type classification information from the labeled data. Second, the TBM operational data are time-dependent and contain redundancies among adjacent time points. The step size τ in the FE module is set to identify these redundancies, and the FE methods (including DC, NDR, Conv and SA) are adopted to capture the sequential characteristics of the TBM operational data. Such a network structure can also be generalized to other engineering prediction tasks that contain time-series inputs and relatively few labeled training samples. More importantly, we introduce two training tricks to improve the training efficiency and stability: the pre-training trick and the teacher-loss trick. The former finds desirable initial generator weights before adversarial training, and the latter balances the individual training progress of the generator and the discriminator during adversarial training. It is worth pointing out that the teacher-loss trick plays a crucial role in the GAN-GP’s training process (*cf.* Remark 2). Benefiting from the two tricks, the proposed GAN-GP outperforms neural networks with the same structure.

In future works, we will consider active learning methods to adaptively select drilling-sampling positions and to improve the geological prediction performance.

CRediT authorship contribution statement

Chao Zhang: Conceptualization, Writing - original draft, Supervision, Writing - review & editing. **Minming Liang:** Methodology, Software. **Xueguan Song:** Project administration, Funding acquisition. **Lixue Liu:** Investigation, Formal analysis. **Hao Wang:** Validation. **Wensheng Li:** Visualization. **Maolin Shi:** Data curation.

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

We are grateful to the anonymous reviewers and the editors for their valuable comments and suggestions. This research is supported by the National Key Research and Development Program of China (No. 2018YFB1702502), the National Natural Science Foundation of China (No. 52075068) and the Fundamental Research Funds for the Central Universities (No. DUT20LK38).

Appendix

Table A.3
Physical–mechanical indicators of the different rock-soil types [27].

	Type					
Indicator	②$_{3}$	②$_{4}$	⑨$_{1}$	⑨$_{2-1}$	⑨$_{2-2}$	⑨$_{3}$
Y (kN/m^3)	17.000	18.5	19.50	20.500	22.50	24.50
FI (°)	4.500	18.0	25.00	27.500	45.00	55.00
EM (MPa)	4.000	5.5	40.000	90.000	10000	10000
P	0.40	0.35	0.250	0.250	0.25	0.22
SITA	0.650	0.5	0.00	0.00	0.00	0.00
K (m/d)	0.003	3.5	0.80	2.5	15.00	1.50
FRB (kPa)	10.000	20.0	45.000	60.000	125.00	380.00

	Type						
Indicator	④$_{2}$	④$_{4}$	④$_{5}$	④$_{8}$	④$_{9}$	④$_{10}$	④$_{11}$
Y (kN/m^3)	19.000	18.000	19.00	19.50	20.00	20.50	21.00
FI (°)	15.000	8.000	20.00	26.00	28.00	32.00	35.00
EM (MPa)	15.000	4.500	6.50	20.00	22.00	25.00	35.00
P	0.320	0.420	0.32	0.28	0.25	0.22	0.25
SITA	0.500	0.700	0.48	0.45	0.40	0.35	0.35
K (m/d)	0.0050	0.0050	4.50	6.50	12.00	20.0	30.0
FRB (kPa)	25.00	18.00	22.00	35.00	50.00	55.00	65.00

	Type						
Indicator	⑦$_{2-1}$	⑦$_{2-2}$	②$_{1}$	12$_{2-1}$	12$_{2-2}$	12$_{3}$	12$_{4}$
Y (kN/m^3)	18.50	18.50	19.50	20.50	22.50	24.50	26.50
FI (°)	20.50	22.50	27.00	30.00	45.00	55.00	70.00
EM (MPa)	18.000	20.000	40.000	90.000	10000.00	10000.00	10000.00
P	0.30	0.28	0.25	0.25	0.25	0.22	0.18
SITA	0.45	0.55	0.00	0.00	0.00	0.00	0.00
K (m/d)	0.50	0.50	1.00	2.50	15.00	1.50	0.50
FRB (kPa)	22.00	28.00	45.00	60.00	125.00	380.00	650.00

Table A.4
The number of times each rock-soil type appears in the 88 drilling samples.

