Hindawi Publishing Corporation
Journal of Construction Engineering
Volume 2014, Article ID 109184, 9 pages
http://dx.doi.org/10.1155/2014/109184

Research Article

An Artificial Intelligence Approach for Groutability Estimation Based on Autotuning Support Vector Machine

Hong-Hai Tran and Nhat-Duc Hoang

Department of Technology and Construction Management, Faculty of Building and Industrial Construction, National University of Civil Engineering, Room 307, A1 Building, No. 55 Giai Phong Road, Hanoi 10000, Vietnam

Correspondence should be addressed to Nhat-Duc Hoang; ducxd85@yahoo.com

Received 23 December 2013; Revised 27 February 2014; Accepted 20 March 2014; Published 10 April 2014

Academic Editor: Eric Lui

Permeation grouting is a commonly used approach for soil improvement in construction engineering. Thus, predicting the results of grouting activities is a crucial task that needs to be carried out in the planning phase of any grouting project. In this research, a novel artificial intelligence approach—autotuning support vector machine—is proposed to forecast the result of grouting activities that employ microfine cement grouts. In the new model, the support vector machine (SVM) algorithm is utilized to classify grouting activities into two classes: *success* and *failure*. Meanwhile, the differential evolution (DE) optimization algorithm is employed to identify the optimal tuning parameters of the SVM algorithm, namely, the penalty parameter and the kernel function parameter. The integration of the SVM and DE algorithms allows the newly established method to operate automatically without human prior knowledge or tedious processes for parameter setting. An experiment using a set of in situ data samples demonstrates that the newly established method can produce an outstanding prediction performance.

1. Introduction

In construction engineering, permeation grouting is the process that involves the injection of suitable particulate grouts or chemical solutions into the geomaterial with the aim of improving its mechanical properties and reducing the water movement through soils [1]. In particular for underground construction works, the inflow of groundwater has always been a substantial challenge for geotechnical engineers [2]. Water inflows often cause construction delays and severe damages to the structure quality. Consequently, the grouting activity is an essential task which needs to be performed in a majority of underground construction projects.

Recently, microfine cement grouts have been increasingly employed by geotechnical engineers. The reason is that microfine cement grouts can provide an improved groutability for the target geomaterial and they do not contaminate the surrounding environment. In addition, these grouts are proven to have the capacity of filling cracks with small openings as well as penetrating fine soils with very low permeability [3].

Nonetheless, one of the main challenges in the utilization of microfine cement grouts is how to accurately estimate the groutability of the target geomaterial [4]. It is because the grouting process is based on the complex time-dependent transport process of cement grains through the soil matrix. Moreover, besides the grain size of the soil and the grout, other factors that affect the outcome of grouting activities should be taken into account. Due to such complexity, existing empirical formulas [5–8] can hardly attain satisfactory results. The reason is that these formula-based approaches, which are mostly based on the grain size of the soil and the grout, are unreliable for seminanometer scale grouts.

Experimental studies done by Akbulut and Saglamer [9] and Ozgurel and Vipulanandan [10] found that, in addition to the grain size of the soil and the grout, the water-to-cement ratio of grout (w/c), the void size in soil, and the fines content (FC) of the total soil should be considered. Liao et al. [11] pointed out that information of soil gradation, namely, the coefficient of uniformity (C_u), which measures the particle size range, and the coefficient of gradation (C_z), which characterizes the particle size curve, can be useful for estimating groutability.

Characteristics of construction projects are highly uncertain and intrinsically context-dependent; therefore, artificial

intelligence (AI) methods can provide feasible alternatives for groutability prediction. From the perspective of AI, the problem at hand can be modeled as a classification task that contains two class labels (*success* and *failure*). Based on the collected data, an AI based approach can be constructed and utilized to classify new input samples.

Artificial neural network (ANN) has been applied to deal with groutability prediction as well as with other problems in the construction industry [4, 11–13]. Although ANN has been proven to be feasible in the task of groutability estimation, its implementation suffers from several drawbacks. The approach has difficulties in selecting a large number of controlling parameters (the number of hidden layers, the number of neurons in each layer, and the learning rate) [14]. Furthermore, one major disadvantage of ANN is that its training process is achieved through a gradient descent algorithm on the error space, which can be very complex and may contain many local minima [15]. Thus, the training process is likely to be trapped into a local solution and this undoubtedly hinders the predictive performance.

Recently, support vector machine (SVM), proposed by Vapnik [16], has been applied to resolve a wide span of classification problems. SVM classifies data with several class labels by identifying a set of support vectors from the set of training data; these support vectors have the role in determining the class decision boundary. Although various studies have indicated the superior performance of SVM over ANN [17–19], none of previous research works has evaluated the potentiality of SVM for groutability estimation. Thus, this paper is an attempt to fill this gap.

