## Electronic Supplementary Materials

For https://doi.org/10.1631/jzus.A2400568

# Machine learning for soil parameter inversion enhanced with Bayesian optimization

Anfeng HU[1], Chi WANG[1,2], Senlin XIE[1], Zhirong XIAO[3], Tang LI[1], Ang XU[1]

[1]*Research Center of Coastal and Urban Geotechnical Engineering, Zhejiang University, Hangzhou 310058, China*
[2]*MOE Key Laboratory of Soft Soils and Geoenvironmental Engineering, Zhejiang University, Hangzhou 310058, China*
[3]*School of Civil Engineering and Architecture, Zhejiang University of Science & Technology, Hangzhou 310023, China*

# Table

Table S1 The geometric and physical properties of the different soil layers

| Layer | Point | $\gamma$/kN·m$^{-3}$ | $\gamma_{sat}$/kN m$^{-3}$ | $e_0$ | $v$ | $h$/m |
|---|---|---|---|---|---|---|
| FL | SP-A | 16 | / | 0.95 | 0.25 | 2/4/3/2 |
| (FL1/FL2/FL3/FL4) | SP-B | | | | | 4/1/4/2 |
| MCL | SP-A | / | 16.6 | 2.14 | 0.3 | 2.3 |
| | SP-B | | | | | 1.7 |
| SCL | SP-A | / | 18.6 | 0.95 | 0.3 | 3.7 |
| | SP-B | | | | | 8.5 |
| RSCSL | SP-A | / | 18.8 | 0.96 | 0.3 | 10.3 |
| | SP-B | | | | | 6.8 |

Table S2 The range of soil constitutive parameters in the FEM script

| Layer | Point | $E$/MPa | $v$ | $\lambda\times10^{-1}$ | $\kappa\times10^{-2}$ | $M$ | $e_0$ | $k$×10$^{-5}$/ m d$^{-1}$ |
|---|---|---|---|---|---|---|---|---|
| FL | SP-A | 15 | 0.25 | | | | | |
| | SP-B | 15 | 0.25 | | | | | |
| MCL | SP-A | 0.5~0.7 | 0.3 | 1.7~2.1 | 2.03~2.51 | 0.65~0.89 | 2.14 | 2.5~6.7 |
| | SP-B | 0.4~0.6 | 0.3 | 2.0~2.4 | 1.8~2.2 | 0.6~0.8 | 2.14 | 2.2~5 |
| SCL | SP-A | 3~5 | 0.3 | 1.3~1.7 | 1.33~1.81 | 1.6 | 0.95 | 20~53.6 |
| | SP-B | | | 1.6~2.0 | 1.1~1.5 | 1.6 | 0.95 | 17.6~40 |
| RSCSL | SP-A | 10~14 | 0.3 | 0.9~1.3 | 0.63~1.11 | 1.8 | 0.96 | 30~80.4 |
| | SP-B | | | 1.2~1.6 | 0.4~0.8 | 1.8 | 0.96 | 26.4~60 |

Table S2 displays the soil constitutive parameters considered in the FEM script. Through the independent combination of various soil parameters, such as the elastic modulus $E$, normal

consolidation line slope $\lambda$, the unloading rebound line slope $\kappa$, and permeability coefficient $k$, within the soil strata, a substantial dataset of settlement measurements is procured. The failure stress ratio $M$ for SCL and RSCSL is derived from the effective friction angle $\varphi'$ obtained from the geological survey report. Other parameters, including the initial yield surface dimension $\beta$, the rheological stress ratio $R$, and the intercept of the regular consolidation line $e_1$, are set to 1, 1, and 3, respectively. The initial yield surface dimension $a_0$ is automatically calculated based on the initial stress and void ratio.

Table S3 The hyperparameters to be optimized for each ML-based model

| Model | Optimized hyperparameter | | | |
|---|---|---|---|---|
| DT | Max depth | Min samples leaf | Min samples split | |
| RF | Max depth | Min samples leaf | Min samples split | Estimators |
| SVR | $C$ | kernel | $\gamma_G$ | |
| DNN | N_neuronsi(i=1,2,3) | Activation | Learning rate | Dropout |
| | Epochs | Batch size | | |
| 1D-CNN | N_kerneli(i=1,2,3) | N_Dense | Activation | Learning rate |
| | Dropout | Epochs | Batch size | |

The correct pruning strategy is pivotal in optimizing the DT model. The optimization of the DT model focuses on three hyperparameters: Max depth, Min samples leaf, and Min samples split. The Max depth parameter constrains the maximum depth of the tree, while the Min samples leaf and Min samples split, respectively, restrict the minimum number of samples required for node splitting and leaf nodes. Optimizing these parameters can effectively prevent overfitting in the DT model.