Type	②$_{3}$	②$_{4}$	④$_{2}$	④$_{4}$	④$_{5}$	④$_{8}$	④$_{9}$	④$_{10}$	④$_{11}$	⑦$_{2-1}$
Number	7	0	4	4	0	0	0	14	0	13
Type	⑦$_{2-2}$	⑨$_{1}$	⑨$_{2-1}$	⑨$_{2-2}$	⑨$_{3}$	12$_{1}$	12$_{2-1}$	12$_{2-2}$	12$_{3}$	12$_{4}$
Number	40	11	7	0	0	8	4	0	1	0

Table A.5
TBM operational parameters [43].

Parameter (Unit)	Parameter (Unit)
Temperature of oil tank (°C)	Temperature of gear oil (°C)
Rotation speed of cutter(r/min)	Cutter power (kw)
Propelling pressure (bar)	Propelling pressure of A group (bar)
Propelling pressure of B group (bar)	Propelling pressure of C group (bar)
Propelling pressure of D group (bar)	Pressure of equipment bridge (bar)
Pressure of articulation system (bar)	Pressure of Shield tail seal at top right front (bar)
Pressure of Shield tail seal at right front (bar)	Pressure of Shield tail seal at bottom left front (bar)
Pressure of Shield tail seal at top right back (bar)	Pressure of Shield tail seal at right back (bar)
Pressure of Shield tail seal at bottom left back (bar)	Pressure of Shield tail seal at left front (bar)
Pressure of Shield tail seal at top left front (bar)	Pressure of Shield tail seal at left back (bar)
Pressure of Shield tail seal at top left back (bar)	Pressure of Shield tail seal at bottom right back (bar)
Rolling angle (°)	Pressure of screw pump at back (bar)
Pressure of chamber at top left (bar)	Pressure of chamber at bottom left (bar)
Pressure of chamber at bottom right (bar)	Bentonite pressure (bar)
Temperature of screw conveyor (°C)	Pitch angle (°)
Thrust of cutterhead (kN)	Advance velocity (mm/min)
Torque of cutterhead (kNm)	Displacement of A group of thrust cylinders (mm)
Displacement of B group of thrust cylinders (mm)	Displacement of C group of thrust cylinders (mm)
Displacement of D group of thrust cylinders (mm)	Displacement of articulated system at top right (mm)
Displacement of articulated system at bottom left (mm)	Displacement of articulated system at top left (mm)
Displacement of articulated system at bottom right (mm)	Bentonite pressure of shield shell (bar)
Pressure of screw conveyor at front (bar)	Pressure of screw pump (bar)

References

[1] M. Cigla, S. Yagiz, L. Ozdemir, Application of tunnel boring machines in underground mine development, in: E. Ünal, B. Ünver, E. Tercan. (Eds.), 17th International Mining Congress and Exhibition of Turkey, Ankara, Turkey, 2001, pp. 155–164. URL http://www.maden.org.tr/resimler/ekler/43857f4a06c8521_ek.pdf.

[2] H. Mroueh, I. Shahrour, A simplified 3d model for tunnel construction using tunnel boring machines, Tunn. Undergr. Space Technol. 23 (1) (2008) 38–45, https://doi.org/10.1016/j.tust.2006.11.008. https://www.sciencedirect.com/science/article/pii/S0886779806001258.

[3] Q. Liu, J. Liu, Y. Pan, X. Zhang, X. Peng, Q. Gong, L. Du, A wear rule and cutter life prediction model of a 20-in. tbm cutter for granite: a case study of a water conveyance tunnel in china, Rock Mech. Rock Eng. 50 (5) (2017) 1303–1320.

[4] W. Sun, M. Shi, C. Zhang, J. Zhao, X. Song, Dynamic load prediction of tunnel boring machine (tbm) based on heterogeneous in-situ data, Autom. Constr. 92 (2018) 23–34.

[5] X. Song, M. Shi, J. Wu, W. Sun, A new fuzzy c-means clustering-based time series segmentation approach and its application on tunnel boring machine analysis, Mech. Syst. Sig. Process. 133 (2019), 106279.

