The 2020 World Congress on
Advances in Civil, Environmental, & Materials Research (ACEM20)
25-28, August, 2020, GECE, Seoul, Korea

Deterministic and Probabilistic Analysis of Tunnel Face Stability using Support Vector Machine

Bin Li[1)], Yong Fu[2)], Yi Hong[3)], *Zijun Cao[4)]

[1)]School of Transportation, Wuhan University of Technology, Hubei Highway Engineering Research Center, 1178 Heping Avenue, Wuhan, Hubei Province 430063, China.
[2)]Department of Ocean Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
[3)]College of Civil Engineering and Architecture, Zhejiang University, Hang Zhou, China
[4)]State Key Laboratory of Water Resources and Hydropower Engineering Science, Key Laboratory of Rock Mechanics in Hydraulic Structural Engineering, Ministry of Education,Wuhan University, 8 Donghu South Road, Wuhan, China
** Corresponding Author. Email: zijuncao@whu.edu.cn*

ABSTRACT

This paper develops a convenient approach for deterministic and probabilistic evaluations of tunnel face stability using support vector machine classifiers. The proposed method is comparised of two major steps, i.e., construction of the training dataset and determination of instance-based classifiers. In step one, the orthogonal design is utilized to produce representative samples after the ranges and levels of the factors that influence tunnel face stability are specified. The training dataset is then labeled by two-dimensional strength reduction analyses embedded within OptumG2. For any unknown instance, the second step applies the training dataset for classification, which is achieved by an ad hoc Python program. The classification of unknown samples starts with selection of instance-based training samples using the *k*-nearest neighbors algorithm, followed by the construction of an instance-based SVM-KNN classifier. It eventually provides labels of the unknown instances, avoiding to calculate its corresponding performance function. Probabilistic evaluations are performed by Monte Carlo simulation based on the SVM-KNN classifier. The ratio of the number of unstable samples to the total number of simulated samples is computed and is taken as the failure probability, which is validated and compared with the response surface method.

Keywords: Tunnel face stability; Support vector machine; The k-nearest neighbors; Strength reduction analysis; Monte Carlo simulation

[1)] Associate Professor, Email: phdlibin@whut.edu.cn

[2)] Assistant Professor, Email: fuy3@sustech.edu.cn

[3)] Associate Professor, Email: yi_hong@zju.edu.cn

[4)] Professor, Email: zijuncao@whu.edu.cn

1. INTRODUCTION

Tunnel face stability is a major concern in tunnelling engineering because the instability of a tunnel face may severely threaten the safety of workers, equipment and existing nearby structures and utilities. Face collapse is very likely to happen at the construction stage for tunnels driven in soils. In these cases, stabilization measures are usually required for the purpose of prevention and control of active failure in front of the tunnel face. However, the application of such measures is often costly, time-consuming, and labor-intensive. Therefore, it is essential to predict where the face collapse may occur during the tunnel construction.

There are various analytical, experimental, and numerical methods for evaluations of tunnel face stability in the literature (Lee, 2016; Khezri et al., 2016; Zhang et al., 2017; Li and Yang, 2019a; Li and Yang, 2019b). Leca and Dormieux (1990) proposed a collapse mechanism composed of two truncated conical blocks to determine retaining pressures against face collapse and blow-out failure. The two-block failure mechanism was then improved to a multiblock collapse mechanism (Mollon et al. 2009; Mollon et al., 2010), and then to a rotational failure mechanism (Mollon et al., 2011). These methods need to prescribe a failure surface, and then establish and solve the governing equations with respect to tunnel face stability through the limit analysis method. In addition, centrifuge model tests have been used to study the face stability of tunnels, with the purpose of observing failure patterns and determining limit support pressures (Chambon and Corte, 1994), validating the proposed analytical methods (Leca and Dormieux, 1990), investigating failure mechanism (Wong et al., 2012), and checking the effects of auxiliary measures (Kamata and Mashimo, 2003; Juneja et al., 2010). Recently, numerical methods have become popular in tunnel face stability analysis as it does not need to prescribe the failure surface prior to the analysis compared with the analytical methods, and has been shown to be a cost-effective tool for investigating the tunnel face stability. Simulations have been performed to evaluate the effects of reinforcements on preventing face collapse (Paternesi et al., 2017), or to design reinforcement parameters (Ng and Lee, 2002; Dias, 2011; Li et al., 2015).

Tunnel face stability is mainly influenced by various factors, including, but not limtied to, the material parameters of surrounding soils, the surcharge loading located on the ground surface, the cover depth, and the diameter of the tunnel. In practice, these factors usually vary from location to location, indicating that the states of tunnel face may change frequently along the axis of the tunnel. This means that one tunnel requires a number of evaluations of tunnel face stability for different cross sections, which is time-consuming for conventional methods. A load factor, called stable ratio or stability number (Klar et al., 2007; ITA/AITES, 2007), is capable of estimating whether or not a tunnel face

is stable with a great efficiency. However, this method is only limited to tunnels driven in purely cohesive soils. A more widely applicable approach that has comparable efficiency is still expected. Essentially, if the main purpose is to determine whether or not a tunnel face is stable, the deterministic evaluation task can be reformulated as a binary classification problem. In a similar spirit, a probabilistic evaluation can be converted to a certain number of binary classifications (Pan and Dias, 2017). Various machine learning algorithms like support vector machine (SVM) have been proven to be efficient in solving such types of problems (Wang, 2005; Pal, 2006; Goh and Goh, 2007; Bourinet et al., 2011; Samui and Karthikeyan, 2013; Tinoco et al., 2014; Pham et al., 2016).

Given a training dataset, a machine learning model can be trained to classify unknown samples. The performance of the model relies on the quantity and quality of training data (or samples) (Chan et al., 2004; Wang, 2005). The training samples can be either derived from monitoring data of real-life projects or produced according to sampling design methods, such as uniform design (Li et al., 2016; Li and Yang, 2019) and orthogonal design (Qiu et al., 2016; Xu et al., 2017). Generally, geometry parameters (e.g., the diameter of a tunnel) is usually not taken into account when choosing variables (Mahdevari et al., 2012; Ge et al., 2013; Shi et al., 2019; Zhang et al., 2019). The trained model is, therefore, not feasible to be used for cases with different geometry parameters.

To overcome the above limitation, this paper constructs a training dataset that includes sufficient labeled samples at the beginning and then selects suitable training samples from this dataset for the classification. The main factors that influence tunnel face stability are taken into consideration when determining training samples, and the experimental domain to be defined wide enough to cover routine cases occurring in design practice. Beyond that, the technique of selection of suitable training samples is also very important. For a new data point, the k-nearest neighbors (KNN) algorithm is capable of searching the k-nearest data points, which has the shortest distances from the data point concerned. KNN was combined with SVM to develop a SVM-KNN method (Zhang et al., 2006), which has been proven to outperform the original SVM (Zhang et al., 2006; Li et al., 2008). This proposed SVM-KNN method is applicable to both deterministic and probabilistic evaluations of tunnel face stability, and it is comprised of two major steps: construction of the training dataset and determination of instance-based classifiers using SVM-KNN, which are described in detail as below.

2. Description of the method

2.1 General Framework

Fig. 1 shows the proposed framework comprised of two major steps. The step one (represented by solid rectangles) constructs a training dataset (denoted by $\boldsymbol{D}$). The

second step (represented by blank rectangles) applies the training dataset to perform classifications of tunnel face stability.

For an unknown instance (denoted by $\boldsymbol{x}^*$), KNN is employed to search the k nearest data points from the training dataset $\boldsymbol{D}$ to construct an instance-based training dataset $\boldsymbol{D}^*$, which is then used to fit an instance-based model to classify the unknown instance $\boldsymbol{x}^*$. Note that once the step one is completed, the next step can be readily repeated for different unknown instances. This enables one to perform a large number of predictions in a cost-effective manner. More importantly, with the trained model, probabilistic analysis of tunnel face stability becomes trivial by performing classifications for a set of instances (or samples) simulated from Monte Carlo simulation. More details of the proposed method are described in the following subsections.