Moreover, when using SVM, the users need to determine its tuning parameters, namely, the penalty and kernel function parameters. Proper settings of these parameters make certain of SVM prediction accuracy. Thus, this research proposes to fuse SVM and differential evolution (DE)—a fast and effective evolution optimization technique—to construct a novel approach for groutability prediction employing microfine cement grouts. In this hybrid mechanism, the SVM technique is used to derive the decision boundary for predicting the consequence of a grouting process. Meanwhile, the DE algorithm is employed to search for the optimal set of SVM tuning parameters.

The remaining part of this paper is organized as follows. The second section of this paper presents the research methodology including the SVM algorithm, the DE algorithm, and the historical data of grouting cases. The framework of the proposed approach is depicted in the third section. The fourth section demonstrates the experimental results. The conclusion of this study is stated in the final section.

2. Research Methodology

2.1. Formula-Based Approaches for Groutability Estimation. In the literature, various researchers have attempted to develop groutability estimation methods by evaluating the relationships between the grain size of the soil and the particle size of the cement. This section of the paper reviews existing groutability prediction approaches which are stated in the form of formulas.

Burwell [6] proposed two formulas to estimate groutability as follows:

$$N_1 = \frac{(D_{15})_{\text{soil}}}{(d_{85})_{\text{grout}}}, \tag{1}$$

$$N_2 = \frac{(D_{10})_{\text{soil}}}{(d_{95})_{\text{grout}}}, \tag{2}$$

where D_{15} and D_{10} denote the diameters through which 15% and 10% of the total soil mass pass, respectively. d_{85} and d_{95} are the diameters through which 85% and 95% of the total grout pass, respectively.

According to [6], it is possible for the grouting process to be successful if N_1 is greater than 25 and N_2 is greater than 11. Meanwhile, the grouting process is not feasible if N_1 is less than 11; if $11 < N_1 < 25$, the outcome is undefined.

Incecik and Ceren [5] suggested an alternate equation as follows:

$$N = \frac{(D_{10})_{\text{soil}}}{(d_{90})_{\text{grout}}}, \tag{3}$$

where D_{10} denotes the diameter through which 10% of the total soil mass passes. d_{90} is the diameter through which 90% of the total grout passes. Based on (3), the grouting process succeeds if N is greater than 10.

Krizek et al. [7] suggested the identical equations to determine N_1 (see (1)) and N_2 (see (2)) as [6], but the authors proposed different thresholds: the grouting process succeeds if N_1 is greater than 15 and N_2 is greater than 8.

2.2. Support Vector Machine. The SVM principles are based on the structural risk minimization and statistical learning theory [16]. In general, the SVM technique classifies a data sample by mapping the data points into a high-dimensional feature space and identifying the classification boundary in such space. The advantages of SVM include strong inference capacity, good generalization, fast learning, and accurate prediction [19, 20]. This section describes the formulation of the SVM algorithm.

Given a training data set $\{x_k, y_k\}_{k=1}^{N}$ with input data $x_k \in R^n$ and corresponding class labels $y_k \in \{-1, +1\}$, the SVM formulation for classification is stated as follows:

$$\begin{aligned} &\text{Minimize} \quad J_p(w, e) = \frac{1}{2} w^T w + c \frac{1}{2} \sum_{k=1}^{N} e_k^2 \\ &\text{Subjected to} \quad y_k \left(w^T \phi(x_k) + b \right) \geq 1 - e_k, \\ &\qquad k = 1, \ldots, N, \ e_k \geq 0, \end{aligned} \tag{4}$$

where $w \in R^n$ is the normal vector to the classification hyperplane and $b \in R$ is the bias; $e_k > 0$ are called slack variables; c denotes a penalty constant; and $\phi(x)$ represents a nonlinear mapping from the input space to the high-dimensional feature space.

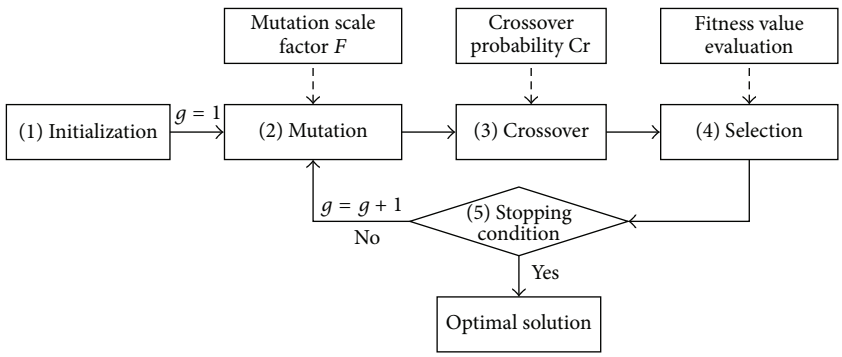

Figure 1: The differential evolution optimization algorithm.