[6] J. Huo, Z. Xu, Z. Meng, J. Li, J. Dong, L. Wang, Coupled modeling and dynamic characteristics of tbm cutterhead system under uncertain factors, Mech. Syst. Sig. Process. 140 (2020), 106664.

[7] B. Yang, S. Chen, S. Sun, L. Deng, Z. Li, W. Li, H. Li, Vibration suppression of tunnel boring machines using non-resonance approach, Mech. Syst. Sig. Process. 145 (2020), 106969.

[8] C. Qin, G. Shi, J. Tao, H. Yu, Y. Jin, J. Lei, C. Liu, Precise cutterhead torque prediction for shield tunneling machines using a novel hybrid deep neural network, Mech. Syst. Sig. Process. 151 (2021), 107386.

[9] K. Schaeffer, M. Mooney, Examining the influence of TBM-ground interaction on electrical resistivity imaging ahead of the TBM, Tunn. Undergr. Space Technol. 58 (2016) 82–98, https://doi.org/10.1016/j.tust.2016.04.003.

[10] L. Jiangang, Prediction study on karst water bursting of urban tail tansit engineering in covered karst region, in: International Conference on Electric Technology and Civil Engineering (ICETCE), IEEE, Lushan, China, April 22–24, 2011, pp. 5267–5271. https://doi.org/10.1109/ICETCE.2011.5776048. URL https://ieeexplore.ieee.org/document/5776048.

[11] L. Wei, D.R. Magee, A.G. Cohn, An anomalous event detection and tracking method for a tunnel look-ahead ground prediction system, Autom. Constr. 91 (2018) 216–225.

[12] A. Simi, G. Manacorda, The nettun project: Design of a gpr antenna for a tbm, in: 16th International Conference on Ground Penetrating Radar (GPR), IEEE, Hong Kong, China, June 13–16, 2016, pp. 1–6. https://doi.org/10.1109/ICGPR.2016.7572648. URL https://ieeexplore.ieee.org/document/7572648.

[13] Y. Khan, C.S. See, Predicting and analyzing water quality using machine learning: A comprehensive model, in: IEEE Long Island Systems, Applications and Technology Conference (LISAT), IEEE, Farmingdale, NY, USA, April 29, 2016, pp. 1–6, https://doi.org/10.1109/LISAT.2016.7494106.

[14] C. Zhang, G. Zhu, L. Yue, An advanced detection approach based on support vector machine during tunneling, in: 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), IEEE, Auckland, New Zealand, Aug 29–31, 2016, pp. 1–4. https://doi.org/10.1109/MESA.2016.7587178. URL https://ieeexplore.ieee.org/document/7587178.

[15] S. Song, D. Jiang, H. Wang, Application of the tgp206 geological prediction method for a tunnel in the karst mountains, in: International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE), IEEE, Nanjing, China, June 24–26, 2011, pp. 2952–2955. https://doi.org/10.1109/RSETE.2011.5964933. URL https://ieeexplore.ieee.org/document/5964933.

[16] S. Leu, T. Joko, A. Sutanto, Applied real-time bayesian analysis in forecasting tunnel geological conditions, in: IEEE International Conference on Industrial Engineering and Engineering Management, IEEE, Macao, China, Dec 7–10, 2010, pp. 1505–1508. https://doi.org/10.1109/IEEM.2010.5674155. URL https://ieeexplore.ieee.org/document/5674155.

[17] D. Zhuang, K. Ma, C. Tang, Z. Liang, K. Wang, Z. Wang, Mechanical parameter inversion in tunnel engineering using support vector regression optimized by multi-strategy artificial fish swarm algorithm, Tunn. Undergr. Space Technol. 83 (2019) 425–436.

[18] A. Alimoradi, A. Moradzadeh, R. Naderi, M.Z. Salehi, A. Etemadi, Prediction of geological hazardous zones in front of a tunnel face using tsp-203 and artificial neural networks, Tunn. Undergr. Space Technol. 23 (6) (2008) 711–717.