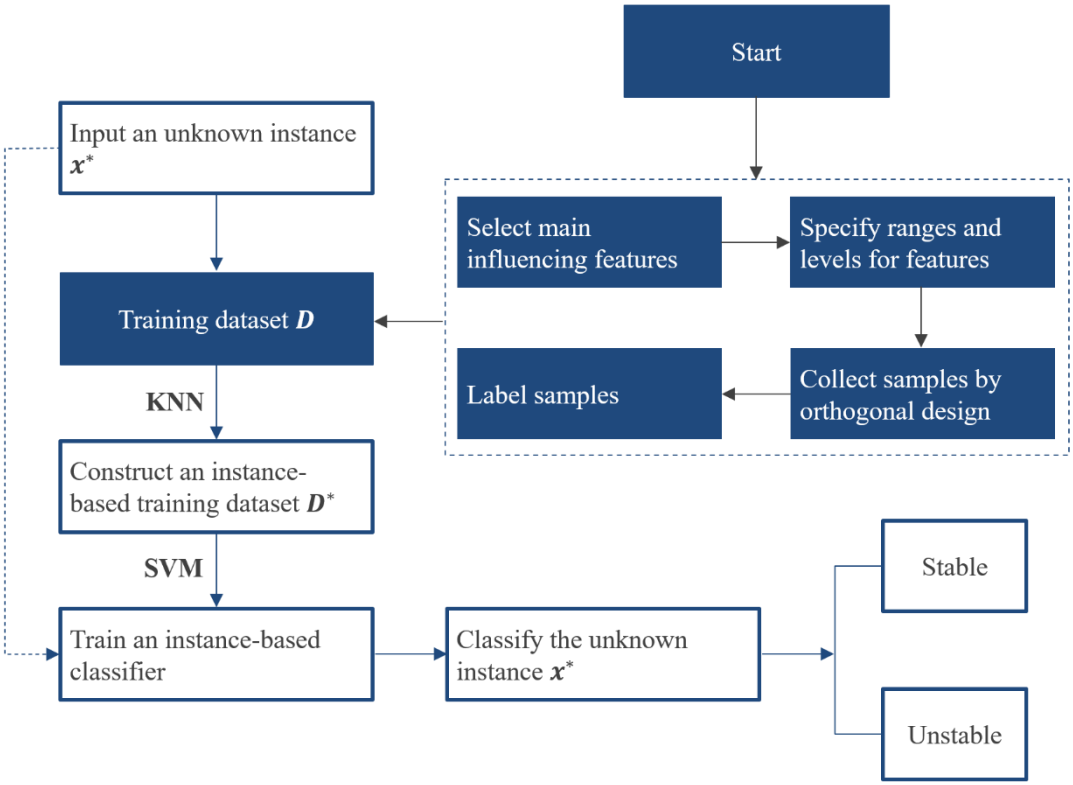


Fig. 1 Framework of the proposed SVM-KNN method for tunnel face stability analysis

2.2 Generation of the training samples

The step one starts with selecting main features affecting the tunnel face stability. In this study, six features are considered, including the tunnel diameter D, the cover depth H, the surcharge load σ, the unit weight γ, the cohesion c, and the friction angle φ of surrounding soils. In other words, an instance can be represented by a six-dimensional vector, $\boldsymbol{x_i} = (D_i, H_i, \sigma_i, \gamma_i, c_i, \varphi_i)$.

Then, an experimental domain is defined within the feature space. This is done by specifying the range of each factor. Consider, for example, a two-dimensional feature space of γ and D, the ranges of which can be specified as $13 \leq x_1 \leq 24$ $(\mathrm{kN/m^3})$ and $4 \leq x_2 \leq 15$ (m), respectively. For the convenience of visualization, Fig. 2 shows the subspace of γ and D bounded by the assigned ranges, which is defined as the experimental domain. The defined domain shall be large enough to cover most of routine cases in design practice. Instances that exceed this domain are deemed to be special cases which deserve special consideration. Within the specified domain, a set of combinations (denoted by $\boldsymbol{x_1, x_2, \ldots, x_i}$) of model parameters are determined, which are then labeled to form the training dataset. After this, a decision boundary is constructed. Unknown samples denoted by vectors can be categorized as either stable (e.g., $\boldsymbol{x^*_1}$) or unstable (e.g., $\boldsymbol{x^*_2}$) samples according to the computed value of the decision function.

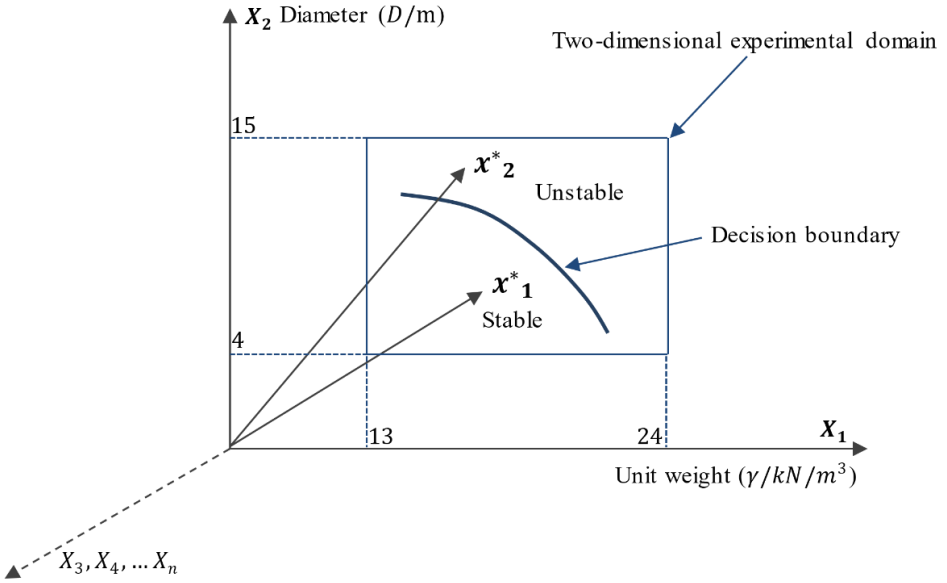
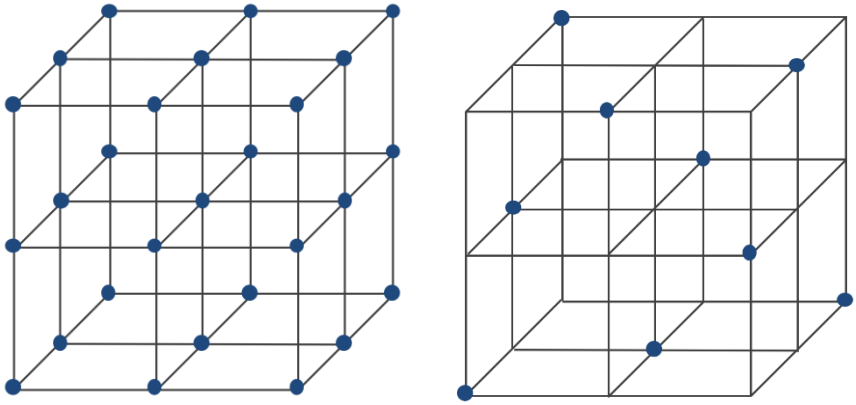


Fig. 2 Schematic diagram of the feature space

The commonly used experimental design methods include full factorial design, orthogonal design, and uniform design. Considering m factors and n levels per factor, full factorial design involves n^m combinations (experiments), which can be significantly decreased to n^2 using an orthogonal design (Cavazzuti, 2013). Consider, for example, $m = 6$ and $n = 12$, full factorial design will produce $12^6 = 2985984$ samples. Labeling such a large number of training samples is computationally expensive. However, the orthogonal design only generates $12^2 = 144$ combinations. Fig. 3 compares the combinations for 3 factors and 3 levels per factor produced by full factorial design and orthogonal design methods. Using the orthogonal design, the number of combinations is decreased from 27 to 9. However, the experimental domain is uniformly covered by

selected combinations of model parameters, which are hence representative (Wang et al., 2013; Wang et al., 2014).



(a) **(b)**

Fig. 3 Comparison of combinations for 3 factors and 3 levels per factor: (a) Full factorial design; (b) Orthogonal design

2.3 Labeling of the training samples

Finite element analyses are then utilized to label the training samples. This task is achieved using the strength reduction technique, by which the safety factor (F_s) is calculated as:

$$F_s = \frac{c}{c_{cr}} = \frac{\tan\varphi}{\tan\varphi_{cr}} \tag{1}$$

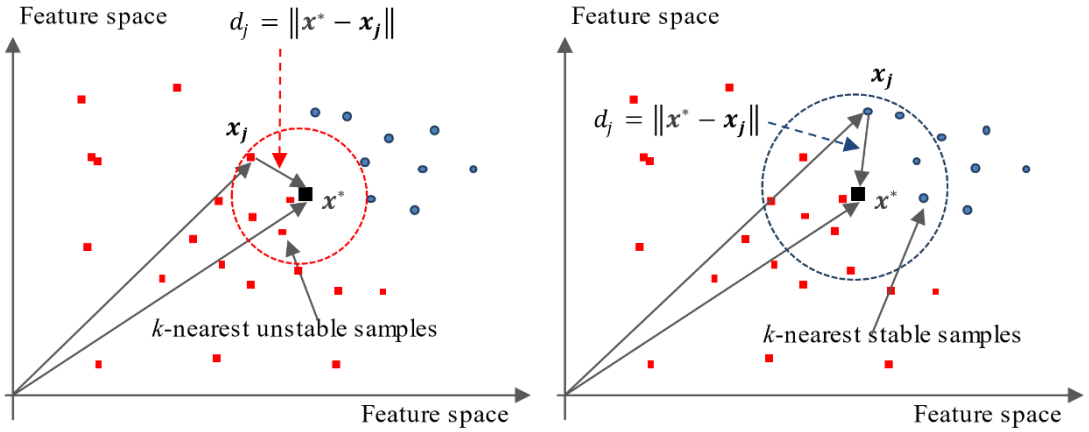
where c and φ denote the strength parameters of surrounding soils; c_{cr} and φ_{cr} denote their critical values in terms of tunnel face stability. If $F_s \geq 1$, the tunnel face is stable; if $F_s < 1$, the funnel face is unstable. Then, each instance, $\boldsymbol{x_i} = (D_i, H_i, \sigma_i, \gamma_i, c_i, \varphi_i)$, is labeled as stable ($y_i = +1$) for $F_s \geq 1$ or unstable ($y_i = -1$), for $F_s < 1$ to obtain a set of training samples:

$$\boldsymbol{D} = \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \ldots, (\boldsymbol{x_i}, y_i)\} \tag{2}$$

2.4 Selection of instance-based training samples

After the selected samples have been labeled, the training dataset $\boldsymbol{D}$ is applied to performing classifications. For an unknown instance $\boldsymbol{x}^*$, the first step is to measure the distance between $\boldsymbol{x}^*$ and each data point (represented by $\boldsymbol{x_j}$) in the feature space. A commonly used distance can be defined by the Euclidean norm, $d_j = \|\boldsymbol{x}^* - \boldsymbol{x_j}\|$, as

shown in Fig.4. The measured distances are stored and sorted in an ascending order. The k data points that have the shortest distances from the unknown instance will be selected to form an instance-based training dataset $\boldsymbol{D}^*$. Considering the balance between the number of stable and unstable training samples, it is recommended to select k nearest unstable samples and k nearest stable samples, respectively. In other words, $2k$ samples will be selected to form $\boldsymbol{D}^*$.