The Lagrangian is given by:

$$L(w,b,e;\alpha;v) = J_p(w,e) - \sum_{k=1}^{N} \alpha_k \left\{ y_k \left(w^T \phi(x_k) + b \right) - 1 + e_k \right\} - \sum_{k=1}^{N} v_k e_k, \quad (5)$$

where $\alpha_k \geq 0$, $v_k \geq 0$ are Lagrange multipliers for $k = 1, 2, \ldots, N$. The conditions for optimality are given by

$$\frac{\partial L}{\partial w} = 0 \longrightarrow w = \sum_{k=1}^{N} \alpha_k y_k \phi(x_k),$$
$$\frac{\partial L}{\partial b} = 0 \longrightarrow \sum_{k=1}^{N} \alpha_k y_k = 0, \quad (6)$$
$$\frac{\partial L}{\partial e_k} = 0 \longrightarrow 0 \leq \alpha_k \leq c, \quad k = 1, \ldots, N.$$

After replacing (6) in (5), we obtain the following dual quadratic programming problem:

$$\max_{\alpha} \quad J_D(\alpha) = -\frac{1}{2} \sum_{k,l=1}^{N} y_k y_l \phi(x_k)^T \phi(x_l) \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k$$
$$\text{Subjected to} \quad \sum_{k=1}^{N} \alpha_k y_k = 0, \quad 0 \leq \alpha_k \leq c, \ k = 1, \ldots, N. \quad (7)$$

And the kernel function is applied as follows:

$$\omega = y_k y_l \phi(x_k)^T \phi(x_l) = y_k y_l K(x_k, x_l). \quad (8)$$

The resulting SVM model for classification is expressed as follows:

$$y(x) = \operatorname{sign}\left(\sum_{k=1}^{\mathrm{SV}} \alpha_k y_k K(x_k, x_l) + b \right), \quad (9)$$

where SV denotes the number of support vectors which are training data points corresponding to $\alpha_k \neq 0$. The kernel function that is often utilized is radial basis function (RBF) kernel. Description of RBF kernel is given as follows:

$$K(x_k, x_l) = \exp\left(\frac{\|x_k - x_l\|^2}{2\sigma^2} \right), \quad (10)$$

where σ is the kernel function parameter.

In the case of the RBF kernel, there are two tuning parameters (c, σ) that need to be determined to establish the SVM prediction model. The penalty parameter (c) is used to weight the importance of classification errors. Meanwhile, the kernel parameter (σ) affects the kernel width.

2.3. Differential Evolution. This section describes the DE algorithm proposed by Price and Storn [21, 22]. DE is a population-based stochastic search engine, which is efficient and effective for global optimization in the continuous domain. It uses mutation, crossover, and selection operators at each generation to move its population toward the global optimum. Superior performance of DE, in terms of accuracy and fast operation, has been verified in many reported research works [21, 22]. The algorithm (see Figure 1) consists of five main stages: initialization, mutation, crossover, selection, and stopping condition verification. Given that the problem at hand is to minimize a cost function $f(X)$, where the number of decision variables is D, we can describe each stage of DE in detail.

(1) Initialization. DE begins the search process by randomly generating NP number of D-dimensional parameter vectors $X_{i,g}$ where $i = 1, 2, \ldots, \mathrm{NP}$ and g represents the current generation. In DE algorithm, NP does not change during the optimization process. Moreover, the initial population (at $g = 0$) ought to cover the entire search space in a uniform manner. Thus, we can simply generate these individuals as follows:

$$X_{i,0} = \mathrm{LB} + \mathrm{rand}[0,1] \times (\mathrm{UB} - \mathrm{LB}), \quad (11)$$

where $X_{i,0}$ is the decision variable i at the first generation. rand$[0, 1]$ denotes a uniformly distributed random number

Table 1: Influencing factors (IF) of historical data.

Factors	Description	Notation
IF1	The diameter through which 10% of the total soil mass passes	D_{10} (μm)
IF2	The diameter through which 15% of the total soil mass passes	D_{15} (μm)
IF3	Void ratio	e
IF4	The fines content of the total soil mass	FC (%)
IF5	The coefficient of gradation	C_z
IF6	The coefficient of uniformity	C_u
IF7	Water-to-cement ratio of grout	w/c

between 0 and 1. LB and UB are two vectors of lower bound and upper bound for any decision variable.

(2) Mutation. A vector in the current population (or parent) is called a target vector. Hereafter, the terms "parent" and "target vector" are used interchangeably. For each target vector, a mutant vector is produced via the following equation:

$$V_{i,g+1} = X_{r1,g} + F\left(X_{r2,g} - X_{r3,g}\right), \tag{12}$$

where $r1$, $r2$, and $r3$ are three random indexes lying between 1 and NP. These three randomly chosen integers are also selected to be different from the index i of the target vector. F denotes the mutation scale factor, which controls the amplification of the differential variation between $X_{r2,g}$ and $X_{r3,g}$. $V_{i,g+1}$ represents the newly created mutant vector.