[19] W.C. Von, M.A.M. Ismail, Evaluation of tunnel seismic prediction (tsp) result using the japanese highway rock mass classification system for pahang-selangor raw water transfer tunnel, in: AIP Conference Proceedings, Vol. 1892, AIP Publishing LLC, 2017, p. 030011.

[20] Q. Zhang, Z. Liu, J. Tan, Prediction of geological conditions for a tunnel boring machine using big operational data, Autom. Constr. 100 (2019) 73–83.

[21] B. Liu, R. Wang, Z. Guan, J. Li, Z. Xu, X. Guo, Y. Wang, Improved support vector regression models for predicting rock mass parameters using tunnel boring machine driving data, Tunn. Undergr. Space Technol. 91 (2019), 102958.

[22] J.-H. Jung, H. Chung, Y.-S. Kwon, I.-M. Lee, An ann to predict ground condition ahead of tunnel face using tbm operational data, KSCE J. Civ. Eng. 23 (7) (2019) 3200–3206.

[23] G.H. Erharter, T. Marcher, C. Reinhold, Application of artificial neural networks for underground construction–chances and challenges–insights from the bbt exploratory tunnel ahrental pfons, Geomech. Tunnel. 12 (5) (2019) 472–477.

[24] G.H. Erharter, T. Marcher, C. Reinhold, Artificial neural network based online rockmass behavior classification of tbm data, in: International Conference on Inforatmion technology in Geo-Engineering Springer, 2019, pp. 178–188.

[25] M. Shi, W. Sun, T. Zhang, Y. Liu, S. Wang, X. Song, Geology prediction based on operation data of tbm: comparison between deep neural network and soft computing methods, in: The First International Conference on Industrial Artificial Intelligence (IAI) IEEE, 2019, pp. 1–5.

[26] B. Liu, R. Wang, G. Zhao, X. Guo, Y. Wang, J. Li, S. Wang, Prediction of rock mass parameters in the tbm tunnel based on bp neural network integrated simulated annealing algorithm, Tunn. Undergr. Space Technol. 95 (2020) 103.

[27] J. Zhao, M. Shi, G. Hu, X. Song, C. Zhang, D. Tao, W. Wu, A data-driven framework for tunnel geological-type prediction based on tbm operating data, IEEE Access 7 (2019) 66703–66713.

[28] X. Li, G. Gong, Predictive control of slurry pressure balance in shield tunneling using diagonal recurrent neural network and evolved particle swarm optimization, Autom. Constr. 107 (2019), 102928.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[30] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv: 1511.06434 (2015).

[31] E.L. Denton, S. Chintala, R. Fergus, et al., Deep generative image models using a laplacian pyramid of adversarial networks, in: Advances in neural information processing systems, 2015, pp. 1486–1494.

[32] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-attention generative adversarial networks, arXiv preprint arXiv:1805.08318 (2018).

[33] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875 (2017).

[34] X. Mao, Q. Li, H. Xie, R.Y. Lau, Z. Wang, S. Paul Smolley, Least squares generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2794–2802.

[35] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[36] J. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[37] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: Sequence generative adversarial nets with policy gradient, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2852–2858.

[38] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, D. Jurafsky, Adversarial learning for neural dialogue generation, 2017 arXiv preprint arXiv:1701.06547.

[39] J. Wu, C. Zhang, T. Xue, B. Freeman, J. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, in: Advances in neural information processing systems, 2016, pp. 82–90.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Łstrok;. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[41] S. Kuhn, M.J. Cracknell, A.M. Reading, Lithologic mapping using random forests applied to geophysical and remote-sensing data: A demonstration study from the eastern goldfields of australia, Geophysics 83 (4) (2018) B183–B193.

[42] J. Halotel, V. Demyanov, A. Gardiner, Value of geologically derived features in machine learning facies classification, Math. Geosci. 52 (1) (2020) 5–29.

[43] X. Gao, M. Shi, X. Song, C. Zhang, H. Zhang, Recurrent neural networks for real-time prediction of tbm operating parameters, Autom. Constr. 98 (2019) 225–235.