(a) **(b)**

Fig. 4 Selection of instance-based training dataset: a) $\boldsymbol{k}$-nearest unstable samples; b) $\boldsymbol{k}$-nearest stable samples

2.5 SVM-KNN classifiers

The instance-based training dataset $\boldsymbol{D}^*$ is then used to train a SVM classifier. Fig. 5 illustrates the SVM algorithm (Duan et al., 2003). For the selected training samples, the aim is to determine a maximum-margin separating hyperplane, or named decision boundary that separates stable and unstable samples such that their distance is maximum.

The expression of the decision boundary may be written as (Goh and Goh, 2007):

$$\vec{\boldsymbol{\omega}} \cdot \vec{\boldsymbol{x}} + b = 0 \tag{3}$$

where $\vec{\boldsymbol{\omega}}$ represents the normal vector to the hyperplane; $\vec{\boldsymbol{x}}$ represents the vector of a sample; b is a constant whose normalized form $b/\|\vec{\boldsymbol{\omega}}\|$ determines the offset of the hyperplane from the original along with the normal vector $\vec{\boldsymbol{\omega}}$. The maximum-margin hyperplane is taken as the optimal hyperplane for separating samples. In addition, there exist another two parallel hyperplanes which are expressed as:

$$\vec{\boldsymbol{\omega}} \cdot \vec{\boldsymbol{x}}_{\boldsymbol{i}} + b = 1, y_i = 1 \tag{4}$$

and

$$\vec{\omega} \cdot \vec{x}_i + b = -1, y_i = -1 \tag{5}$$

These $\vec{x}_i$ are called support vectors that lie on the boundaries on both sides. The distance between these two hyperplanes is $2/\|\vec{\omega}\|$. The strategy is to maximize this distance, which is equivalent to minimize $\|\vec{\omega}\|$.

For all the stable samples, the constraint can be written as:

$$\vec{\omega} \cdot \vec{x}_i + b \geq 1, \text{if } y_i = 1 \tag{6}$$

For the unstable samples, the constraint is written as:

$$\vec{\omega} \cdot \vec{x}_i + b \leq -1, \text{if } y_i = -1 \tag{7}$$

The two constraints can be synthesized and be rewritten as:

$$y_i(\vec{\omega} \cdot \vec{x}_i + b) \geq 1, \text{for all } 1 \leq i \leq n \tag{8}$$

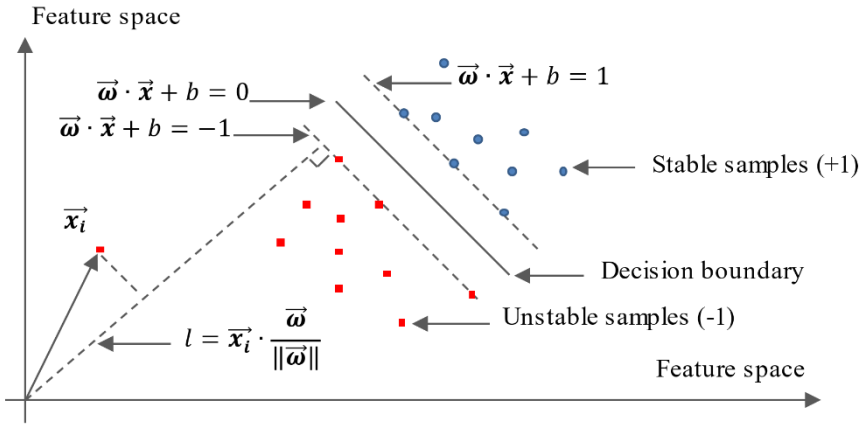


Fig. 5 The SVM classification algorithm (Duan et al., 2003)

The minimization of $\|\vec{\omega}\|$ subject to the constraint given in Eq. (8) is known as a class of convex quadratic optimization problem. It can be solved by the method of Lagrange multipliers which introduces a set of variables called Lagrange multipliers (denoted by $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots \alpha_m), \alpha_i \geq 0$) to define an auxiliary function:

$$L(\vec{\omega}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\vec{\omega}\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i(\vec{\omega} \cdot \vec{x}_i + b)) \tag{9}$$

and solve:

$$\frac{\partial L}{\partial \vec{\omega}} = 0; \quad \frac{\partial L}{\partial b} = 0 \tag{10}$$

The optimum of $\vec{\boldsymbol{\omega}}$ and b can be determined after the Lagrange multipliers have been solved based on Eqs. (9) and (10). Thereafter, the value of the decision function (Eq. (9)) of a new sample (denoted by $\vec{\boldsymbol{x}}_*$) can be computed as follow:

$$z = \mathrm{sgn}(\vec{\boldsymbol{\omega}} \cdot \vec{\boldsymbol{x}}_* + b) \tag{9}$$

A positive value will assign the new sample to the stable category, while a negative value will assign it to the unstable category. Note that Eq. (9) represents a linear classifier, which is applicable to cases where the data is linearly separable. On the other hand, if the data is linearly inseparable, nonlinear classifiers can be developed by applying other kernels (Goh and Goh, 2007; Samui, 2008). Some common kernels include Polynomial (Eq. (10)), Gaussian radial basis function, RBF (Eq. (11)), and Sigmoid (Eq. (12)):

$$\kappa(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j})^d \tag{10}$$

$$\kappa(\vec{x_i}, \vec{x_j}) = \exp\left(-\gamma \left\|\vec{x_i} - \vec{x_j}\right\|^2\right) \tag{11}$$

$$\kappa(\vec{x_i}, \vec{x_j}) = \tanh(\kappa \vec{x_i} \cdot \vec{x_j} + c) \tag{12}$$

where $\vec{x_i}$ and $\vec{x_j}$ denote vectors in the feature space, d and c are constants, γ denotes the variance of Gaussian kernel. By incorporating one kernel into the decision function, a new sample can be classified by computing:

$$\mathrm{z} = \mathrm{sgn}(\vec{\boldsymbol{\omega}} \cdot \varphi(\vec{\boldsymbol{z}}) - b) = \mathrm{sgn}\left(\left[\sum_{i=1}^{n} c_i y_i k(\vec{\boldsymbol{x_i}}, \vec{\boldsymbol{z}})\right] - b\right) \tag{13}$$

where $\vec{\boldsymbol{z}}$ denotes the vector of the new sample; $\vec{\boldsymbol{x_i}}$ denotes the support vectors which depend on the used kernel and the training samples, c_i denote coefficients, b is a constant. Before the classifier is applied to classifying new samples, validation is needed to examine its prediction accuracy. A commonly used indicator is the average accuracy:

$$\text{Average accuracy} = \frac{n_F}{n_F + n_T} \tag{14}$$

where n_F and n_T represent the number of false and true classifications, respectively. Note that the prediction accuracy on both training and test datasets have to be checked. A test dataset refers to a dataset that is different from the training dataset. It can be obtained either by collecting some new samples that are not included in the training dataset or by employing the *k*-fold cross-validation technique (Kohavi, 1995). The *k*-fold cross-validation method splits the training dataset into *k* subsets, each of which is held

as testing dataset to check the prediction accuracy of the classifier trained from the remaining subsets.

3. ILLUSTRATIVE EXAMPLES

3.1 Training of SVM Classifiers

In this section, deterministic and probabilistic application examples are presented to illustrate the proposed method. Consider, for example, the six impact factors shown in Table 1. Their values are specified within certain ranges so as to cover routine cases in practice. Each feature is assigned to has 12 levels within these ranges. For cases that exceed the specified domain, it is recommended to perform specific studies case by case. If it is necessary, both the ranges and levels of the six features considered in this example can be modified deemed appropriate to meet particular requirements. After the experimental domain and the level of each factor have been determined, the orthogonal design is utilized to produce 144 samples, each of which is labeled by the two-dimensional finite element analysis.

It should be emphasized that tunnel face stability is more appropriate to be simulated by three-dimensional modeling. However, it requires great effort to build 144 different 3D numerical models, and huge computational costs to complete 144 3D numerical simulations as well. Since the main purpose here is to validate the feasibility and efficiency of the proposed method, this example utilizes 2D simulations to label the samples. Moreover, the main concern is not the values of the safety factors, but the labels that depend on if the safety factors are larger than 1.0. This means that the labels of most samples are reliable even though they are determined based on 2D simulation results. Only a few samples that are close to the limit state may have different labels due to the simplification in 2D modeling in comparison with 3D modeling (Li, 2019).