(3) Crossover. The purpose of the crossover stage is to diversify the current population by exchanging components of target vector and mutant vector. In this stage, a new vector, named as trial vector, is created. The trial vector is also called the offspring. The trial vector can be formed as follows:

$$U_{j,i,g+1} = \begin{cases} V_{j,i,g+1}, & \text{if } \text{rand}_j \le \text{Cr or } j = \text{rnb}(i) \\ X_{j,i,g}, & \text{if } \text{rand}_j > \text{Cr}, \ j \ne \text{rnb}(i), \end{cases} \tag{13}$$

where $U_{j,i,g+1}$ is the trial vector. j denotes the index of element for any vector. rand_j is a uniform random number lying between 0 and 1. Cr is the crossover probability, which needs to be determined by users. rnb(i) is a randomly chosen index of $\{1, 2, \ldots, \text{NP}\}$ which guarantees that at least one parameter from the mutant vector ($V_{j,i,g+1}$) is copied to the trial vector ($U_{j,i,g+1}$).

(4) Selection. In this stage, the trial vector is compared to the target vector. If the trial vector can yield a lower objective function value than its parent, then the trial vector replaces the position of the target vector. The selection operator is expressed in the following way:

$$X_{i,g+1} = \begin{cases} U_{i,g} & \text{if } f\left(U_{i,g}\right) \le f\left(X_{i,g}\right) \\ X_{i,g} & \text{if } f\left(U_{i,g}\right) > f\left(X_{i,g}\right). \end{cases} \tag{14}$$

(5) Stopping Criterion Verification. The optimization process terminates if the stopping criterion is met. The type of this condition can be specified by users. Commonly, maximum generation ($G_{\max}$) can be used as the stopping condition. When the optimization process terminates, the final optimal solution is readily presented.

Table 2: Descriptive statistics of historical data.

Factors	Notation	Maximum	Minimum	Mean	Standard deviation
IF1	D_{10} (μm)	85	0.1	20.07	20.83
IF2	D_{15} (μm)	125	0.2	32.01	27.54
IF3	e	1.04	0.35	0.71	0.13
IF4	FC (%)	99.6	6.9	41.67	29.94
IF5	C_z	27.84	0.02	2.44	2.42
IF6	C_u	581.82	2.11	26.38	63.12
IF7	w/c	4.65	3.34	4*	0.53

*Median.

2.4. Historical Data. The database used in this research consists of 240 in situ permeation grouting cases [11] for sandy silt soil collected from highway and mass rapid transportation projects. A mixture of microfine cement and microslag in equal proportions was utilized as the injected grout. The diameters through which 95%, 90%, and 85% of the total grout pass are 7.4 μm, 6.4 μm, and 4.5 μm, respectively. Moreover, the diameter through which 70% of the total grout passes is less than 1 μm. Thus, the grout is considered to be a seminanometer material.

Recent research works have pointed out that other parameters of the soil and the grout (namely, the fines content of the total soil mass, the water-to-cement ratio of the grout, the void size in soil, the coefficient of uniformity, and the coefficient of gradation of the soil) can be useful for the prediction process [4, 9–11]. In this study, seven influencing factors are considered to estimate the outcome of a grouting activity (see Table 1). For each data case, the corresponding output is either +1, which means that the grouting is successful, or −1, which indicates unsuccessful grouting. In our research, inherited from the previous research work [11], a grouting process is considered to be successful if the injected grout is at least two times the volume of the void space under the split pressure. Table 2 provides descriptive statistics of influencing factors of the historical data.

Before being used, the data set has been normalized into a $[0, 1]$ range which helps prevent the situation in which inputs with greater magnitudes dominate those with

Table 3: Historical data.

Case	IF1	IF2	IF3	IF4	IF5	IF6	IF7	Desired output
1	2.987	4.443	0.373	10.330	16.935	456.023	4.650	−1
2	1.798	2.446	0.378	8.754	20.579	563.269	4.650	−1
3	6.128	11.058	0.367	15.614	13.707	205.009	4.000	−1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
238	9.015	18.046	0.360	23.957	16.935	228.777	4.000	−1
239	20.985	31.026	0.355	14.409	11.677	126.748	4.000	+1
240	11.986	27.032	0.360	19.878	14.125	145.878	4.000	+1

Note: output = −1: unsuccessful grouting. Output = +1: successful grouting.

smaller magnitudes [23–25]. The historical data is illustrated in Table 3. The function used for normalizing data is shown as follows:

$$X_n = \frac{X_o - X_{\min}}{X_{\max} - X_{\min}}, \tag{15}$$

where X_n is the normalized data. X_o is the original data. $X_{\max}$ and $X_{\min}$ denote the maximum and minimum values of the data, respectively.

3. The Proposed Groutability Estimation Model Based on Autotuning Support Vector Machines (GE-SVM$_{AT}$)

This section of the paper describes the proposed groutability prediction method, named as GE-SVM$_{AT}$, in detail. The model (see Figure 2) is constructed by a hybridization of the SVM and DE optimization algorithm. The GE-SVM$_{AT}$ employs SVM as an AI technique for carrying out classification tasks. In addition, the new approach utilizes the DE algorithm for automatically identifying the optimal values of SVM's tuning parameters. The establishment of this prediction model is dependent on two tuning parameters: the penalty parameter (c) and the kernel function parameter (σ). Equipped with the DE optimization technique, the SVM algorithm can automatically adapt its parameters according to different learning circumstances without human intervention.