Compared to DT, the RF model introduces a hyperparameter called Estimators, which denotes the number of decision trees within the forest. Typically, increasing the number of Estimators enhances model performance. However, it should be noted that the model's accuracy tends to stabilize or exhibit fluctuations beyond a specific threshold. Furthermore, incorporating additional Estimators results in heightened computational and memory demands, thereby necessitating a judicious balance between the enhancement of model performance and the complexity of training.

In the SVR model, selecting the regularization constant $C$, kernel function, and $\gamma_G$ are pivotal hyperparameters for optimization. The regularization constant $C$ notably impacts the model's generalization performance, while the kernel function is pivotal in determining the suitability of sample mapping to the feature space. In the case of the model employing the prevalent Gaussian kernel function, the hyperparameter $\gamma_G$ defines the impact of individual samples on the overall classification hyperplane. Its value dictates whether the model tends towards a smoother or more intricate representation.

The hyperparameters required for constructing a Deep Neural Network (DNN) model include the network structure, number of epochs, batch size, and learning rate, among others. The network architecture encompasses the model's layering, the number of nodes in each layer, and the activation functions used. Given the scale of the dataset, this study establishes a fully connected neural network with three hidden layers, adjusting the complexity of the network structure by varying the activation functions and the number of neurons in each connecting layer. The learning rate refers to the step size at which the model updates weights during the gradient descent algorithm. The batch size denotes the number of samples used for training during each model update. The number of epochs represents the number of times the model iterates over the training set, and optimizing this parameter aims to ensure thorough model training while avoiding overfitting. Dropout is a commonly used regularization method that involves randomly deactivating a certain proportion of neurons to prevent model overfitting.

This study employs a classic 1D-CNN consisting of three convolutional layers. Considering the model's complexity and computational cost, the kernel size within each convolutional layer is set to 9, and the stride, which determines the speed at which the convolutional filters move across the input data, is set to 1 to ensure comprehensive feature extraction from the input data. The complexity of the convolutional model architecture is controlled by adjusting the number of kernels in each convolutional layer. Additionally, the number of neurons in the Flatten layer, learning rate, Dropout, batch size, and training epochs are optimized for training. The Dense layer is utilized to process the two-dimensional output after global pooling, and the number of neurons within it influences the model's decision-making capability. The batch size represents the quantity of data processed by the entire network at once, and optimizing this parameter aims to strike a balance between maximizing GPU utilization and ensuring good training dynamics.

Table S4 The range of hyperparameter values used for sensitivity analysis in the RF model

| RF | Estimators | Max depth | Min samples leaf | Min samples split |
|---|---|---|---|---|
| range | (1,101) | (1,51) | (1,51) | (2,52) |
| step | 10 | 5 | 5 | 5 |

Table S5 The range of hyperparameter values used for sensitivity analysis in the 1D-CNN model

| Hyperparameter | range |
|---|---|
| N_kerneli(i=1,2,3) | [1,11,21,31,41,51,61,71,81,91,101] |
| N_Dense | [1,11,21,31,41,51,61,71,81,91,101] |
| Learning rate | [1e-6,1e-5,1e-4,1e-3,1e-2,1e-1] |
| Batch size | [8,16,32,48,64,128,256,512] |
| Epochs | range from 1 to 2000 |

Table S6 The hyperparameter search range for SVR models on the SP-A and SP-B datasets

| SVR | SP-A | | | SP-B | | |
|---|---|---|---|---|---|---|
| | $C$ | kernel | $\gamma_G$ | $C$ | kernel | $\gamma_G$ |
| Model1 | (0.1,5) | [Linear, RBF] | (0.001,0.1) | (0.1,5) | [Linear, RBF] | (0.001,0.1) |
| Model2 | (0.1,5) | [Linear, RBF] | (0.0001,0.1) | (0.1,5) | [Linear, RBF] | (0.001,0.1) |
| Model3 | (0.1,10) | [Linear, RBF] | (0.01,1) | (1,100) | [Linear, RBF] | (0.01,1) |
| Model4 | (0.1,10) | [Linear, RBF] | (0.001,1) | (0.1,10) | [Linear, RBF] | (0.001, 0.1) |