Table 1 Six impact factors and twelve levels per factor

Factors / Levels	D/m	H/m	σ_s/kPa	γ(kN/m^3)	c/kPa	φ/°
1	4	5	0	13	0	0
2	5	10	10	14	4	4
3	6	15	20	15	8	8
4	**7**	20	30	16	12	12
5	8	25	40	17	16	16
6	9	30	50	18	20	20

7	10	35	60	19	24	24
8	11	40	70	20	28	28
9	12	45	80	21	32	32
10	13	50	90	22	36	36
11	14	55	100	23	40	40
12	15	60	110	24	44	44

The software OptumG2 (Krabbenhoft, 2018) was used for numerical simulations. Fig. 6 shows the schematic diagram of a two-dimensional finite element model, in which the geometry parameters and boundary conditions are indicated. The norm fixities were applied to left and right boundaries, while full fixities were applied to the bottom boundary. Some other geometry parameters (H_1, L_1 and L_2) are constant, values of which were chosen so as not to affect the tunnel face stability (avoid boundary effects).

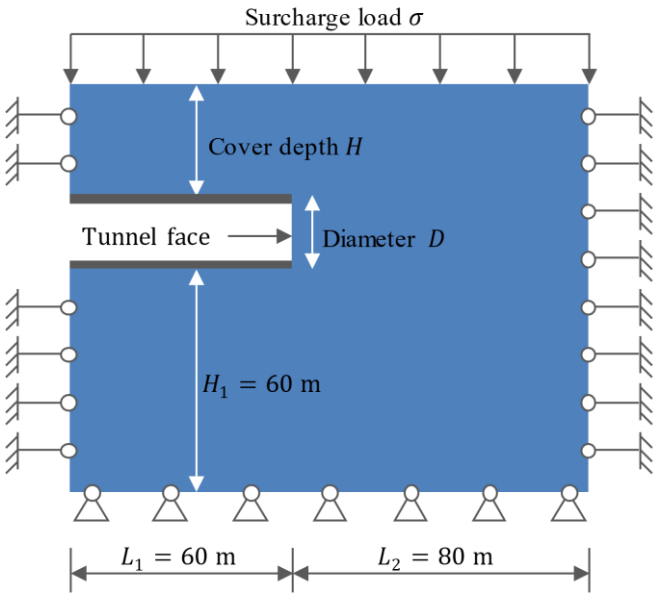


Fig. 6 Schematic diagram of the numerical model and boundary conditions

The soil is modeled as an elasto-plastic material following Mohr-Coulomb failure criterion. The mesh is assigned to 2000 adaptive elements. The built-in strength reduction analysis was employed to compute the safety factors of the training samples, among which there are 95 unstable samples and 49 stable samples, as shown Fig. 7. Thereafter, the step one of the proposed approach is completed and the total training data $\boldsymbol{D}$ is obtained, as summarized in Table 2, where $y = 1$ and $y = -1$ represent stable and unstable samples, respectively.

Step two starts with determining an instance-based training dataset $\boldsymbol{D}^*$, followed by training an instance-based SVM classifier. In this example, the value of k is set to its

maximum, 49, as it is the number of stable samples in the complete training dataset. As a result, for each unknown instance, all the stable samples and 49 nearest unstable samples will be selected to form $\boldsymbol{D}^*$. This procedure is implemented with an ad hoc Python program, which also imports the SVM modulus from scikit-learn (Pedregosa et al., 2011) to fit the training data $\boldsymbol{D}^*$. Kernel functions including Linear kernel, Polynomial kernel, Gaussian kernel (RBF), and Sigmoid kernel are employed to construct both linear and nonlinear SVM-KNN classifiers.

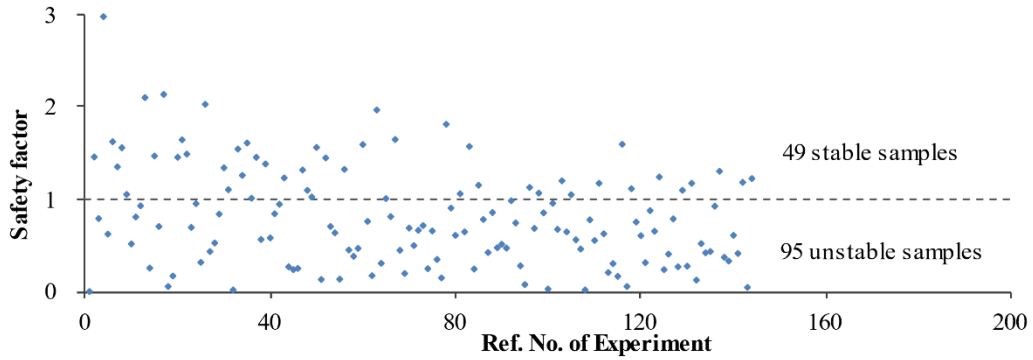


Fig. 7 Safety factors of the training samples

Table 2 Summary of training data

Ref. no.	$(d,H,\sigma,\gamma,c,\varphi)$	y	Ref. no.	$(d,H,\sigma,\gamma,c,\varphi)$	y	Ref. no.	$(d,H,\sigma,\gamma,c,\varphi)$	y
1	4,5,0,13,0,0	-1	**25**	6,5,90,24,4,16	-1	**49**	8,5,60,23,20,36	1
2	4,10,60,16,32,16	1	**26**	6,10,30,20,44,36	1	**50**	8,10,0,18,36,32	1
3	4,15,80,19,4,44	-1	**27**	6,15,20,15,8,8	-1	**51**	8,15,110,20,12,0	-1
4	4,20,20,14,44,40	1	**28**	6,20,80,18,40,0	-1	**52**	8,20,50,22,28,44	1
5	4,25,70,22,20,8	-1	**29**	6,25,100,21,12,28	-1	**53**	8,25,40,17,16,16	-1
6	4,30,10,24,36,28	1	**30**	6,30,40,16,28,24	1	**54**	8,30,100,14,24,8	-1
7	4,35,90,15,12,32	1	**31**	6,35,10,22,32,20	1	**55**	8,35,20,16,0,28	-1
8	4,40,110,21,40,24	1	**32**	6,40,50,19,0,4	-1	**56**	8,40,80,15,44,20	1
9	4,45,40,18,8,36	1	**33**	6,45,110,17,20,40	1	**57**	8,45,30,24,40,4	-1
10	4,50,100,17,28,4	-1	**34**	6,50,70,23,24,32	1	**58**	8,50,10,21,8,12	-1
11	4,55,50,20,24,12	-1	**35**	6,55,0,14,16,44	1	**59**	8,55,70,13,4,24	-1
12	4,60,30,23,16,20	-1	**36**	6,60,60,13,36,12	1	**60**	8,60,90,19,32,40	1
13	5,5,20,19,40,32	1	**37**	7,5,50,17,32,28	1	**61**	9,5,110,15,28,12	-1
14	5,10,10,14,4,4	-1	**38**	7,10,100,19,8,20	-1	**62**	9,10,70,24,0,40	-1
15	5,15,70,17,36,20	1	**39**	7,15,40,21,24,40	1	**63**	9,15,10,13,40,36	1
16	5,20,90,20,8,24	-1	**40**	7,20,30,16,12,12	-1	**64**	9,20,60,21,16,4	-1
17	5,25,30,15,24,44	1	**41**	7,25,90,13,44,4	-1	**65**	9,25,0,23,32,24	1

18	5,30,80,23,0,12	-1	**42**	7,30,110,22,16,32	-1	**66**	9,30,50,18,20,20	-1
19	5,35,40,24,20,0	-1	**43**	7,35,70,14,40,16	1	**67**	9,35,100,20,36,44	1
20	5,40,100,16,16,36	1	**44**	7,40,20,23,36,0	-1	**68**	9,40,30,17,4,32	-1
21	5,45,60,22,44,28	1	**45**	7,45,0,20,4,8	-1	**69**	9,45,90,16,24,0	-1
22	5,50,50,13,12,40	1	**46**	7,50,60,18,0,44	-1	**70**	9,50,40,19,44,8	-1
23	5,55,110,18,32,8	-1	**47**	7,55,80,24,28,36	1	**71**	9,55,20,22,12,16	-1
24	5,60,0,21,28,16	-1	**48**	7,60,10,15,20,24	1	**72**	9,60,80,14,8,28	-1

Table 2 Summary of training data (continued)

