Archives of Computational Methods in Engineering
https://doi.org/10.1007/s11831-026-10529-3

REVIEW

Probabilistic and Data-Efficient Modelling of Soil Properties: Review, Tool Development and Practical Applications

Geng-Fu He[1,2] · Zhen-Yu Yin[2,3] · Jidong Zhao[4] · Yin-Fu Jin[1]

Received: 26 September 2025 / Revised: 14 December 2025 / Accepted: 15 February 2026

Abstract

Reliable estimation of soil properties is crucial for geotechnical design. While machine learning (ML) offers significant potential in this domain, its application remains challenging for practitioners lacking expertise in data-driven modelling, including data acquisition, model development, evaluation, and deployment. Moreover, conventional data sampling in geotechnical practice is often random and inefficient, leading to excessive costs and suboptimal model performance. This study provides a comprehensive review of ML algorithms applied to soil properties modelling and introduces five representative probabilistic methods: quantile random forest (QRF), variational inference-based evolutionary polynomial regression (VIEPR), ensemble-based support vector regression (ESVR), Bayesian neural networks (BNN), and Gaussian process regression (GPR). These algorithms are integrated into a generic and user-friendly graphical interface platform, ErosMLM, enabling seamless model development, optimization, evaluation, and application through intuitive click-based operations. Furthermore, an active learning strategy is proposed to prioritize informative data acquisition, reducing data demands and associated costs. The platform and strategy are validated through a case study on soil creep index prediction, demonstrating high accuracy, reliability, and practical utility. This work highlights the potential of ML to advance geotechnical modelling while lowering the barrier to its adoption.

Abbreviations

C_α	Creep index
e_0	Initial void ratio
CI	Clay content
PL	Plastic limit
PI	Plastic index
D	Datasets
n	Number of training samples
Y_i	ith output variable
X_i	$(x_1,\ x_2,\ ...,\ x_m)$ ith input variable
m	Dimensions of X_i
w_i	Weight of Y_i
y	Predictions
p	Specified cumulative probability
y_p	Quantile of y corresponding to p
$\boldsymbol{E}$	Exponent matrix
XT	Transformed variables
$\boldsymbol{w}$	Weights/coefficients
$\boldsymbol{\theta}$	Variational parameters
L	Loss function
s	Number of samples
γ	Kernel coefficient
ξ	Slack factors

✉ Zhen-Yu Yin
zhenyu.yin@polyu.edu.hk

✉ Yin-Fu Jin
yinfujin@szu.edu.cn

Geng-Fu He
gengfu.he@connect.polyu.hk

Jidong Zhao
jzhao@ust.hk

[1] State Key Laboratory of Intelligent Geotechnics and Tunnelling, Shenzhen University, Guangdong, China

[2] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, China

[3] State Key Laboratory of Climate Resilience for Coastal Cities, The Hong Kong Polytechnic University, Hong Kong, China

[4] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China

Published online: 15 February 2026

C	Regularization parameter
ε	Allowable deviation
W	Coefficient vector
ϕ	Basis function vector
b	Bias/intercept
p	Dropout probability
r	Bernoulli distribution with probability of p
$k(\cdot,\cdot)$	Kernel function
σ_f^2	Signal variance
l	Length scale
$\boldsymbol{q}$	Quantities of interest
$\boldsymbol{o}$	Observations
$\boldsymbol{K}_{\boldsymbol{oq}}$	Covariance matrix between $\boldsymbol{o}$ and $\boldsymbol{q}$
$\boldsymbol{u}(\boldsymbol{q})$	Conditional mean of $\boldsymbol{q}$
$\boldsymbol{K}(\boldsymbol{q})$	Conditional covariance matrix of $\boldsymbol{q}$
R^2	Coefficient of determination
RI	Reliability index
p	Dropout probability
ntree	Numbers of trees
mtry	Number of features for split
pop	Population size
gen	Number of generations

1 Introduction

As natural products of geological processes, soils exhibit complex properties and involve substantial uncertainty [1], [2]. Experiments in soils can expose their mechanical properties as design parameters [3, 4], but take much time, manpower and budget. Instead, some easily measured indices are often used to estimate these properties through certain transformation models [5–7]. Data-efficient estimation of soil properties can transfer costs into investments [8], such that a reliable and cost-effective design can be attained. Based on certain measurements, numerous empirical equations were proposed to estimate soil properties using low-cost indices [9–13]. However, their functional forms were typically pre-defined, and the corresponding parameters were calibrated by simple curve-fitting without uncertainty quantification. In this context, the accuracy of these equations is constrained by fixed functional forms, estimated properties, and a small set of indices used as predictors, which are also subject to limited repeatability and measurement errors. Once established, equations are difficult to update when new measurements or additional variables arrive, as they generally need to start from scratch rather than being flexibly adapted [14].

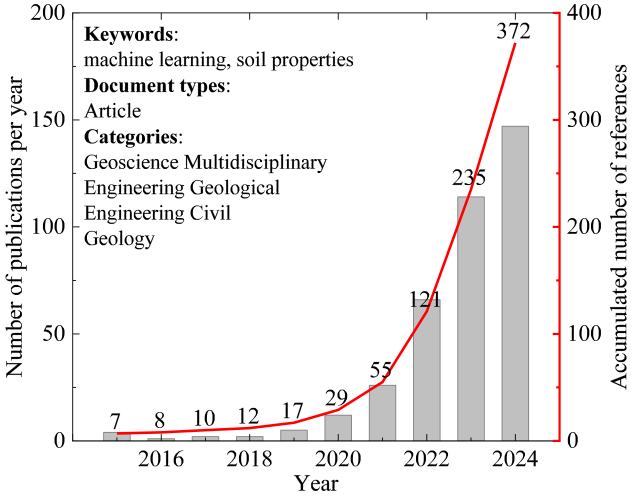


Fig. 1 Evolution of publications associated with ML and soil properties

Alternatively, machine learning (ML) can automatically learn from data, without requiring parametric forms of correlations [15, 16]. It has offered flexible algorithms, e.g., feedforward neural network (FNN) [15, 17–23], random forest (RF) [24–28], support vector regression (SVR) [29–35], evolutionary polynomial regression (EPR) [36–38], genetic programming (GP) [39–42] and extreme learning machine (ELM) [43–47] for data mining in vast fields. As reflected in the Web of Science (Fig. 1), applications of ML in identifying soil properties have grown exponentially since 2018. However, the majority of aforementioned algorithms primarily focus on point estimates and neglect uncertainty quantification, hindering their broader applications, especially in geotechnics with various sources of uncertainty [14, 48–52]. The direct application of estimates without uncertainty quantification may introduce huge risks in engineering practice [2].