Table S7 The hyperparameter search space for the DT, RF, DNN, and 1D-CNN models on SP-A

| Model | Hyperparameter search scope | | | |
|---|---|---|---|---|
| DT | Max depth | Min samples leaf | Min samples split | |
| | (5,30) | (1,10) | (2,15) | |
| RF | Estimators | Max depth | Min samples leaf | Min samples split |
| | (1,50) | (5,30) | (1,10) | (2,15) |
| DNN | N_neuronsi(i=1,2,3) | Activation | Learning rate | Dropout |
| | (8,32) | [ReLU, ELU, Sigmoid] | (1e-4,1e-3) | (0.1,0.5) |
| | Batch size | Epochs | | |
| | (24,64) | (1200,1800) | | |
| 1D-CNN | N_kerneli(i=1,2,3) | N_dense | Activation | Learning rate |
| | (12,48) | (24,64) | [Relu,ELU,Sigmoid] | (1e-4,1e-3) |
| | Dropout | Batch size | Epochs | |
| | (0.1,0.5) | (24,64) | (1200,1800) | |

Table S8 The hyperparameter search space for the DT, RF, DNN, and 1D-CNN models on SP-B

| Model | Hyperparameter search scope | | | |
|---|---|---|---|---|
| DT | Max depth | Min samples leaf | Min samples split | |
| | (5,20) | (1,10) | (2,15) | |
| RF | Estimators | Max depth | Min samples leaf | Min samples split |
| | (1,80) | (5,30) | (1,10) | (2,15) |
| DNN | N_neuronsi(i=1,2,3) | Activation | Learning rate | Dropout |
| | (12,36) | [ReLU, ELU, Sigmoid] | (1e-4,1e-3) | (0.1,0.5) |
| | Batch size | Epochs | | |
| | (24,64) | (1200,1800) | | |
| 1D-CNN | N_kerneli(i=1,2,3) | N_dense | Activation | Learning rate |
| | (8,36) | (24,64) | [Relu,ELU,Sigmoid] | (1e-4,1e-3) |
| | Dropout | Batch size | Epochs | |
| | (0.1,0.5) | (24,64) | (1200,1800) | |

Table S9 The detailed performance metrics of each model during the training and testing phases on the SP-A dataset

| Model | Phase | $R^2$ | MAPE | RMSE | a20-index |
|---|---|---|---|---|---|
| DT | Train | 0.837 | 0.0278 | 0.0295 | 99.64 |
| | Test | 0.814 | 0.0310 | 0.0314 | 99.37 |
| RF | Train | 0.949 | 0.0149 | 0.0160 | 100.00 |
| | Test | 0.923 | 0.0195 | 0.0199 | 100.00 |
| SVR | Train | 0.895 | 0.0264 | 0.0255 | 100.00 |
| | Test | 0.900 | 0.0260 | 0.0239 | 99.81 |
| DNN | Train | 0.961 | 0.0357 | 0.0369 | 100.00 |
| | Test | 0.966 | 0.0359 | 0.0348 | 100.00 |
| 1D-CNN | Train | 0.996 | 0.0127 | 0.0117 | 100.00 |
| | Test | 0.994 | 0.0133 | 0.0116 | 100.00 |

Table S10 The detailed performance metrics of each model during the training and testing phases on the SP-A dataset

| Model | Phase | $R^2$ | MAPE | RMSE | a20-index |
|---|---|---|---|---|---|
| DT | Train | 0.866 | 0.0226 | 0.0223 | 99.44 |
| | Test | 0.821 | 0.0254 | 0.0250 | 99.43 |
| RF | Train | 0.934 | 0.0164 | 0.0154 | 100.00 |
| | Test | 0.907 | 0.0197 | 0.0182 | 100.00 |
| SVR | Train | 0.884 | 0.0245 | 0.0222 | 100.00 |
| | Test | 0.895 | 0.0251 | 0.0208 | 99.93 |
| DNN | Train | 0.974 | 0.0351 | 0.0332 | 100.00 |
| | Test | 0.977 | 0.0356 | 0.0316 | 100.00 |
| 1D-CNN | Train | 0.996 | 0.0093 | 0.0116 | 100.00 |
| | Test | 0.998 | 0.0093 | 0.0092 | 100.00 |