Ref. no.	$(d, H, \sigma, \gamma, c, \varphi)$	y	**Ref. no.**	$(d, H, \sigma, \gamma, c, \varphi)$	y	**Ref. no.**	$(d, H, \sigma, \gamma, c, \varphi)$	y
73	10,5,30,21,36,8	-1	**97**	12,5,70,16,8,44	-1	**121**	14,5,80,22,24,4	-1
74	10,10,50,15,16,0	-1	**98**	12,10,110,13,24,28	1	**122**	14,10,20,21,20,44	-1
75	10,15,100,24,32,12	-1	**99**	12,15,50,23,44,16	-1	**123**	14,15,90,18,16,28	-1
76	10,20,40,23,4,28	-1	**100**	12,20,10,17,0,8	-1	**124**	14,20,70,15,32,36	1
77	10,25,110,14,0,36	-1	**101**	12,25,60,20,40,20	-1	**125**	14,25,10,19,28,0	-1
78	10,30,90,17,40,44	1	**102**	12,30,0,19,12,36	-1	**126**	14,30,30,13,8,16	-1
79	10,35,60,19,24,24	-1	**103**	12,35,30,18,28,40	1	**127**	14,35,0,17,44,12	-1
80	10,40,0,22,8,40	-1	**104**	12,40,90,14,20,12	-1	**128**	14,40,60,24,12,8	-1
81	10,45,20,13,28,20	1	**105**	12,45,80,21,32,32	1	**129**	14,45,50,14,36,24	1
82	10,50,80,20,20,16	-1	**106**	12,50,20,24,16,24	-1	**130**	14,50,110,16,4,20	-1
83	10,55,10,16,44,32	1	**107**	12,55,40,15,36,4	-1	**131**	14,55,100,23,40,40	1
84	10,60,70,18,12,4	-1	**108**	12,60,100,22,4,0	-1	**132**	14,60,40,20,0,32	-1
85	11,5,100,18,44,24	1	**109**	13,5,10,20,16,40	-1	**133**	15,5,40,14,12,20	-1
86	11,10,40,22,40,12	-1	**110**	13,10,80,17,12,24	-1	**134**	15,10,90,23,28,8	-1
87	11,15,0,16,20,4	-1	**111**	13,15,60,14,28,32	1	**135**	15,15,30,22,0,24	-1
88	11,20,110,19,36,16	-1	**112**	13,20,0,24,24,20	-1	**136**	15,20,100,13,20,32	-1
89	11,25,50,24,8,32	-1	**113**	13,25,20,18,4,12	-1	**137**	15,25,80,16,36,40	1
90	11,30,60,15,4,40	-1	**114**	13,30,70,21,44,0	-1	**138**	15,30,20,20,32,4	-1
91	11,35,80,13,16,8	-1	**115**	13,35,110,23,8,4	-1	**139**	15,35,50,21,4,36	-1
92	11,40,70,20,28,28	-1	**116**	13,40,40,13,32,44	1	**140**	15,40,10,18,24,16	-1
93	11,45,10,23,12,44	-1	**117**	13,45,100,15,0,16	-1	**141**	15,45,70,19,16,12	-1
94	11,50,30,14,32,0	-1	**118**	13,50,90,22,36,36	1	**142**	15,50,0,15,40,28	1
95	11,55,90,21,0,20	-1	**119**	13,55,30,19,20,28	-1	**143**	15,55,60,17,8,0	-1
96	11,60,20,17,24,36	1	**120**	13,60,50,16,40,8	-1	**144**	15,60,110,24,44,44	1

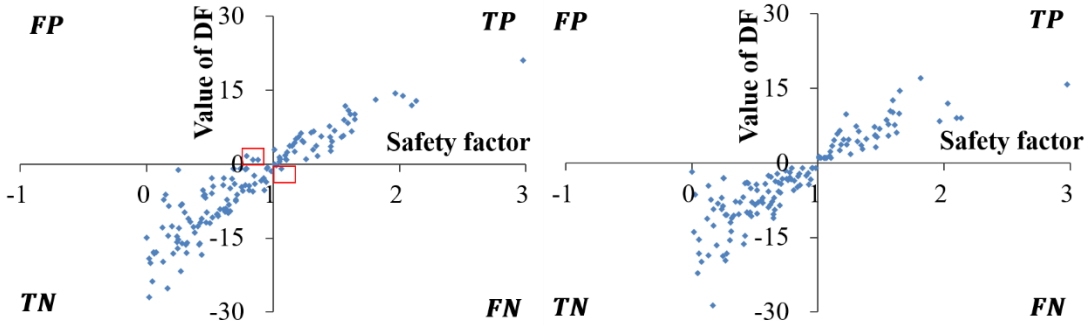
For comparison, the complete training dataset $\boldsymbol{D}$ is also used to construct several SVM classifiers with different kernel functions. The average accuracies of these classifiers are shown in Table 3. Generally, combining SVM with KNN slightly improves

the prediction accuracies on the training data. The performance of the classifier with the sigmoid kernel function is the worst as indicated by its lowest prediction accuracy on the training data (Van der Aalst et al., 2010). It is therefore not suitable for separating new samples.

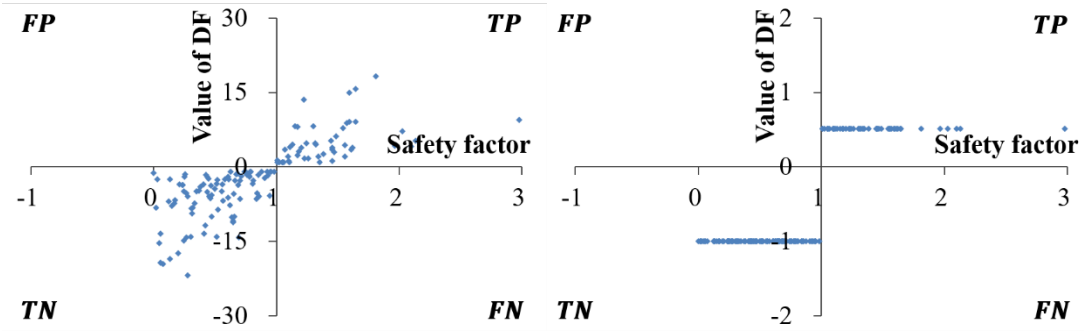
The prediction performance of the SVM classifiers can be observed according to the relationship between the simulated safety factors (denoted as FS) and the computed values of decision functions (denoted as DF), as shown in Fig 8. The points that lie in the first and third quadrants represent true positive (TP) and negative (TN) samples, respectively. In contrast, the points that lie in the second and fourth quadrants represent false positive and negative samples, respectively, which are misclassified samples as indicated by red rectangles. The comparison between Fig.8 (a) and Fig.8 (b) demonstrates that the polynomial kernel function can give better prediction than the linear kernel function for samples that are very close to the limit state (DF=0, FS=1).

Table 3 Average accuracies of different classifiers

Kernel functions	**Average accuracy**	
	SVM-KNN	SVM
Linear	0.972	0.958
Polynomial (degree = 2)	1.000	1.000
Polynomial (degree = 3)	1.000	1.000
RBF	0.993	1.000
Sigmoid	0.683	0.660



(a) (b)



(c) (d)

Fig. 8 Safety factor versus value of decision function: a) linear kernel; b) polynomial kernel (degree=2); c) polynomial kernel (degree=3); d) RBF kernel

Although the above classifiers are not underfitting, it still needs to examine their performance on test data to avoid overfitting. This study employs a 10-fold cross-validation to perform this task. For this purpose, the training data was divided into 10 subsets, each of which was held as testing data to check the prediction accuracy of the classifier trained by the remaining 9 subsets. The average accuracy of SVM-KNN and SVM classifiers on testing data are presented in Table 4.

The classifiers that are constructed based on the RBF have low prediction accuracies on testing data, which means that they are deemed to be overfitting. On the contrary, other classifiers that have relatively high prediction accuracies can be therefore used to classify unknown samples in the following analysis. During this validation process, the use of KNN also slightly improved the prediction accuracies of the classifiers using the linear, polynomial ($\mathrm{degree} = 2$) and RBF kernel functions.

3.2 Deterministic application example

This section presents results of deterministic applications. Table 5 listed some experimental and computed collapse pressures in the existing literature (Chambon and Corte, 1994; Mollon et al., 2010), where the computed values of the decision functions (based on polynomial with $\mathrm{degree} = 2$) are included as well as the simulated safety factors.

The variable σ_c represents the limit pressure that is required to be applied on the tunnel face to avoid face collapse. Almost all the presented cases have positive values of σ_c, indicating that the tunnel faces are initially unstable. The two cases with no collapse pressure (Mollon et al., 2010) means that the tunnel faces are initially stable. These states can be alternatively evaluated using the SVM-KNN and SVM classifiers. All the computed values of the decision function are negative, classifying all these cases into the unstable category. On the other hand, results are validated according to

numerical calculations, in which all the simulated safety factors are less than 1.0.

Table 4 Results of k-fold cross-validation (k = 10)

Kernel functions	**Average accuracy**	
	SVM-KNN	SVM
Linear	0.959	0.945
Polynomial (degree = 2)	0.945	0.924
Polynomial (degree = 3)	0.938	0.966
RBF	0.683	0.660

Table 5 Evaluations of tunnel face stability with existing samples

D	H	σ	γ	c	φ	**Mollon, 2010 (σ_c)**	**Chambon, 1994(σ_c)**	**SVM-KNN**	**SVM**	F_s
5	2.5	0	16.1	0	38	6.8	3.6	-2.44	-4.07	0.186
5	2.5	0	16.1	5	38	0.4	3.6	-1.55	-2.45	0.776
5	2.5	0	16.1	0	42	5.3	3.6	-2.80	-4.67	0.215
5	2.5	0	16.1	5	42	Stable	3.6	-1.81	-2.87	0.776
5	2.5	0	15.3	0	38	6.5	4.2	-2.44	-4.09	0.186
5	2.5	0	15.3	5	38	0.1	4.2	-1.55	-2.48	0.801
5	2.5	0	15.3	0	42	5	4.2	-2.81	-4.70	0.215
5	2.5	0	15.3	5	42	Stable	4.2	-1.82	-2.90	0.856
10	10	0	16	0	38	13.6	7.4	-3.20	-5.23	0.155
10	10	0	16	5	38	7.1	7.4	-2.32	-3.71	0.547
10	10	0	16	0	42	10.5	7.4	-3.52	-5.77	0.179
10	10	0	16	5	42	5	7.4	-2.54	-4.07	0.598
13	42	0	16.2	0	38	17.9	13	-3.10	-4.98	0.154
13	42	0	16.2	5	38	11.4	13	-2.08	-3.35	0.487
13	42	0	16.2	0	42	13.8	13	-3.00	-4.92	0.177
13	42	0	16.2	5	42	8.3	13	-1.88	-3.11	0.518

3.3 Probabilistic application example

Probabilistic evaluations may be performed by Monte Carlo simulations based on deterministic models (Cao et al., 2017). As illustrated in Fig. 9, a number of Monte Carlo samples can be first simulated from prescribed distributions of uncertain parameters (e.g., uncertain features concerned in design), and each Monte Carlo sample is then used as

input in deterministic models (e.g., SVM-KNN classifiers in this study) to obtain its corresponding predictions (e.g., stable or unstable).