(1) Input Data. The database of 240 in situ grouting cases is divided into training set (90%) and testing set (10%). The training data cases are used to establish the prediction model. Meanwhile, testing data cases are used to verify the predictive performance of the proposed GE-SVM$_{AT}$.

(2) Tuning Parameter Initialization. The aforementioned tuning parameters of the model are randomly generated within the range of lower and upper boundaries. In this study, the lower and upper boundaries of the tuning parameters are 10^{-5} and 10^{5}, respectively. Moreover, the equation used for generating the model tuning parameters can be shown as follows:

$$X_{i,0} = \mathrm{LB} + \mathrm{rand}\,[0, 1] \times (\mathrm{UB} - \mathrm{LB}), \tag{16}$$

where $X_{i,0}$ is the tuning parameter i at the first generation. rand$[0, 1]$ denotes a uniformly distributed random number

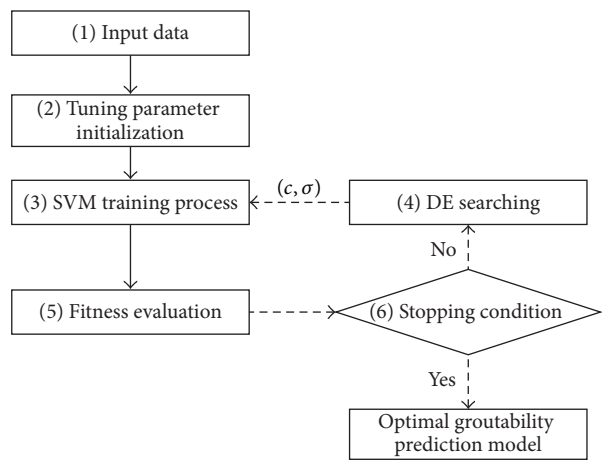

Figure 2: Groutability estimation model based on autotuning support vector machine.

between 0 and 1. LB and UB are two vectors of lower bound and upper bound for any parameter.

(3) SVM Training Process. In this step, the SVM algorithm is deployed to learn the decision boundary to separate the input data into two classes of groutability (−1 and +1). It is noted that the class label −1 represents a failed grouting process; the class label +1 indicates a successful grouting activity.

(4) DE Searching. The DE optimization approach is applied to explore the various combinations of the tuning parameters (c and σ). At each generation, the optimizer carries out the mutation, crossover, and selection processes to guide the population to the optimal solution. By evaluating the fitness of each individual, the algorithm discards inferior combinations of c and σ and permits robust combinations of these parameters to be passed on the next generations.

(5) Fitness Evaluation. In GE-SVM$_{AT}$, in order to determine the optimal set of tuning parameters, the following objective function is used in the step of fitness function evaluation:

$$F_{\text{fitness}} = \frac{1}{\mathrm{AR}_{\mathrm{TR}} + \mathrm{AR}_{\mathrm{VA}}}, \tag{17}$$

where $\mathrm{AR}_{\mathrm{TR}}$ and $\mathrm{AR}_{\mathrm{VA}}$ denote the classification accuracy rates for the training and validating set, respectively. It is

Table 4: The prediction result of GE-SVM$_{AT}$ for the first data fold.

Case	IF1	IF2	IF3	IF4	IF5	IF6	IF7	Actual result	Predicted result
1	0.0021	0.0024	0.2798	0.0182	0.9274	0.9935	0.5038	−1	−1
2	0.1048	0.1026	0.0111	0.2391	0.4918	0.6257	1.0000	−1	−1
3	0.0000	0.0000	0.8934	0.0193	0.9419	0.9935	1.0000	−1	−1
4	0.1519	0.2468	0.0110	0.1602	0.4482	0.1942	0.5038	+1	+1
5	0.5995	0.5593	0.0041	0.0300	0.4337	0.1079	1.0000	+1	+1
6	0.0236	0.0192	0.0140	0.0127	0.8693	0.9935	1.0000	−1	−1
7	0.0094	0.0088	0.0807	0.0131	0.8548	0.9924	1.0000	−1	−1
8	0.2462	0.3189	0.0062	0.0906	0.5208	0.1877	0.5038	+1	+1
9	0.1284	0.1907	0.0127	0.1162	0.5789	0.3074	0.5038	+1	+1
10	0.8469	0.7196	0.0045	0.0369	0.3611	0.0367	1.0000	+1	+1
11	0.0836	0.0865	0.0203	0.1790	0.5498	0.3139	0.5038	+1	−1
12	0.0001	0.0002	1.0000	1.0000	0.7096	0.9342	0.5038	−1	−1
13	0.1284	0.1747	0.0128	0.1606	0.4627	0.2567	0.5038	+1	+1
14	0.2462	0.2468	0.0070	0.0815	0.4192	0.2147	0.5038	+1	+1
15	0.0130	0.0144	0.0402	0.0290	0.7967	0.9827	1.0000	−1	−1
16	0.0130	0.0112	0.0452	0.0156	0.8403	0.9881	0.5038	−1	−1
17	0.0931	0.0946	0.0181	0.2299	0.4192	0.2524	1.0000	+1	+1
18	0.8469	0.6955	0.0052	0.0221	0.2739	0.0410	0.5038	+1	+1
19	0.5642	0.5673	0.0097	0.0461	0.4046	0.1003	0.5038	+1	+1
20	0.0389	0.0417	0.0309	0.0000	0.5353	0.7843	0.5038	−1	−1
21	0.2226	0.2548	0.0064	0.1039	0.3756	0.2082	0.5038	+1	+1
22	0.8233	0.6394	0.0052	0.0182	0.3030	0.0496	0.5038	+1	+1
23	0.2108	0.2468	0.0066	0.1188	0.4918	0.1942	1.0000	+1	+1
24	0.8704	0.7516	0.0050	0.0289	0.3175	0.0270	0.5038	+1	+1