Table S11 The inversion results of soil parameters for ML-based models before and after optimization on the SP-A dataset

| Model | State | Inversion parameters | | | |
|---|---|---|---|---|---|
| | | $\lambda_{\mathrm{MCL}}$ | $\kappa_{\mathrm{MCL}}$ | $M_{\mathrm{MCL}}$ | $k_{\mathrm{MCL}}$ (m/d) |
| DT | Initial | 0.2368 | 0.0317 | 0.5048 | 3.70e-5 |
| | Optimized | 0.2075 | 0.0209 | 0.6514 | 3.10e-5 |
| RF | Initial | 0.1857 | 0.0221 | 0.6730 | 2.98e-5 |
| | Optimized | 0.2019 | 0.0227 | 0.6467 | 3.40e-5 |
| SVR | Initial | 0.1792 | 0.0245 | 0.7394 | 4.27e-5 |
| | Optimized | 0.1963 | 0.0225 | 0.6552 | 4.83e-5 |
| DNN | Initial | 0.2010 | 0.0236 | 0.7752 | 5.59e-5 |

| | | | | | |
|---|---|---|---|---|---|
| | Optimized | 0.2113 | 0.0187 | 0.4350 | 5.31e-5 |
| 1D-CNN | Initial | 0.2021 | 0.0214 | 0.7437 | 3.22e-5 |
| | Optimized | 0.2017 | 0.0205 | 0.5574 | 4.73e-5 |

Table S12 The inversion results of soil parameters for ML-based models before and after optimization on the SP-B dataset

| Model | State | Inversion parameters | | | |
|---|---|---|---|---|---|
| | | $\lambda_{MCL}$ | $\kappa_{MCL}$ | $M_{MCL}$ | $k_{MCL}$ (m/d) |
| DT | Initial | 0.2406 | 0.0182 | 0.7474 | 4.37e-5 |
| | Optimized | 0.2531 | 0.0155 | 0.4972 | 3.25e-5 |
| RF | Initial | 0.2394 | 0.0271 | 0.7838 | 2.96e-5 |
| | Optimized | 0.2343 | 0.0201 | 0.6989 | 2.86e-5 |
| SVR | Initial | 0.2226 | 0.0195 | 0.4180 | 2.31e-5 |
| | Optimized | 0.1963 | 0.0204 | 0.6552 | 4.83e-5 |
| DNN | Initial | 0.2292 | 0.0201 | 0.6461 | 2.72e-5 |
| | Optimized | 0.2267 | 0.0181 | 0.6333 | 2.62e-5 |
| 1D-CNN | Initial | 0.2150 | 0.0195 | 0.6999 | 3.58e-5 |
| | Optimized | 0.2315 | 0.0189 | 0.6379 | 2.60e-5 |

Table S13 The MAE of five ML-based models before and after optimization on the SP-A dataset

| SP-A | DT | RF | SVR | DNN | 1D-CNN |
|---|---|---|---|---|---|
| Initial | 0.09998 | 0.07487 | 0.06407 | 0.06398 | 0.03894 |
| Optimized | 0.05329 | 0.03101 | 0.01406 | 0.01459 | 0.00706 |

Table S14 The MAE of five ML-based models before and after optimization on the SP-B dataset

| SP-B | DT | RF | SVR | DNN | 1D-CNN |
|---|---|---|---|---|---|
| Initial | 0.1296 | 0.10167 | 0.07905 | 0.0569 | 0.0464 |
| Optimized | 0.07238 | 0.04576 | 0.03855 | 0.02544 | 0.01176 |