For this reason, increasing researchers have been equipping traditionally deterministic approaches with uncertainty quantification to improve their reliability and practicability [49, 53–55]. For example, the Bayesian theorem enables neural networks to quantify uncertainty by approximating interior weight distributions through variational inference (VI) [55] or adjusting network structures through Monte Carlo dropout [56], respectively. Similarly, Gaussian process utilizes Bayesian principles to conduct uncertainty quantification for regression and to provide probabilistic predictions [57]. In contrast, multivariate probability distributions serve as statistical models and can describe the variability and dependence of various soil properties in a probabilistic manner [8, 58, 59]. Besides, quantile regression enables tree-based models to extract the distribution information of observations in leaf nodes for interval estimates [60], rather than merely using average observations inside to generate point estimates. Ensemble learning has also been employed to provide interval estimates for geotechnical parameters through integrations with adaptive Bayesian frameworks [61] or various deterministic approaches [53]. Nevertheless, these enhanced methods involve specialized training strategies and hyper-parameter optimization, which are difficult to apply for users without expertise in data-driven modelling, not to mention subsequent model storage and application [14].

Another long-ignored issue in data-centric geotechnics is that data are often randomly sampled without efficient acquisition principles [62, 63]. Random sampling (RS) cannot evaluate the benefit of each new observation for model development, such that large amounts of costly data are acquired with limited gain in model performance [64]. In contrast, active learning (AL) recognizes that not all data points are equally informative. It uses prior knowledge, such as predictions from a model trained on existing observations, to identify and label the most informative locations in the space of interest [65]. This allows adaptive sampling of expensive experiments under budget constraints and provides practical guidance for efficient data acquisition and data-efficient modelling of soil properties.

Table 1 ML algorithms and geotechnical applications

Mechanism	Algorithm	Main characteristics	Applications
Symbolic regression	MARS	Explicit expression	Wall deflections [66]
	GP		Deformation modulus [67]
	EPR		Creep index [68]
Support vector	SVM	Structural risk minimization	Soil type classification [33]
	SVR		Surface support pressure [69]
	ESVR		Landslide displacements [70]
Tree structure	DT	Random feature and subspace	Soil permeability [71]
	GBDT		Slope displacement [72]
	RF		Grain reconstruction [73]
Neural network	FNN	Flexible structure in data mining	Ground surface movements [74]
	CNN		Particle morphology [75]
	GNN		Compression modulus [76]
	BNN		Undrained shear strength [55]
Distance metric	*k*-NN	Voting scheme	Rock classification [77]
Matrix inversion	ELM	Analytical closed-form solution	Clay compressibility [44]
Random field	GPR	Intrinsic uncertainty	Soil variability [78]
Polynomial	PCE	Explicit polynomial	Tunnelling deformation [79] Consitutive modelling [80] Piles [61, 81]
Statistics	MPD	Statistical models	Rock property [8]

Note: MARS = multivariate adaptive regression splines; GP = genetic programming; EPR = evolutionary polynomial regression; SVM = support vector machine; SVR = support vector regression; ESVR = ensemble based support vector regression; DT = decision tree; GBDT = gradient boosting decision tree; RF = random forest; FNN = feedforward neural network; CNN = convolutional neural network; GNN = graph neural network; BNN = Bayesian neural network; *k*-NN = *k*-nearest neighbour; ELM = extreme learning machine; GPR = Gaussian process regression; PCE = polynomial chaos expansion; MPD = multivariate probability distribution

To this end, this study first reviews the applications of ML algorithms in characterizing soil properties and summarizes their characteristics, with detailed introductions to five probabilistic algorithms, i.e., quantile random forest (QRF), variational inference based evolutionary polynomial regression (VIEPR), ensemble based support vector regression (ESVR), Bayesian neural networks (BNN) and Gaussian process regression (GPR). These algorithms are then compiled into a user-friendly graphical user interface (GUI) platform, namely ErosMLM, to streamline their applications. Without the need to install Python and numerous packages, click is all users need to perform data pre-processing, algorithm selection, model development, evaluation, optimization, storage and application through the platform. Based on equipped probabilistic methods, an uncertainty based AL strategy is further proposed to alleviate data demand and save acquisition costs by effectively navigating data acquisition for data-efficient modelling. Lastly, the creep index C_α and common soil physical properties are fed into the platform to build transformation models, which utilize easily obtained physical indices to predict key mechanical properties as design parameters. This helps examine the feasibility of ML algorithms, the applicability of the platform and the effectiveness of the proposed acquisition strategy.

2 Review of ML Algorithms for Modelling Soil Properties

2.1 Overview of Applied ML Algorithms

Apart from traditional non-linear relations, ML algorithms used in geotechnical practice can be generally categorized by their mechanism, as summarized in Table 1. For instance, symbolic data-driven algorithms, such as EPR, can generate explicit solutions similar to traditionally empirical equations and hence become popular among engineers [68]. SVR is characterized by structural risk minimization [82], which balances model complexity and generalization, and results in strong robustness to noisy data. Based on various trees, RF and its variants have exhibited strong fitting capability [83], while interval estimates are seldom found in their geotechnical applications. Based on flexible structures, neural networks can directly learn from numbers, texts and images, and hence become the most popular in vast fields.