noted that, in the proposed model, the ratio of validating and training cases is set as 1/5. The classification accuracy rate is calculated as the number of correct classification divided by the number of all data instances within a data set.

The fitness function represents the trade-off between model generalization and model complexity. It is worth noticing that a good prediction of the training set may reflect the model complexity. However, a complex model tends to suffer from overfitting [26]. Overfitting arises when a model predicts the training set very well but performs poorly on the new data set. In order to mitigate the undesirable effect of overfitting, prediction performance of the validating data should be taken into account. Therefore, the proposed fitness function can help identify the model that features the balance of minimizing training error and generalization property.

(6) Stopping Condition. The DE's optimization process terminates when the maximum number of generation (G_{max}) is achieved. If the stopping condition is not met, the DE algorithm will continue its searching process. When the program terminates, the optimal set of tuning parameters has been successfully identified and GE-SVM$_{AT}$ is ready to predict new input patterns.

4. Experimental Results

The proposed GE-SVM$_{AT}$ uses 216 data cases for model construction and 24 data cases for testing. This means that 90% of the historical data is used for constructing the prediction model. Meanwhile, 10% of the historical data is reserved for testing process. The groutability results of testing data points are unknown for the model. Therefore, the testing data has the role of new grouting cases which need to be predicted and they can be employed to verify the trained model.

However, due to the randomness in selecting testing cases, the evaluation of model performance can be biased. To avoid such issue, the whole data set (containing 240 cases) is divided into ten data folds in which each fold in turn serves as testing cases, and the model performance can be evaluated via average predictive results of the ten folds. This process is the tenfold cross-validation which is commonly used for verifying model performance [26–28]. Since all of the subsamples are mutually exclusive, this approach can estimate how accurately GE-SVM$_{AT}$ performs in practice.

Moreover, it is noted that the proposed model utilized the same number of DE generations (G_{max}) and objective function ($F_{fitness}$) for each of the 10 subsets. The parameter G_{max} needs to be sufficient for the DE algorithm to converge; in our research, G_{max} is set experimentally to be 300. Details of the GE-SVM$_{AT}$ prediction results for the first fold are provided in Table 4. Moreover, confusion matrices for all of testing data folds are employed for visualizing the performance of GE-SVM$_{AT}$ (see Table 5). Each column of the matrices represents the instances in a predicted class. Meanwhile, each row indicates the instances in an actual class. Performance of the new groutability estimation method can

Table 5: The result comparison for AI methods using classification accuracy rates (%).

Fold	1	2	3	4	5	6	7	8	9	10	Average
GE-SVM$_{AT}$											
Train	91.7	92.6	92.6	93.1	92.6	95.4	92.6	93.5	93.5	92.1	93.0
Test	95.8	91.7	95.8	91.7	91.7	91.7	91.7	91.7	91.7	100.0	93.3
ANN											
Train	93.5	93.5	93.1	93.1	90.3	92.6	92.6	91.2	92.1	91.2	92.3
Test	91.7	91.7	87.5	87.5	83.3	91.7	87.5	83.3	87.5	100.0	89.2
SVM											
Train	99.5	100.0	99.5	99.1	99.5	100.0	99.5	99.5	100.0	99.5	99.6
Test	70.8	83.3	75.0	58.3	62.5	75.0	83.3	70.8	75.0	79.2	73.3

Table 6: The result comparison for AI and formula-based approaches.

Method	Prediction accuracy (%)
GE-SVM$_{AT}$	93.3
ANN	89.2
SVM	73.3
Burwell [6]	48.7
Krizek et al. [7]	45.8
Incecik and Ceren [5]	45.4

be evaluated by using the data in the matrices. Observed from the confusion matrices, the proposed GE-SVM$_{AT}$ obtains very high numbers of true positives and negatives. The average true positives and negatives of the proposed method in one fold are 15.1 and 7.3, respectively. Moreover, the numbers of false positives and negatives yielded by GE-SVM$_{AT}$ are remarkably low. The average false positives and negatives in one fold of GE-SVM$_{AT}$ are 0.04 and 0.12, respectively. These results indicate a very robust prediction capability of the new approach.