# Figure

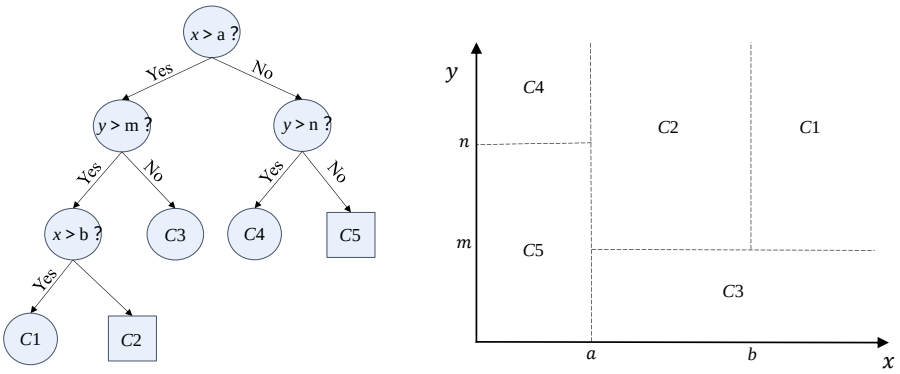


Fig. S1 Visualization of the principles of the regression tree algorithm

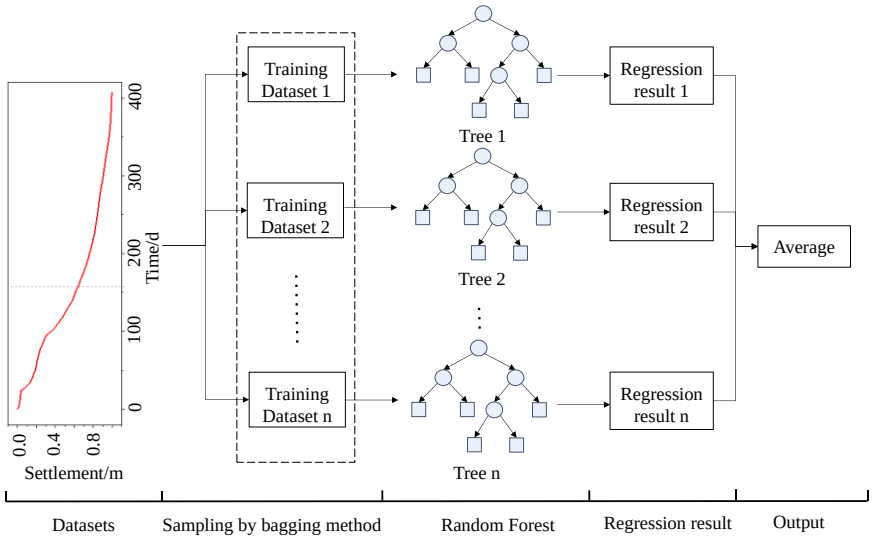


Fig. S2 The visualization of the random forest algorithm

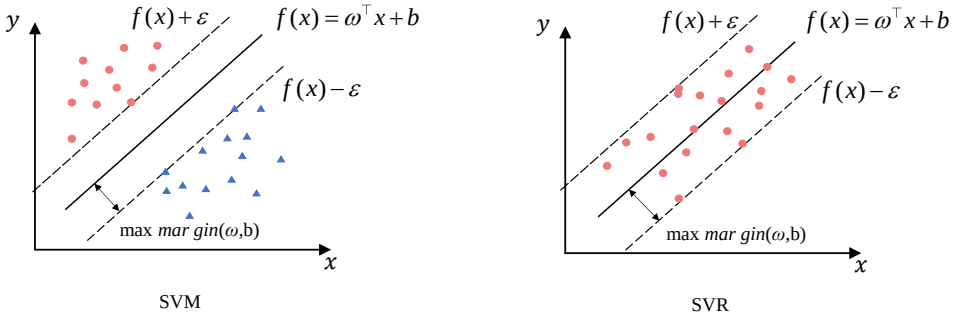


Fig. S3 The visualization of the principles of the SVM and SVR models

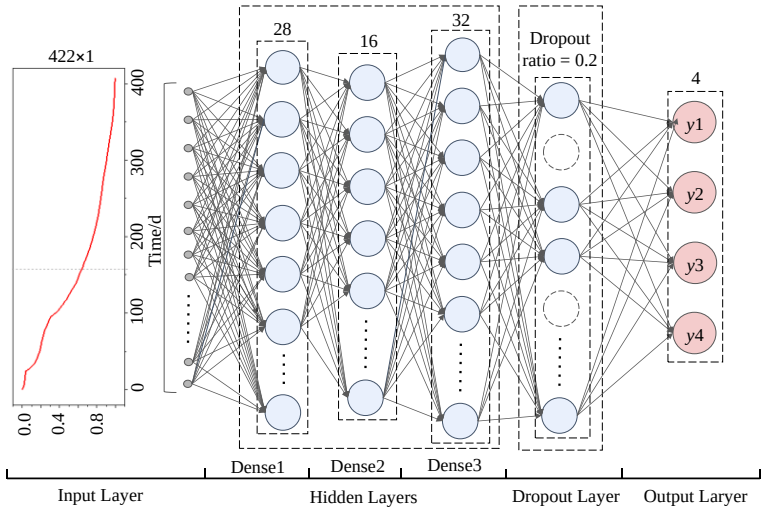


Fig. S4 The DNN model architecture used for monitoring point SP-A

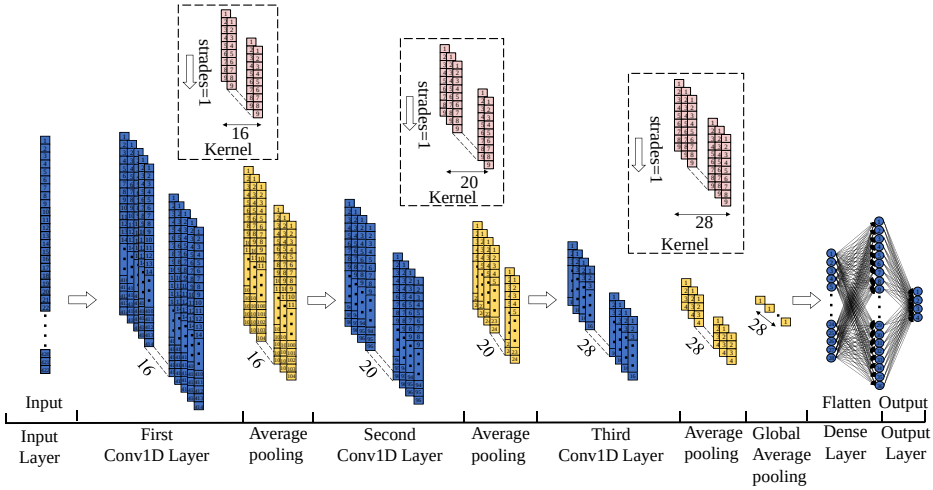


Fig. S5 The 1D-CNN model architecture used for monitoring point SP-A

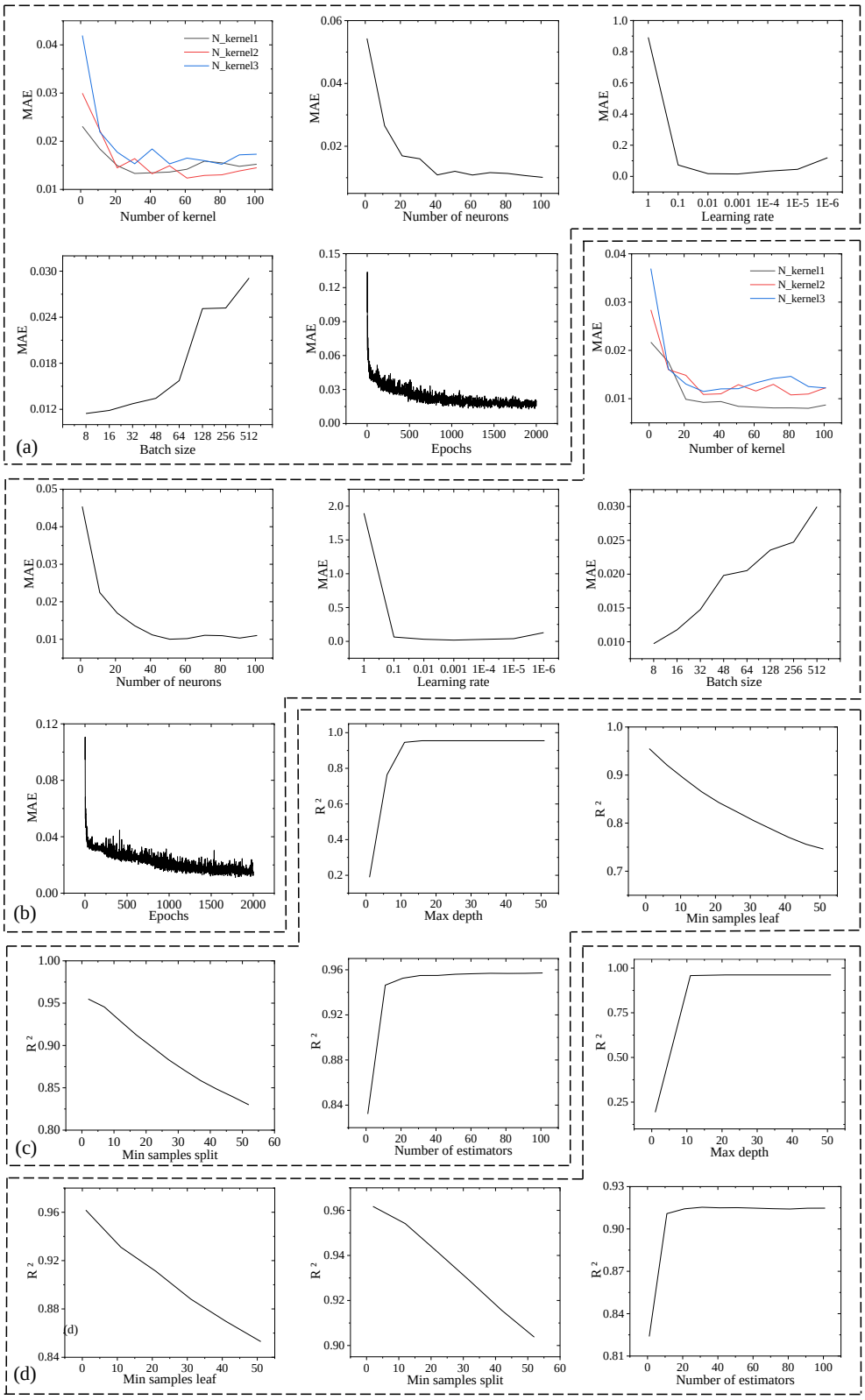


Fig. S6 (a) The results of the hyperparameter sensitivity analysis for the 1D-CNN model on the SP-A dataset; (b) The results of the hyperparameter sensitivity analysis for the 1D-CNN model on the SP-B dataset; (c) The results of the hyperparameter sensitivity analysis for the RF model on the SP-A dataset; (d) The results of the hyperparameter sensitivity analysis for the RF model on the SP-B dataset

# Equation

The expression for the loss function in the CART algorithm is shown as follows

$$C_\alpha(t) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha \mid T \mid \tag{S1}$$