Despite strong fitting capability, the aforementioned approaches cannot quantify underlying uncertainty and interpret predicted deviations. Consequently, uncertainty is neglected in numerous geotechnical applications such as predictions of landslide displacement [70], tunelling-induced settlement [84], wall deflections [66], particle morphology [75], slope stability [85], soil [33] or rock types [77], various soil physical and mechanical properties including hydraulic conductivity [34], undrained shear strength [86], friction angle [87], compression and creep indices [68, 88–92], etc. Under these circumstances, the usage of predictions without uncertainty quantification underuses data and also poses large risks to engineering practice.

For these reasons, more efforts have been made in equipping traditionally deterministic approaches with uncertainty quantification to improve their reliability and practicability. As a result, various enhanced approaches have been proposed in recent years, including BNN [55, 93], ESVR [94], VIEPR and QRF [95], etc. However, compared to deterministic counterparts, these state-of-the-art approaches involve diverse training strategies and model structures, which are challenging for engineers lacking relevant expertise and pose practical obstacles for widespread applications. Additionally, among existing geotechnical applications, many models are simply equipped with default hyper-parameters or those determined by a trial-and-error method, which is inefficient and cannot ensure model performance. Meanwhile, numerous studies generate interval estimates such as a 95% credible interval (CI), but seldom examine the reliability of these intervals [53, 96]. Moreover, models are often built and applied without the quantification of feature importance and analysis of captured correlations, resulting in limited interpretability. Last but not least, sparse geotechnical data inevitably limits the potential of ML due to expensive acquisition costs, and hence data-efficient modelling strategy is worthy of exploration.

Overall, the aforementioned challenges serve as the basis and motivation of this study. Indeed, existing approaches for building probabilistic models to characterize soil properties significantly lag behind the development in the ML domain, not to mention practical adoption by non-experts. To fill these gaps, five probabilistic ML algorithms containing QRF, VIEPR, ESVR, BNN, and GPR were further introduced and tailored into the ErosMLM platform, in which click is all users need for model development, evaluation, optimization, application and illustration of captured correlations. An AL strategy is also proposed to guide data acquisition in a data-efficient manner for reducing data demands and associated costs.

2.2 Quantile Random Forest

Different from RF using mean observations in leaf nodes to generate point estimates [24], QRF weights the distribution of observations in each leaf to calculate the cumulative distribution function (CDF) of predictions and make interval estimates, as illustrated in Fig. 2. Such a difference can also be revealed by comparing Eq. (1) and Eq. (2), which respectively generates point and interval estimate as follows:

$$y = \sum_{i=1}^{n} w_i(x) Y_i \tag{1}$$

where n represents the number of samples used in training, Y_i is the ith observation, y is the point estimate given input x and $w_i(x)$ represents the weight of the ith observation given by averaging the mean observation of the situated leaf in each tree. In contrast, QRF directly replaces Y_i in Eq. (1) with a conditional distribution $P(Y_i \leq y)$ to estimate the CDF of predictions y by

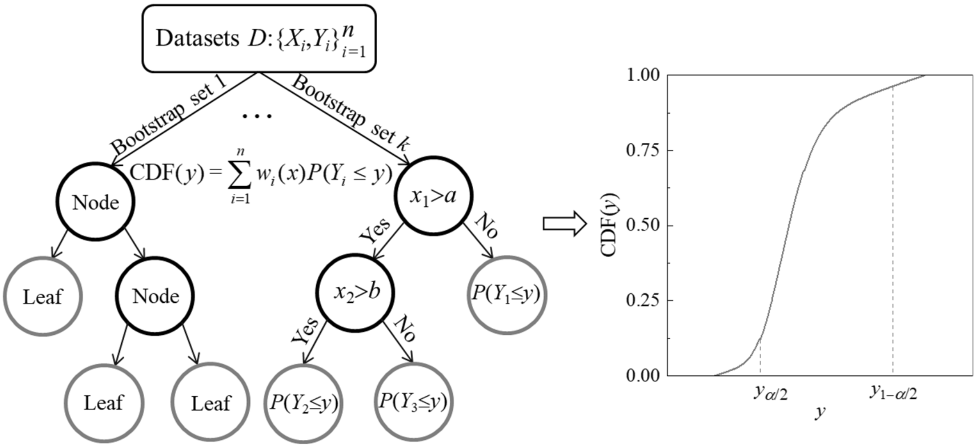


Fig. 2 Schematic of QRF

$$\mathrm{CDF}(y) = \sum_{i=1}^{n} w_i(x) P(Y_i \leq y) \tag{2}$$

where $P(Y_i \leq y)$ represents the probability of $Y_i \leq y$, which equals one if $Y_i \leq y$ or zero otherwise. Consequently, the resulting CDF can be illustrated on the right side of Fig. 2. For QRF, by executing the inverse of Eq. (2), an arbitrarily specified quantile y_p of predictions can be given as follows:

$$y_p = \inf\{\hat{\boldsymbol{y}} : \mathrm{CDF}(y) \geq p\} \tag{3}$$

where CDF(y) represents the cumulative distribution function of y; p is a specified cumulative probability; y_α is the infimum of a subset $\hat{\boldsymbol{y}}$ satisfying $\mathrm{CDF}(\hat{\boldsymbol{y}}) \geq p$. The resulting $y_{0.025}$ and $y_{0.975}$ can constitute the commonly used 95% CI as an interval estimate of y to represent its uncertainty.