Furthermore, to better demonstrate the capability of the proposed GE-SVM$_{AT}$, its performance is compared to results acquired from other benchmark approaches including the SVM and ANN algorithms. For these benchmark approaches, similar to the GE-SVM$_{AT}$, 216 data cases are used for model construction and 24 data cases are reserved for model testing.

When using the ANN algorithm, it is needed to specify the number of hidden layers, the number of neurons in the hidden layer, the learning rate, and the number of training epochs [14]. These parameters of ANN are generally selected via repetitive trial and error processes [29]. The network configuration is described as follows: the number of hidden layers is set to be 1; the number of neurons in the hidden layer is 7; and the number of training epochs is selected to be 2000. The back-propagation approach [30] is used as the method for training the ANN model. For the SVM algorithm, as suggested by [25], the penalty parameter is 1 and the kernel function parameter is set to be $1/D$, where $D = 7$ is the dimension of the input data.

Table 6 provides the result obtained from the tenfold cross-validation of the GE-SVM$_{AT}$ and other benchmark methods. The classification accuracy rates of the GE-SVM$_{AT}$, ANN, and SVM for training data are 93.0%, 92.2%, and 99.6%, respectively. Meanwhile, the classification accuracy rates of the GE-SVM$_{AT}$, ANN, and SVM for testing data are 93.3%, 89.2.2%, and 73.3%, respectively.

In addition, the average prediction accuracy of AI methods, obtained from the tenfold cross-validation process, and the prediction results of formula-based approaches are provided in Table 7. The main difference between formula-based methods and AI approaches is that formula-based methods only rely on information of the grain size of the soil and the grout. Meanwhile, in addition to that information, AI approaches take into account other influencing factors of the grouting process. Additionally, it is worth noticing that, in Table 7, the capability of groutability prediction approaches is quantified by classification accuracy rates.

Thus, it is obvious that AI methods can deliver much better prediction accuracy compared to formula-based approaches. The experiment has also proven that the integration of SVM and DE can improve the accuracy of the SVM method. Furthermore, the performance of the ANN and SVM methods is inferior to that of GE-SVM$_{AT}$. GE-SVM$_{AT}$ achieves the best prediction outcome for testing data. Moreover, it can be observed that the proposed model has successfully overcome the issue of overfitting since it yields a relatively balanced performance between training and testing data sets.

5. Conclusion

This research has presented and verified a new groutability prediction method, named as GE-SVM$_{AT}$, to assist construction engineers in appraising the possibility of grouting processes that employ microfine cement grouts. The proposed approach was developed by a hybridization of the SVM and DE algorithms. GE-SVM$_{AT}$ utilizes the SVM technique to classify high-dimensional input data so that the model can yield prediction outcomes whenever new input patterns are available. Meanwhile, the DE searching algorithm is implemented to select the most appropriate tuning parameters. Therefore, this mechanism eliminates the need for experience or trial and error process in SVM's parameter setting. Consequently, the newly established model has the ability to operate automatically without human intervention and domain knowledge. Performance comparison has shown

Table 7: GE-SVM$_{AT}$'s confusion matrices for testing data prediction.

		Predicted	
		Success	Failure
	Fold 1		
Actual	Success	14	1
	Failure	0	9
	Fold 2		
Actual	Success	17	1
	Failure	1	5
	Fold 3		
Actual	Success	13	0
	Failure	1	10
	Fold 4		
Actual	Success	13	0
	Failure	2	9
	Fold 5		
Actual	Success	12	0
	Failure	2	10
	Fold 6		
Actual	Success	16	0
	Failure	2	6
	Fold 7		
Actual	Success	18	1
	Failure	1	4
	Fold 8		
Actual	Success	14	1
	Failure	1	8
	Fold 9		
Actual	Success	19	0
	Failure	2	3
	Fold 10		
Actual	Success	15	0
	Failure	0	9

the strong potential of GE-SVM$_{AT}$ as an alternative for groutability prediction in the construction industry.

The current approach GE-SVM$_{AT}$ is established by historical grouting cases performed in sandy silt soil. Even though experimental results have shown that the proposed method can deliver outstanding prediction results for this type of soil, more historical grouting cases in which the targeted objects of the grouting process involve different soil types should be included to improve the generalization of the GE-SVM$_{AT}$. Thus, this task can be a promising future direction of this research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] S. Zebovitz, R. J. Krizek, and D. K. Atmatzidis, "Injection of fine sands with very fine cement grout," *Journal of Geotechnical Engineering*, vol. 115, no. 12, pp. 1717–1733, 1989.

[2] C. Butrón, G. Gustafson, Å. Fransson, and J. Funehag, "Drip sealing of tunnels in hard rock: a new concept for the design and evaluation of permeation grouting," *Tunnelling and Underground Space Technology*, vol. 25, no. 2, pp. 114–121, 2010.