where $t$, $N_t$, $H_t$, and $|T|$ represent the leaf node identifier, the sample quantity of the node, the information entropy of the node, and the total number of leaf nodes, respectively; Parameter $\alpha$ governs the equilibrium between the fit on the left and the model's complexity on the right.

In many practical scenarios, the original sample space may not inherently contain a hyperplane capable of effectively separating two classes of samples. To address this, SVR employs a kernel function to map samples from the original space into a higher-dimensional feature space, thereby enabling linear separability among the samples within this transformed feature space. Commonly utilized kernel functions encompass linear kernels, Gaussian kernels, and others. Eq. (S2) reflects the fundamental principle of the Gaussian kernel function.

$$k(x_i, x_j) = \exp(-\gamma_G \left\| x_i - x_j \right\|^2), \quad \gamma > 0 \tag{S2}$$

where $x_i$ and $x_j$ are input samples, and $\left\| x_i - x_j \right\|$ represents the Euclidean distance between the input samples. $\gamma_G$ represents a parameter of the Gaussian kernel, governing the width of the Gaussian kernel, thereby influencing the model's complexity and fitting capacity.

Eq. (S3) denotes the expression for MAE, which quantifies the proximity between predicted and actual values by computing the mean of their absolute differences.

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - \hat{y}_i \right| \tag{S3}$$

where $y_i$, $\hat{y}_i$, $n$ represent the actual value, the predicted value, and the sample count, respectively.

$$\mathrm{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{\left| y_i - \hat{y}_i \right|}{y_i} \right) \times 100\% \tag{S4}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{S5}$$

$$a20 - index = \frac{N_{\left| \frac{y_{pred} - y_{true}}{y_{true}} \right| \le 0.2}}{N_{total}} \times 100\% \tag{S6}$$

where $n$, $y_i$, and $\hat{y}_i$ denote the sample size, actual settlement, and predicted settlement, respectively. $y_{true}$ and $y_{pred}$ represent the actual values and predicted values, respectively. $N_{\left| \frac{y_{pred} - y_{true}}{y_{true}} \right| \le 0.2}$ denotes the number of samples where the relative error is within 20%, and $N_{total}$ is the total number of samples. $N_{\left| \frac{y_{pred} - y_{true}}{y_{true}} \right| \le 0.2}$ denotes the number of samples where the relative error is within 20%, and $N_{total}$ is the total number of samples.