2.3 Variational Inference Based Evolutionary Polynomial Regression

To improve the reliability and practicability of polynomials, VIEPR has been proposed to automatically search for a polynomial with interval estimates, as illustrated in Fig. 3. In detail, VIEPR works by incorporating symbolic and numerical regressions in two phases: structure identification and coefficient estimation [14]. For structure identification, evolutionary algorithms such as particle swarm optimization (PSO) are used to search for an exponent matrix $\boldsymbol{E} \in \mathbb{R}^{nt \times m}$ such that m original input variables can be combined with the matrix $\boldsymbol{E}$ to form nt new transformed variables XT as follows:

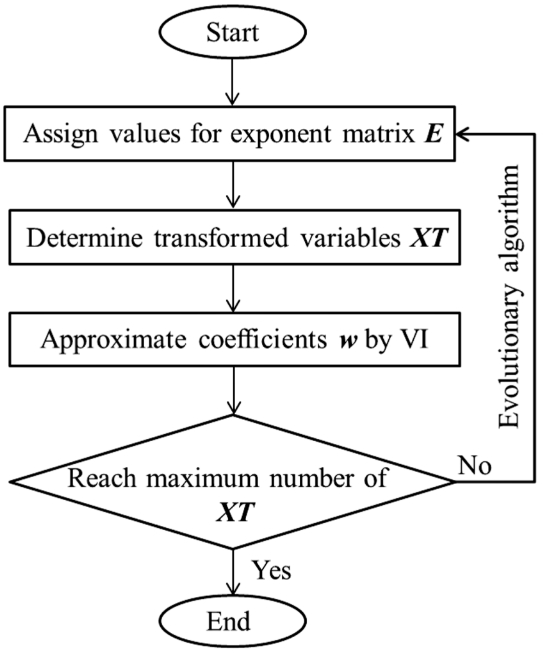

Start
Assign values for exponent matrix E
Determine transformed variables XT
Approximate coefficients w by VI
Reach maximum number of XT
No
Yes
Evolutionary algorithm
End

Fig. 3 Schematic of VIEPR

$$XT_j = x_1^{\boldsymbol{E}(j,1)} \cdot x_2^{\boldsymbol{E}(j,2)} \cdot x_i^{\boldsymbol{E}(j,i)} \cdots x_m^{\boldsymbol{E}(j,m)} \tag{4}$$

where x_i = ith original input variable; XT_j =jth transformed variable; $\boldsymbol{E}(j,m)$=(j,m)-entry of $\boldsymbol{E}$; Next, coefficients are estimated by VI to obtain a general EPR expression as follows:

$$y = \sum_{j=1}^{nt} w_j \cdot XT_j + w_0 \tag{5}$$

where y=predicted output; w_j =an adjustable coefficient of XT_j; w_0=an additional coefficient as bias. Given a prior distribution $P(\boldsymbol{w})$ to coefficients $\boldsymbol{w}$ in Eq. (5), their posterior distribution can be formulated by Bayes' theorem as follows:

$$P(\boldsymbol{w}|D) = \frac{P(D|\boldsymbol{w})P(\boldsymbol{w})}{P(D)} \tag{6}$$

where $P(D\,|\,\boldsymbol{w})$ is the likelihood of observing datasets D given $\boldsymbol{w}$; $P(D)$ is model evidence, which can be regarded as a constant. VIEPR employs a parametric distribution q to approximate the target posterior $P(\boldsymbol{w}|D)$ by learning variational parameters $\boldsymbol{\theta}$ in q to make the parametric probability density $q(\boldsymbol{w}|\boldsymbol{\theta})$ and target posterior PDF $P(\boldsymbol{w}|D)$ as close as possible. This objective is mathematically achieved by minimizing the distance between two distributions, measured by Kullback–Leibler (KL) divergence [97]. Given two distributions p and q, the KL divergence between p and q can be expressed by

$$\mathrm{KL}(q\|p) = E_q[\ln q - \ln p] = \int q(\boldsymbol{w}) \ln \frac{q(\boldsymbol{w})}{p(\boldsymbol{w})}\, d\boldsymbol{w} \tag{7}$$

where $q(\boldsymbol{w})$ and $p(\boldsymbol{w})$ denote the probability densities given certain $\boldsymbol{w}$.

Therefore, the variational parameters $\boldsymbol{\theta}$ are found by minimizing the KL divergence between parameterized $q(\boldsymbol{w}|\boldsymbol{\theta})$ and target posterior $P(\boldsymbol{w}|D)$ as follows:

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \mathrm{KL}\left[q\left(\boldsymbol{w}|\,\boldsymbol{\theta}\right) \,\|\, P\left(\boldsymbol{w}|\,D\right)\right] \\ &= \arg\min_{\boldsymbol{\theta}} \int q\left(\boldsymbol{w}|\,\boldsymbol{\theta}\right) \ln \frac{q\left(\boldsymbol{w}|\,\boldsymbol{\theta}\right)}{P\left(\boldsymbol{w}\right) P\left(D|\,\boldsymbol{w}\right)} d\boldsymbol{w} \\ &= \arg\min_{\boldsymbol{\theta}} \mathrm{KL}\left[q\left(\boldsymbol{w}|\,\boldsymbol{\theta}\right) \,\|\, P\left(\boldsymbol{w}\right)\right] - E_{q(\boldsymbol{w}|\boldsymbol{\theta})}\left[\ln P\left(D|\,\boldsymbol{w}\right)\right]\end{aligned} \tag{8}$$

Taking Monte Carlo sampling to evaluate the expectation term in Eq. (8), the KL divergence is further expressed by the following loss function [55, 98]:

$$L(D,\boldsymbol{\theta}) \approx \frac{1}{s}\sum_{i=1}^{s}\left(\ln q(\boldsymbol{w}^{(i)}|\boldsymbol{\theta}) - \ln P(\boldsymbol{w}^{(i)}) - \ln P(D|\boldsymbol{w}^{(i)})\right) \tag{9}$$

where $\boldsymbol{w}^{(i)}$ denotes the ith sample drawn from the variational posterior $q(\boldsymbol{w}^{(i)}|\boldsymbol{\theta})$ and s represents the number of samples. For brevity, the variational posterior of $\boldsymbol{\theta}$ is supposed to comply with normal distributions with two parameters: mean and standard deviation, denoted by u and ρ, which can be optimized by gradient descent [55]. It is noteworthy that VI can work well in a few polynomials but may suffer from expensive computational costs and convergence issues when applied to numerous weights in neural networks. Nonetheless, data pre-processing, such as the parameter transformation through the Johnson distribution [99], can alleviate these issues.