Let n^- and n^+ be the number of unstable and stable predictions, respectively. The failure probability of the tunnel face is calculated as:

$$p_f = \frac{n^-}{n^- + n^+} \tag{15}$$

Generating Monte Carlo samples and calculating the failure probability is also performed by an in-house Python program in this study. For instance, consider a tunnel example with the vector of parameters $\boldsymbol{x} = [4{,}20{,}0{,}19, c, \varphi]$, in which c and φ are assumed to be normal random variables. The probability density functions of c and φ can be given as follow:

$$f_c = \frac{1}{\sqrt{2\pi \cdot {\sigma_c}^2}} e^{\left(\frac{-(c-\mu_c)^2}{2\times {\sigma_c}^2}\right)} \tag{16}$$

$$f_\varphi = \frac{1}{\sqrt{2\pi \cdot {\sigma_\varphi}^2}} e^{\left(\frac{-(c-\mu_\varphi)^2}{2\times {\sigma_\varphi}^2}\right)} \tag{17}$$

where μ_c and σ_c denote the mean and standard deviation of the cohesive strength, respectively; μ_φ and σ_φ denote the mean and standard deviation of the frictional angle, respectively. Fig. 10 shows the two sets of statistics of c and φ, and their corresponding samples generated from Monte Carlo simulations.

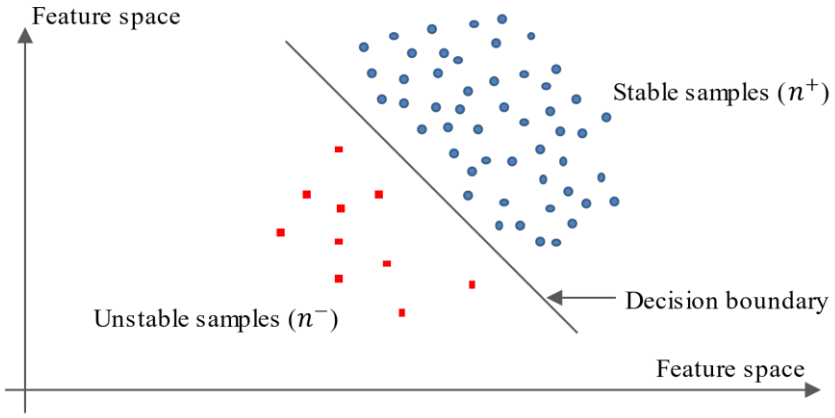
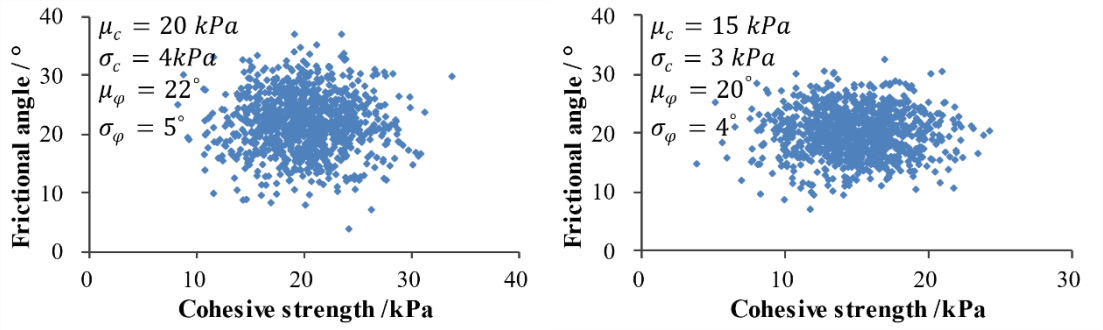


Fig. 9 Illustration of probabilistic applications



(a) (b)

Fig. 10 Two set of statistics of c and φ and their corresponding samples: a) Set 1; b) Set 2

For the first set of samples, the values of decision function with polynomial kernel function (degree $= 2$) were calculated and sorted in an ascending order. As shown in Fig. 11, 153 out of 1000 samples were predicted to be unstable. Hence, the failure probability of the tunnel face is 15.3%.

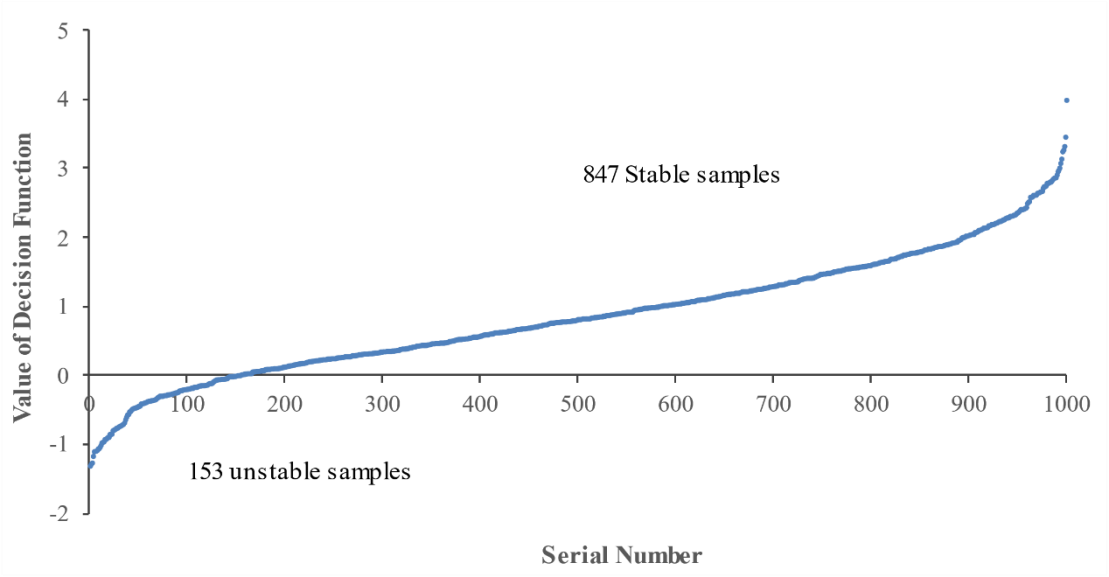


Fig. 11 The values of the decision function sorted in ascending order

For validation, the failure probabilities of the two cases are also computed using the Response Surface Method, which utilizes a quadratic polynomial to approximate the response surface of the performance function concerning tunnel face stability, as shown in Eq. (18):

$$Z = F_s - 1 = a_0 + a_1 c + a_2 \varphi + a_3 c^2 + a_4 \varphi^2 \tag{18}$$

where a_0 and a_i denote the unknown coefficients. For each case, the central composite design (CCD) is used to sample five points. The performance function is solved with the safety factors of the five points simulated in strength reduction analyses, enabling the reliability indexes of the two cases to be computed using the Advanced First-Order Second-Moment method, as shown in Table 6.

Based on Eq. (19), the failure probabilities of the two cases are determined to be 14.1% and 47.1%, respectively.

$$p_f = 1 - \Phi(\beta) \tag{19}$$

where Φ represents the standard normal cumulative distribution function (CDF).

Table 6 Simulated safety factors and computed reliability indexes

Dataset	Set 1			Set 2		
	c/kPa	$\varphi/°$	F_s	c/kPa	$\varphi/°$	F_s
Point 1	20	22	1.223	15	20	1.011
Point 2	24	22	1.327	15	24	1.127
Point 3	16	22	1.101	15	16	0.888
Point 4	20	27	1.383	18	20	1.099
Point 5	20	17	1.052	12	20	0.913
Reliability index, $\boldsymbol{\beta}$		1.075			0.073	

In the SVM-KNN method, Monte Carlo samples are randomly generated, the failure probability may vary from one run to another due to random fluctuation. Fig. 12 shows the computed failure probabilities obtained by 10 runs of Monte Carlo simulations, yielding 10 estimates of failure probabilities. The mean values of these 10 failure probabilities is 15.2% and 54.0%, respectively. Generally, these results compare well with those determined by RSM, which validates the proposed method.