[3] S. Perret, K. H. Khayat, E. Gagnon, and J. Rhazi, "Repair of 130-year old masonry bridge using high-performance cement grout," *Journal of Bridge Engineering*, vol. 7, no. 1, pp. 31–38, 2002.

[4] E. Tekin and S. O. Akbas, "Artificial neural networks approach for estimating the groutability of granular soils with cement-based grouts," *Bulletin of Engineering Geology and the Environment*, vol. 70, no. 1, pp. 153–161, 2011.

[5] M. Incecik and I. Ceren, "Cement grouting model tests," *Bulletin of the Technical University of Istanbul*, vol. 48, pp. 305–317, 1995.

[6] E. B. Burwell, "Cement and clay grouting of foundations: practice of the corps of engineering," *Journal of the Soil Mechanics and Foundations Division*, vol. 84, pp. 1551/1–1551/22, 1958.

[7] R. J. Krizek, H.-J. Liao, and R. H. Borden, "Mechanical properties of microfine cement/sodium silicate grouted sand," in *Proceedings of the ASCE Specialty Conference on Grouting, Soil Improvement and Geosynthetics*, pp. 688–699, February 1992.

[8] C. L. Huang, J. C. Fan, and W. J. Yang, "A study of applying microfine cement grout to sandy silt soil," *Sino-Geotech*, vol. 111, pp. 71–82, 2007.

[9] S. Akbulut and A. Saglamer, "Estimating the groutability of granular soils: a new approach," *Tunnelling and Underground Space Technology*, vol. 17, no. 4, pp. 371–380, 2002.

[10] H. G. Ozgurel and C. Vipulanandan, "Effect of grain size and distribution on permeability and mechanical behavior of acrylamide grouted sand," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 131, no. 12, pp. 1457–1465, 2005.

[11] K.-W. Liao, J.-C. Fan, and C.-L. Huang, "An artificial neural network for groutability prediction of permeation grouting with microfine cement grouts," *Computers and Geotechnics*, vol. 38, no. 8, pp. 978–986, 2011.

[12] Y.-L. Chen, R. Azzam, T. M. Fernandez-Steeger, and L. Li, "Studies on construction pre-control of a connection aisle between two neighbouring tunnels in Shanghai by means of 3D FEM, neural networks and fuzzy logic," *Geotechnical and Geological Engineering*, vol. 27, no. 1, pp. 155–167, 2009.

[13] A. Kalinli, M. C. Acar, and Z. Gündüz, "New approaches to determine the ultimate bearing capacity of shallow foundations based on artificial neural networks and ant colony optimization," *Engineering Geology*, vol. 117, no. 1-2, pp. 29–38, 2011.

[14] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering*, Taylor and Francis, 2006.

[15] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization," *Neural Networks*, vol. 22, no. 10, pp. 1448–1462, 2009.

[16] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[17] K. Gopalakrishnan and S. Kim, "Support vector machines approach to HMA stiffness prediction," *Journal of Engineering Mechanics*, vol. 137, no. 2, pp. 138–146, 2010.

[18] M.-Y. Cheng, N.-D. Hoang, and Y.-W. Wu, "Hybrid intelligence approach based on LS-SVM and Differential Evolution

for construction cost index estimation: a Taiwan case study," *Automation in Construction*, vol. 35, pp. 306–313, 2013.

[19] P. Samui, "Slope stability analysis: a support vector machine approach," *Environmental Geology*, vol. 56, no. 2, pp. 255–267, 2008.

[20] K. C. Lam, E. Palaneeswaran, and C.-Y. Yu, "A support vector machine model for contractor prequalification," *Automation in Construction*, vol. 18, no. 3, pp. 321–329, 2009.

[21] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution a Practical Approach to Global Optimization*, Springer, 2005.

[22] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[23] H.-L. Chen, B. Yang, G. Wang et al., "A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1348–1359, 2011.

[24] M.-Y. Cheng, A. F. V. Roy, and K.-L. Chen, "Evolutionary risk preference inference model using fuzzy support vector machine for road slope collapse prediction," *Expert Systems with Applications*, vol. 39, no. 2, pp. 1737–1746, 2012.

[25] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Tech. Rep., Department of Computer Science, National Taiwan University, 2010.

[26] C. Bishop, *Pattern Recognition and Machine Learning*, Springer Science+Business Media, Singapore, 2006.

[27] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[28] P. Zhang, "Model selection via multifold cross validation," *The Annals of Statistics*, vol. 21, pp. 299–313, 1993.

[29] J.-S. Chou, C.-K. Chiu, M. Farfoura, and I. Al-Taharwa, "Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques," *Journal of Computing in Civil Engineering*, vol. 25, no. 3, pp. 242–253, 2011.

[30] S. J. Russell and P. Norvig, *Artificial Intelligence a Modern Approach*, Prentice Hall, Person Education, 2nd edition, 2003.