2.4 Ensemble Based Support Vector Regression

Characterized by structural risk minimization and sparseness solution [29], ESVR is proposed to explore uncertainty in different sampled data, as shown in Fig. 4. In ESVR, datasets for developing each sub-SVR are obtained by bootstrap aggregating [100], which generates a bootstrap set with almost two-thirds of complete datasets by sampling with replacement [5]. For datasets that cannot be fitted well in a low-dimensional space, SVR can map datasets to a higher-dimensional and even infinite space to perform regression using kernel functions [30]. The radial basis function (RBF) is the most commonly used kernel, which can be specified by

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right) \tag{10}$$

where$\|$ $\|$ denotes Euclidean distance, γ is the kernel coefficient, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ denote arbitrary inputs.

Figure 4 illustrates that the training of each sub-SVR is to find a function that allows certain deviations from observations and meanwhile is as smooth as possible, which can be mathematically expressed by

$$\begin{aligned} &\min_{W,b,\xi,\xi^*} \sum_{i=1}^{n} C(\xi_i+\xi_i^*) + \frac{1}{2}\|W\|^2 \\ \text{s.t.}\quad & W^T\phi(X) + b - \varepsilon - \xi_i \le Y_i \le W^T\phi(X) + b + \varepsilon + \xi_i^* \end{aligned} \tag{11}$$

where W, $\phi(X)$ and b are coefficients, basis functions and intercepts of the fitted function. ξ represents slack factors outside the allowable deviation ε without penalty [29]. In this context, structural risk minimization is expressed in Eq. (11), where the former and latter denote empirical risk and the regularization term, respectively, with a regularization parameter C to leverage these two terms. As a result, interval estimates can be obtained using sufficient bootstrap sets in ESVR.

2.5 Bayesian Neural Network

Given the strong capability of neural networks and the necessity of uncertainty quantification, Gal and Ghahramani [56] proposed a novel theoretical framework that implements dropout based training in neural networks to approximate BNN for interval estimates, as shown in Fig. 5. Specifically, Monte Carlo dropout alleviates computational costs and inactivates each neuron in neural networks with a probability of p to approximate the exact values of the traditional Bayesian methods [55]. This can be expressed mathematically by

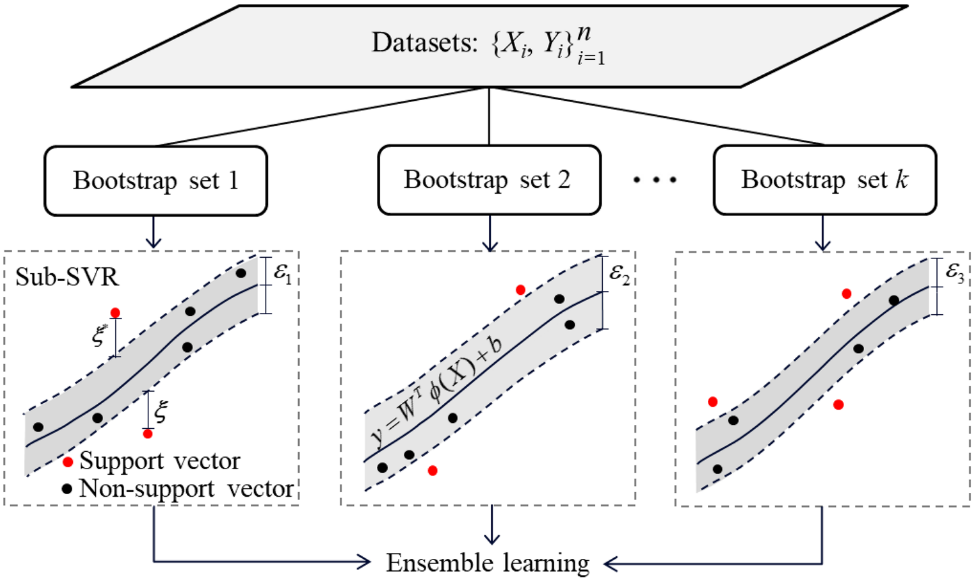


Fig. 4 Schematic of ESVR

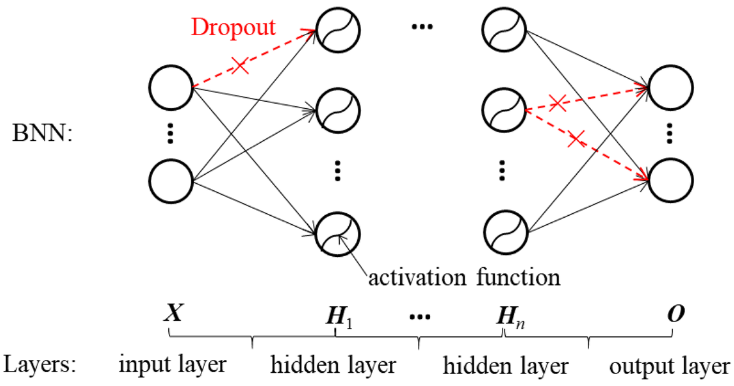


Fig. 5 Schematic of BNN

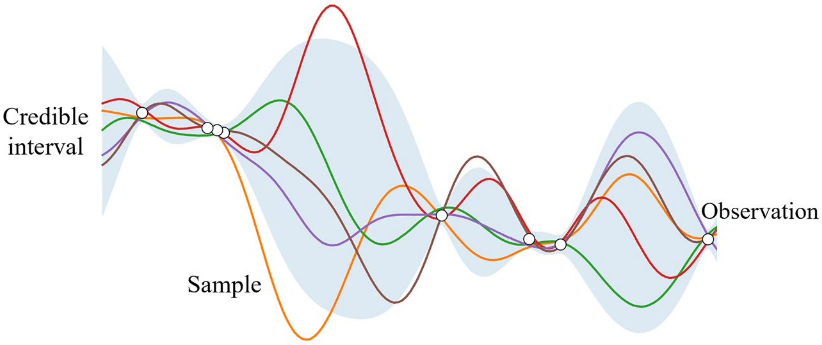


Fig. 6 Schematic of GPR

$$y^{l+1} = f\left(w^{l+1} y^l r^l + b^{l+1}\right), r^l \sim \text{Bernoulli}(p) \tag{12}$$