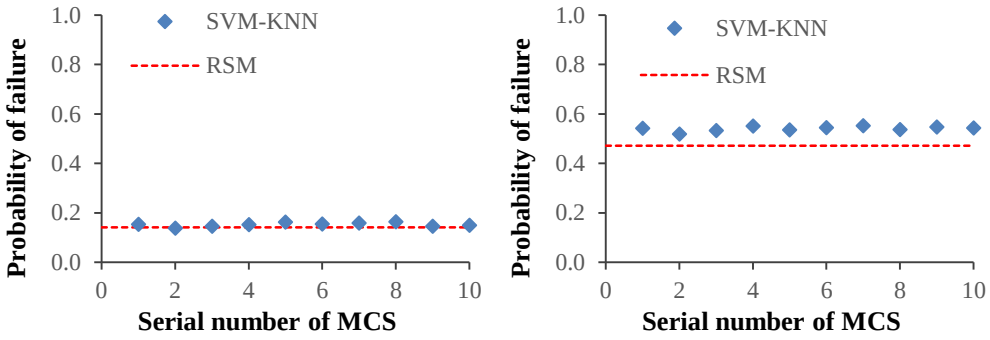
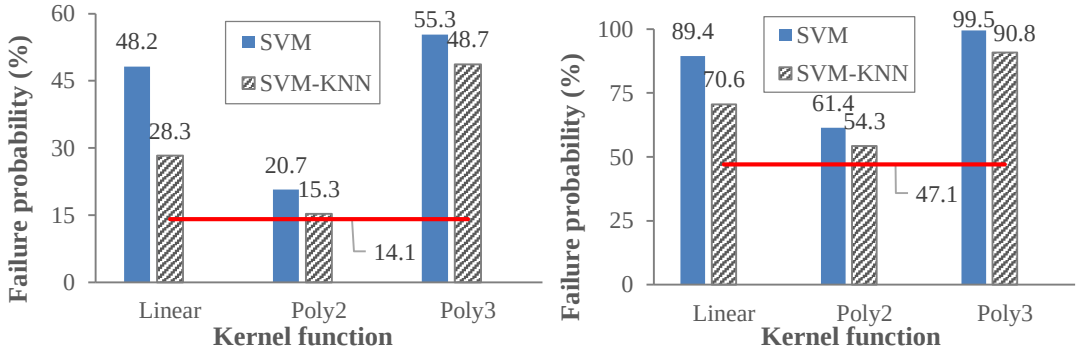


Fig. 12 Failure probabilities determined by RSM and SVM-KNN (10 times MCS)

Fig. 13 compares the performance in probabilistic applications between SVM-KNN and SVM classifiers constructed based on different kernel functions, where the failure probabilities calculated using RSM are represented by red solid lines. Results show that SVM-KNN classifiers can give better estimations of the failure probabilities than SVM classifiers. On the other hand, the polynomial kernel function with $\text{degree} = 2$ is found to be the most suitable kernel function in probabilistic applications.

In fact, the accuracy of the failure probability relies on the similarity between the trained decision function and the real limit state function. The SVM uses only one decision function to separate all input samples, whereas SVM-KNN utilizes a instance-dependent localized decision function for each sample. The instance-based classifier has a better performance in classification since it makes use a localized decision boundary. This is why SVM-KNN gives more accurate estimations of failure probabilities.

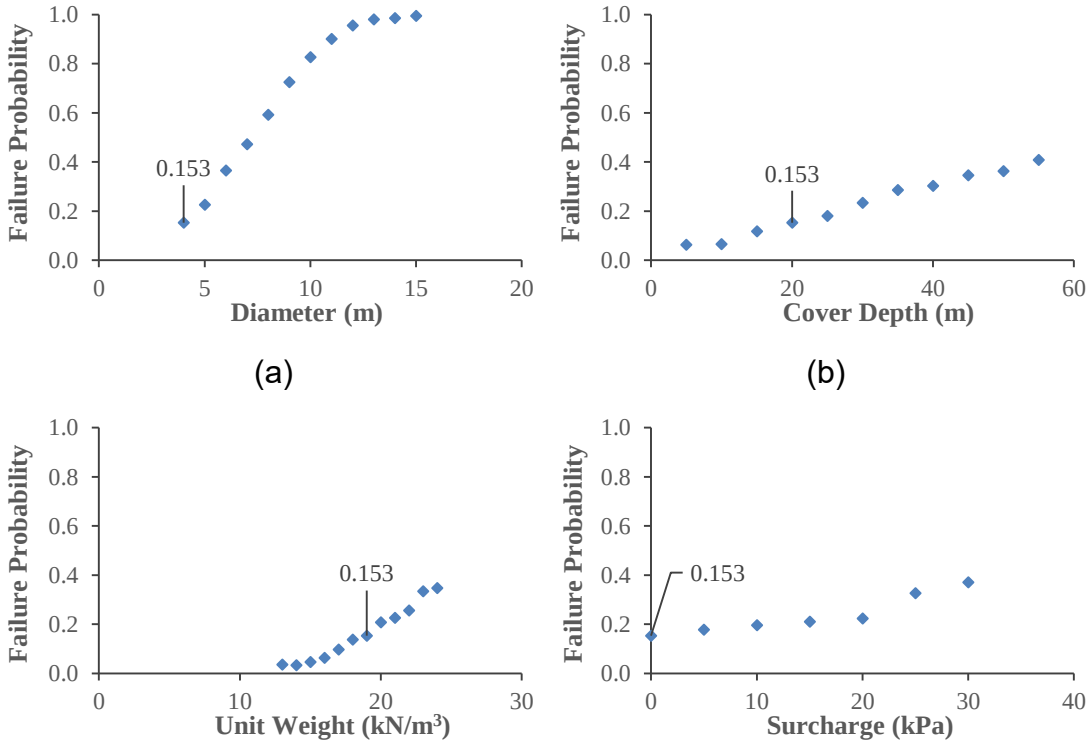
Moreover, the procedures of the proposed approach (including generation of Monte Carlo samples, selection of instance-based training dataset, construction of instance-based classifiers, classification of Monte Carlo samples, as well as the calculation of failure probability) can be completed within several seconds. Whereas RSM requires five numerical simulations for each case specified by geometric parameters (e.g., *H*, *D*) to obtain the safety factors to develop the instance-based performance function. Such a process is relatively easy for a single case with specific geometric parameters but very time-consuming and tedious for a number of cases if geometric parameters can vary, which is of great interest in design practice.



(a) (b)

Fig. 13 Comparison of failure probabilities: a) Set 1; b) Set2

To show the convenience and efficiency of this method, six groups of failure probabilities are estimated. The failure probability of each case was computed in several seconds. In this way, the relationship between each factor and the failure probability can be obtained, as shown in Fig. 14, where the failure probability ($p_f = 0.153$) of the original case is also indicated. Generally, the failure probability increases with the increase of the diameter, the cover depth, the unit weight, and the surcharge, but decreases with the increase of the cohesion and the friction angle, which are consistent with intuitive experience in design practice.



(a) (b)

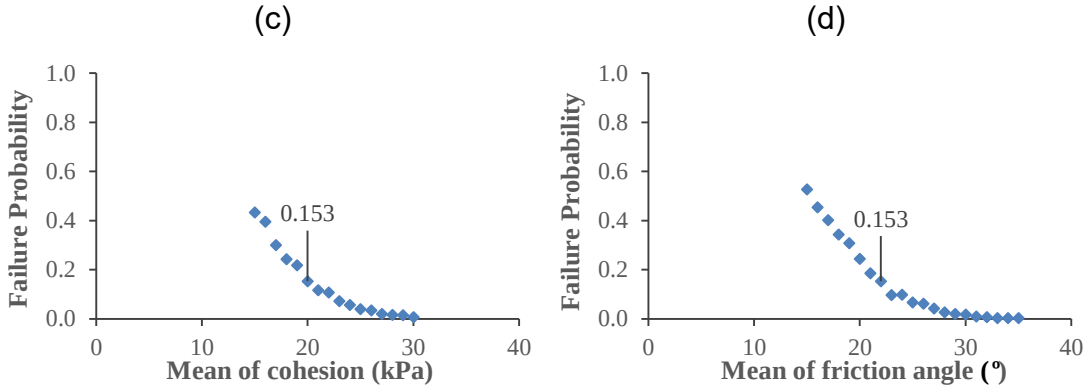


Fig. 14 Influence of factors on failure probability: a) The diameter; b) The cover depth; c) Unit weight; d) Surcharge; e) Mean of cohesion; f) Mean of friction angle

4. CONCLUSION

This paper proposed an efficient method to facilitate deterministic and probabilistic evaluations of tunnel face stability according to binary classifications. In the illustrative examples, the procedures of proposed appraoch are performed in an ad hoc python program. By this means, both deterministic and probabilistic evaluations of tunnel face stability can be automatically completed within negligible computational efforts. Based on this study, several major conclusions can be drawn:

1. Applicability of the classifiers can be improved by expanding the ranges deemed appropriate, whereas the performance of the classifiers can be improved by collecting more training samples. SVM-KNN gives relatively accurate results, particularly in probabilistic evaluations because it trains a localized decision function for each sample, whereas SVM separates all the samples using a single decision function.

2. Another crucial influence on the performance of a classifier is the kernel function. The polynomial kernel function with $\mathrm{degree} = 2$ is found to be the most suitable one to train nonlinear classifiers with relatively high performance for tunnel face stability analysis in this study. Selection of the kernel function in probabilistic evaluations is much more important than in deterministic evaluations. This is mainly because there might be many instances located close to the decision function. These instances are sensitive to the change in the position of the decision function.

3. Both deterministic and probabilistic evaluations can be completed within several seconds, allowing a number of evaluations to be performed in a cost-effective manner. The convenience and efficiency of the proposed method make it very useful in design practice, particularly when real-time evaluations are required, because the factors (including geometric parameters) that influence tunnel face stability may vary from one location to another along the tunnel.