where y^l and y^{l+1} represent the outputs at the *l*th and (*l*+1)th layers, respectively, w^{l+1} and b^{l+1} are the weights and biases at the (*l*+1)th layer, respectively, *r* equals either zero or one, following a Bernoulli distribution associated with dropout probability *p* [101]. In this context, weights and biases are fixed, while the structure of neural networks varies during both the training and testing phases to capture uncertainty. By performing sufficiently stochastic inferences on data, the resulting mean and variance can be used to generate interval estimates and represent predicted uncertainty.

2.6 Gaussian Process Regression

Based on Bayes' theorem, GPR operates by estimating the conditional distribution of quantities of interest $\boldsymbol{q}$ based on existing observations $\boldsymbol{o}$ [57], as shown in Fig. 6. GPR supposes that both $\boldsymbol{q}$ and $\boldsymbol{o}$ are Gaussian random vectors and have a joint multivariate Gaussian distribution with a zero mean vector and a kernel-based covariance matrix as follows [102]:

$$\begin{bmatrix} \boldsymbol{o} \\ \boldsymbol{q} \end{bmatrix} \sim N\left(0, \begin{bmatrix} \boldsymbol{K_{oo}} + \sigma_n^2 \boldsymbol{I} & \boldsymbol{K_{oq}} \\ \boldsymbol{K_{oq}} & \boldsymbol{K_{qq}} \end{bmatrix}\right) \tag{13}$$

where $\boldsymbol{K_{oq}} = \boldsymbol{K}_{\boldsymbol{qo}}^T$ represents the covariance matrix between $\boldsymbol{o}$ and $\boldsymbol{q}$ by feeding inputs into a kernel function $k(\cdot,\cdot)$, which is commonly specified by [103]:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(\sum_i \frac{(\boldsymbol{x}_i - \boldsymbol{x}_i')^2}{2l_i^2}\right) \tag{14}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_i'$ are the *i*th dimension of arbitrary inputs, length scale l_i represents a distance within which the quantity of interest is significantly correlated and signal variance σ_f^2 is the variance of random processes, controlling the amplitude of underlying functions [78]. Given observations *o*, the conditional distribution $p(\boldsymbol{q}|\boldsymbol{o})$ is yet a multivariate Gaussian distribution, and can be formulated by the following mean vector $\boldsymbol{u}(\boldsymbol{q})$ and covariance matrix $\boldsymbol{K}(\boldsymbol{q})$ [103]:

$$\boldsymbol{u}(\boldsymbol{q}) = \boldsymbol{K_{qo}} \boldsymbol{K}_{\boldsymbol{oo}}^{-1} \boldsymbol{o} \tag{15}$$

$$\boldsymbol{K}(\boldsymbol{q}) = \boldsymbol{K_{qq}} - \boldsymbol{K_{qo}} \boldsymbol{K}_{\boldsymbol{oo}}^{-1} \boldsymbol{K_{oq}} \tag{16}$$

As the above predictions involve hyper-parameters in $k(\cdot,\cdot)$, they are optimized using maximum likelihood estimation that minimizes the negative logarithmic marginal likelihood by

$$\frac{1}{2}\boldsymbol{m}^T\boldsymbol{K}_{\boldsymbol{oo}}^{-1}\boldsymbol{o}+\frac{1}{2}\log|\boldsymbol{K}_{\boldsymbol{oo}}|+\frac{n}{2}\log 2\pi \quad (17)$$

where $|\cdot|$ denotes a determinant and n denotes the number of measurements.

3 Summary

The introduction of the above five probabilistic ML algorithms has revealed their respective characteristics as summarized in Table 2. The development, evaluation, optimization, and application of these ML-based models require extensive expertise, which remains challenging for practitioners and hinders the applications of ML in vast fields. Therefore, these approaches deserve to be packaged in a user-friendly manner such that users have access to them without any threshold.

4 Development of ML-Based Modelling Platform

4.1 Graphical User Interface Platform

Herein, a GUI platform is developed to offer engineers a user-friendly tool to build, optimize, evaluate and apply probabilistic ML-based models. To ensure the feasibility of the platform across different computers and systems, it is programmed in Python and compiled with all necessary packages as an executable file. Hence, it can be easily delivered between different users and used immediately and independently without any threshold.

Figure 7 presents the main interface of the GUI platform, where six main modules are designed in the left-side toolbar: (1) "Development" is compiled as a submodule to train ML-based models and optimize their configurations; (2) "Application" is compiled as another submodule to directly apply ML-based models to make predictions on new data and parametric analysis; (3) "About" is compiled to show copyright information and announcements; (4) "Help" is compiled to provide users with an operation manual of ErosMLM; (5) "Reference" is compiled to introduces some relevant research works; (6) "Exit" is compiled to close the platform.

More specifically, the workflow of developing and applying ML-based models is presented in Fig. 7a and primarily involves the following six steps: (1) selecting a specific ML algorithm; (2) loading a database; (3) preprocessing datasets; (4) automatically searching for an optimal configuration and saving the model; (5) loading the model for predictions on new data or parametric analysis as shown in Figs. 7b–7d; (6) saving the results. Herein, key steps from (3) to (5) are detailed later to facilitate the understanding of the platform.

Table 2 Representative ML algorithms for characterizing soil properties

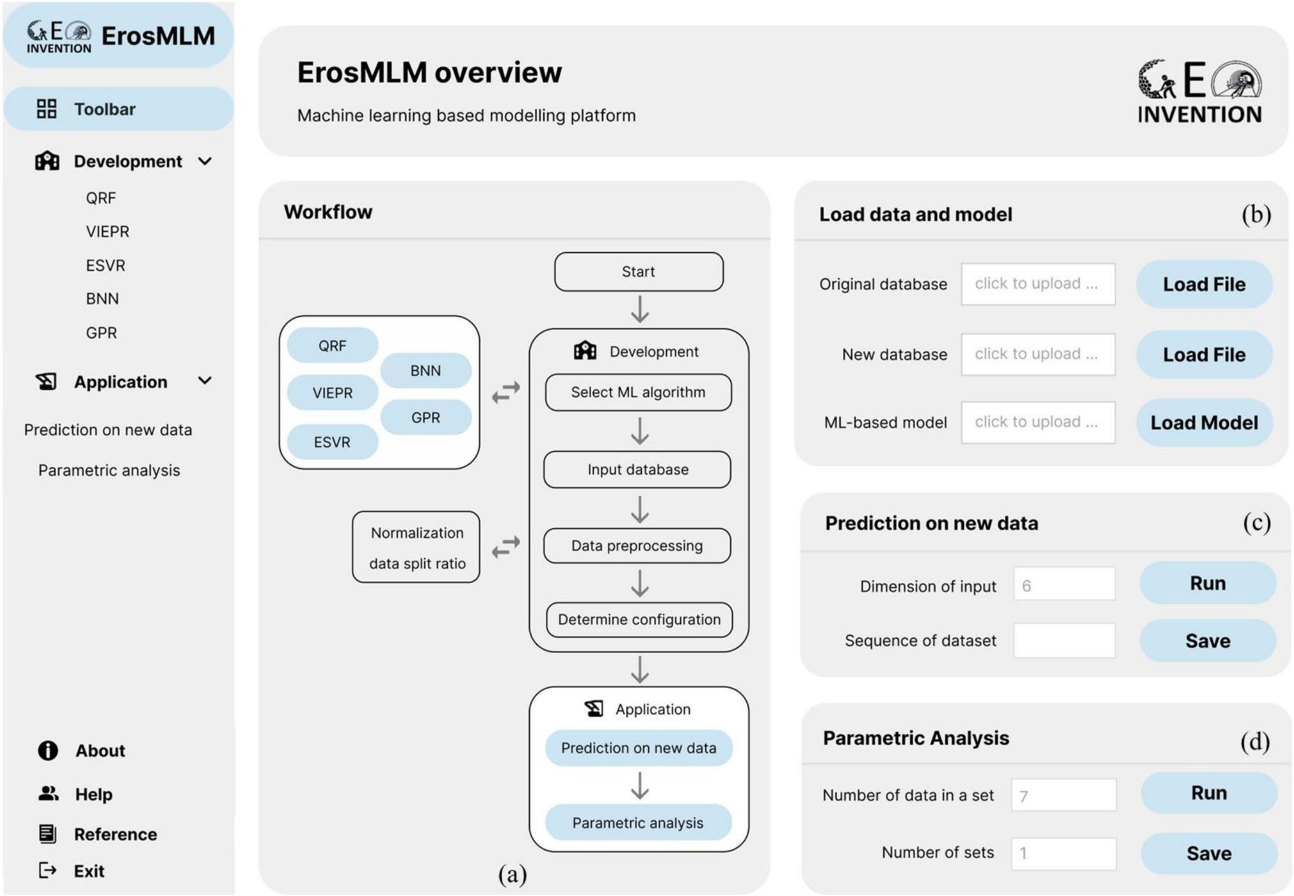
Algorithms	Advantages	Limitations
QRF	Strong fitting ability; Fast training	Poor extrapolation ability
VIEPR	Explicit expression	Limited non-linear mapping ability
ESVR	Structural risk minimization	Poor readability and interpretability
BNN	Strong non-linear mapping ability	Numerous hyper-parameters; Huge computational costs
GPR	Explicit expression; Fast training; Efficient hyper-parameters optimization	Limited extrapolation ability

4.2 Data Preprocessing

After specifying one algorithm in Fig. 7a, the corresponding operation panel will pop up. Once the database is loaded through the "Load file" button in Fig. 7b, the next step would be data preprocessing, which includes the normalization, assignment of input and output dimensions and random split of training and testing datasets. The min-max normalization is used such that the entire datasets are rescaled within the range [−1, 1] to eliminate the scale effect of variables. On the other hand, the dimension of inputs can be automatically calculated once the dimension of output variables is imported by users. Lastly, the proportion of datasets in the training set is determined by a split ratio typically chosen as 80%, and the same training set can be used by different algorithms as long as random seeds are fixed. It is noteworthy that additional processing, such as batch size and cross-validation, is also included in BNN to enhance its robustness.

4.3 Automatic Optimization of Configuration

After data preprocessing, a pivotal task in model development is to determine hyper-parameters associated with models. In the "Development" module, the hyper-parameters for each algorithm can be optimized through grid or iterative search because they can systematically search the parameter space. Specifically, optimized ranges are initially pre-assigned and can be customized by users to search for an optimal configuration. This process helps elucidate the relationship between model performance and its hyper-parameters. Compared to absolute errors with ever-changing scales and possibly unstable relative errors, the coefficient of determination (R^2) is recommended to evaluate model performance in the process of hyper-parameter



Dimension of input 6

Sequence of dataset

Number of data in a set 7

Number of sets 1

Fig. 7 Schematic of ErosMLM: (**a**) workflow; (**b**) “Load data and model” module; (**c**) “Prediction on new data” module; (**d**) “Parametric analysis” module

optimization for ensuring scale-independence and generality (see Eq. (18)). This is because it is quite understandable and also directly reflects the agreement between predicted and measured quantity of interest. Specifically, a larger R^2 denotes better precision, with a value of 1 representing perfect consistency between predictions and measurements.