ACKNOWLEDGMENT:

This study is supported by the National Natural Science Foundation of China (No.51608407), the NRF-NSFC 3rd Joint Research Grant (Earth Science) (No. 41861144022), the Fundamental Research Funds for the Central Universities (No. 2042019kf1022), and the China Scholarship Council (No. 201706955065).

REFERENCE:

Bourinet, J.-M., Deheeger, F., Lemaire, M. (2011), "Assessing small failure probabilities by combined subset simulation and Support Vector Machinhttps," strusafe., 06, 001

Cao, Z., Wang, Y., Li, D. (2017), "Practical reliability analysis of slope stability by advanced Monte Carlo simulations in a spreadsheet", Probabilistic Approaches for Geotechnical Site Characterization and Slope Stability Analysis. Springer., 147–167.

Cavazzuti, M. (2013), "Design of experiments", Optimization Methods. Springer., 13–42.

Chambon, P., Corte, J.-F. (1994), "Shallow tunnels in cohesionless soil: stability of tunnel face," J. Geotech. Eng., 120, 1148–1165.

Chan, H.-P., Sahiner, B., Hadjiiski, L. (2004), "Sample size and validation issues on the development of CAD systems," Int. Congr. Ser., 1268, 872–877.

Dias, D. (2011), "Convergence-confinement approach for designing tunnel face reinforcement by horizontal bolting," Tunn. Undergr. Space Technol., 26, 517–523.

Duan, K., Keerthi, S.S., Poo, A.N. (2003), "Evaluation of simple performance measures for tuning SVM hyperparameters," Neurocomputing., 51, 41–59.

Ge, Y., Wang, J., Li, K. (2013), "Prediction of hard rock TBM penetration rate using least square support vector machine," IFAC Proc. Vol., 13th IFAC Symposium on Large Scale Complex Systems: Theory and Applications., 46, 347–352.

Goh, A.T.C., Goh, S.H. (2007), "Support vector machines: Their use in geotechnical engineering as illustrated using seismic liquefaction data," Comput. Geotech., 34, 410–421.

ITA/AITES (2007), "Settlements induced by tunnelling in soft ground," Tunn. Undergr. Space Technol., 22, 119–149.

Juneja, A., Hegde, A., Lee, F.H., Yeo, C.H. (2010), "Centrifuge modelling of tunnel face reinforcement using forepoling," Tunn. Undergr. Space Technol., 25, 377–381.

Kamata, H., Mashimo, H. (2003), "Centrifuge model test of tunnel face reinforcement by bolting," Tunn. Undergr. Space Technol., 18, 205–212.

Khezri, N., Mohamad, H., Fatahi, B. (2016), "Stability assessment of tunnel face in a layered soil using upper bound theorem of limit analysis," Geomech. Eng., 11, 471–492.

Klar, A., Osman, A.S., Bolton, M. (2007), "2D and 3D upper bound solutions for tunnel

excavation using ‘elastic’flow fields," Int. J. Numer. Anal. Methods Geomech., 31, 1367–1374.

Kohavi, R. (1995), "A study of cross-validation and bootstrap for accuracy estimation and model selection," Ijcai. Montreal, Canada., 1137–1145.

Krabbenhoft, K. (2018), "Optum Computational Engineering".

Leca, E., Dormieux, L. (1990), "Upper and lower bound solutions for the face stability of shallow circular tunnels in frictional material," Geotechnique., 40, 581–606.

Lee, Y.-J. (2016), "Determination of tunnel support pressure under the pile tip using upper and lower bounds with a superimposed approach," Geomech. Eng., 11, 587–605.

Li, B., Li Hong. (2019), "Prediction of Tunnel Face Stability Using a Naive Bayes Classifier," Appl. Sci., 9, 4139.

Li, B., Hong, Y., Gao, B., Qi, T.Y., Wang, Z.Z., Zhou, J.M. (2015), "Numerical parametric study on stability and deformation of tunnel face reinforced with face bolts," Tunn. Undergr. Space Technol., 47, 73–80.

Li, T.Z., Yang, X.L. (2019a), "Probabilistic analysis for face stability of tunnels in Hoek-Brown media," Geomech. Eng., 18, 595–603.

Li, T.Z., Yang, X.L. (2019b), "Face stability analysis of rock tunnels under water table using Hoek-Brown failure criterion," Geomech. Eng., 18, 235–245.

Li, T.Z., Yang, X.L. (2019c), "An efficient uniform design for Kriging-based response surface method and its application," Comput. Geotech., 109, 12–22.

Li, Xiang, Li, Xibing, Su, Y. (2016), "A hybrid approach combining uniform design and support vector machine to probabilistic tunnel stability assessment," Struct. Saf., 61, 22–42.

Mahdevari, S., Torabi, S.R., Monjezi, M. (2012), "Application of artificial intelligence algorithms in predicting tunnel convergence to avoid TBM jamming phenomenon," Int. J. Rock Mech. Min. Sci., 55, 33–44.

Mollon, G., Dias, D., Soubra, A.-H. (2011), "Rotational failure mechanisms for the face stability analysis of tunnels driven by a pressurized shield," Int. J. Numer. Anal. Methods Geomech., 35, 1363–1388.

Mollon, G., Dias, D., Soubra, A.-H. (2010), "Face stability analysis of circular tunnels driven by a pressurized shield," J. Geotech. Geoenvironmental Eng., 136, 215–229.

Mollon, G., Dias, D., Soubra, A.-H. (2009), "Probabilistic analysis and design of circular tunnels against face stability," Int. J. Geomech., 9, 237–249.

Ng, C.W.W., Lee, G.T.K. (2002), "A three-dimensional parametric study of the use of soil nails for stabilising tunnel faces," Comput. Geotech., 29, 673–697.

Pal, M. (2006), "Support vector machines-based modelling of seismic liquefaction potential," Int. J. Numer. Anal. Methods Geomech., 30, 983–996.

Pan, Q., Dias, D. (2017), "An efficient reliability method combining adaptive Support Vector Machine and Monte Carlo Simulation," Struct. Saf., 67, 85–95.

Paternesi, A., Schweiger, H.F., Scarpelli, G. (2017), "Numerical analyses of stability and deformation behavior of reinforced and unreinforced tunnel faces," Comput. Geotech., 88, 256–266.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. (2011), "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., 12, 2825–2830.

Pham, B.T., Bui, D.T., Dholakia, M.B., Prakash, I., Pham, H.V. (2016), "A comparative study of least square support vector machines and multiclass alternating decision trees for spatial prediction of rainfall-induced landslides in a tropical cyclones area," Geotech. Geol. Eng., 34, 1807–1824.

Qiu, Z., Ruan, J., Huang, D., Wei, M., Tang, L., Huang, C., Xu, W., Shu, S. (2016), "Hybrid prediction of the power frequency breakdown voltage of short air gaps based on orthogonal design and support vector machine," IEEE Trans. Dielectr. Electr. Insul., 23, 795–805.

Samui, P. (2008), "Support vector machine applied to settlement of shallow foundations on cohesionless soils," Comput. Geotech., 35, 419–427.

Samui, P., Karthikeyan, J. (2013), "Determination of liquefaction susceptibility of soil: a least square support vector machine approach," Int. J. Numer. Anal. Methods Geomech., 37, 1154–1161.

Shi, S., Zhao, R., Li, S., Xie, X., Li, L., Zhou, Z., Liu, H. (2019), "Intelligent prediction of surrounding rock deformation of shallow buried highway tunnel and its engineering application," Tunn. Undergr. Space Technol., 90, 1–11.

Tinoco, J., Correia, A.G., Cortez, P. (2014), "Support vector machines applied to uniaxial compressive strength prediction of jet grouting columns," Comput. Geotech., 55, 132–140.

Van der Aalst, W.M., Rubin, V., Verbeek, H.M.W., van Dongen, B.F., Kindler, E., Günther, C.W. (2010), "Process mining: a two-step approach to balance between underfitting and overfitting," Softw. Syst. Model., 9, 87.

Wang, L. (2005), "Support vector machines: theory and applications," Springer Science & Business Media.

Wang, Y., Zhao, X., Wang, B. (2013), LS-SVM and Monte Carlo methods based reliability analysis for settlement of soft clayey foundation. J. Rock Mech. Geotech. Eng., 5, 312–317.

Wang, Z., Qiao, C., Song, C., Xu, J. (2014), "Upper bound limit analysis of support pressures of shallow tunnels in layered jointed rock strata," Tunn. Undergr. Space Technol., 43, 171–183.

Wong, K.S., Ng, C.W.W., Chen, Y.M., Bian, X.C. (2012), "Centrifuge and numerical investigation of passive failure of tunnel face in sand," Tunn. Undergr. Space Technol., 28, 297–303.

Xu, J., Ren, Q., Shen, Z. (2017), "Sensitivity analysis of the influencing factors of slope stability based on LS-SVM," Geomech. Eng., 13, 447–458.

Zhang, B., Wang, X., Zhang, J.S., Meng, F. (2017), "Three-dimensional limit analysis of seismic stability of tunnel faces with quasi-static method," Geomech. Eng., 13, 301–318.

Zhang, H., Berg, A.C., Maire, M., Malik, J. (2006), "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2126–2136.

Zhang, P., Chen, R.-P., Wu, H.-N. (2019), "Real-time analysis and regulation of EPB shield steering using Random Forest," Autom. Constr. 106, 102860.