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i^m - \bar{y}_i^p\right)^2}{\sum_{i=1}^{n}\left(y_i^m - \bar{y}_i^m\right)^2} \tag{18}$$

where n is the total number of data points,y_i^p, y_i^m and $\bar{y}_i^m$ are the predicted, measured and average measured quantity of interest, respectively. Once optimized ranges are assigned, a “Training” button can be clicked to directly start the training and optimization processes, and thereafter an optimal configuration and the resulting R^2 in the training and testing sets will pop up in an “Output” panel. Meanwhile, the relationships between hyper-parameters and model performance will be automatically illustrated via figures, and a “Save” button is designed to store the optimal model, predictions and figures. The above complete training process for each algorithm will be demonstrated later through a case study.

4.4 Application of Developed ML-Based Models

Once developed models are saved, users can turn to the “Application” module to manifest two underlying submodules: “Predictions on new data” and “Parametric analysis”. By clicking these submodules, their interfaces will pop up the same as Figs. 7c and 7d.

4.4.1 Prediction on New Data

No matter how big the database is, it is probable to observe data that exceed the range of variables in the existing training set. It is challenging for novices to apply previously developed models to other databases without any expertise because direct applications require data processing and relevant packages. To this end, the “Load data and model” and “Predictions on new data” modules are developed as shown in Figs. 7b and 7c, in which click is all users need to make predictions on unseen data. Specifically, the original database is loaded through the “Load file” and “Load model” buttons to ensure the same preprocessing for the new database, because previous models are built upon processed data. Once the database and the model are loaded, users can specify the dimension of inputs and the sequence of datasets of interest in the loaded database in Fig. 7c to make predictions, although the default is to predict all loaded datasets. As a result, the predictions for the new database can be directly generated and saved by clicking the "Run" and "Save" buttons.

4.4.2 Parametric Analysis

For ML-based models, captured correlations reflect what has been learnt from data. Apart from predictions on new data, a "Parametric analysis" module is also designed to reveal captured correlations and facilitate the understanding of developed models. In this module, a CDF based method is employed to investigate and expose the relationships between the input and output variables [55]. For parametric analysis, a series of quantiles of a studied input variable can be directly generated by the MATLAB command "prctile" according to the statistics in the original database. Meanwhile, the remaining input variables are fixed and represented by their mean or other specified values to explore and reflect underlying correlations. To avoid extreme ranges, quantiles spanning from 20% to 80% with an interval of 10% are suggested to form a set of data as an example for parametric analysis of a studied variable. The above example corresponds to 7 and 1 (the number of quantiles and studied variables) in Fig. 7d, in which their number can be flexibly specified by users.

4.5 Active Learning Guided Data-Efficient Modelling

Apart from the tool for data-driven modelling, another long-ignored issue is that data are often randomly sampled and directly used in geotechnical practice without efficient acquisition principles. Different from unprincipled acquisition, AL recognizes that different data have varying contributions to model development. It can utilize prior knowledge, e.g., uncertainty quantification, probability of improvement and expected improvement [104, 105], to find the most pivotal location in the space of interest and to iteratively add data for model development [106, 107]. Based on probabilistic algorithms compiled in the platform, an uncertainty based AL strategy is further proposed to guide data acquisition for reducing data demand and saving acquisition costs.

Figure 8 illustrates the proposed AL strategy, which utilizes the model built on observed data (circles) and uncertainty quantification to iteratively pinpoint and label the most uncertain sample among unobserved data (crosses) for data-efficient modelling. This is because an unobserved dataset with the largest uncertainty represents the sample that the model has the least knowledge of and therefore deserves to be prioritized and actively acquired for subsequent training (red circles). More specifically, the width of the predicted CI is employed as the acquisition function to represent uncertainty and guide data acquisition. Such a strategy enables autonomous sampling of costly experiments under constrained experimental costs and provides engineers with constructive suggestions for data acquisition. To examine its effectiveness, known data can be treated as unobserved data intentionally and are gradually acquired through AL or RS, such that the resulting model performance can be compared. In a real-world scenario, the unexplored locations of interest in the studied space represent unobserved data, and their acquisition can be guided in a data-efficient manner to save data demand and associated costs.

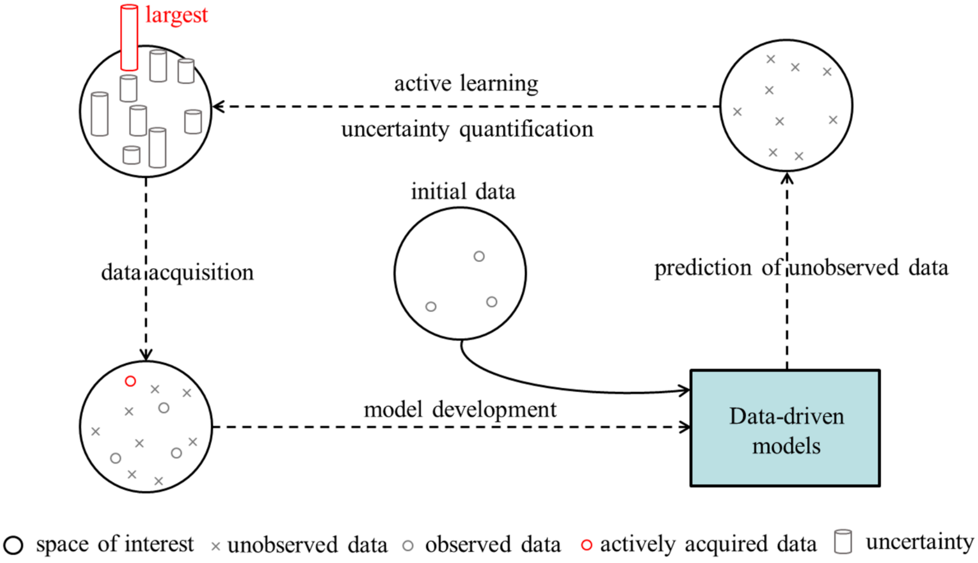


Fig. 8 Schematic of uncertainty based active learning

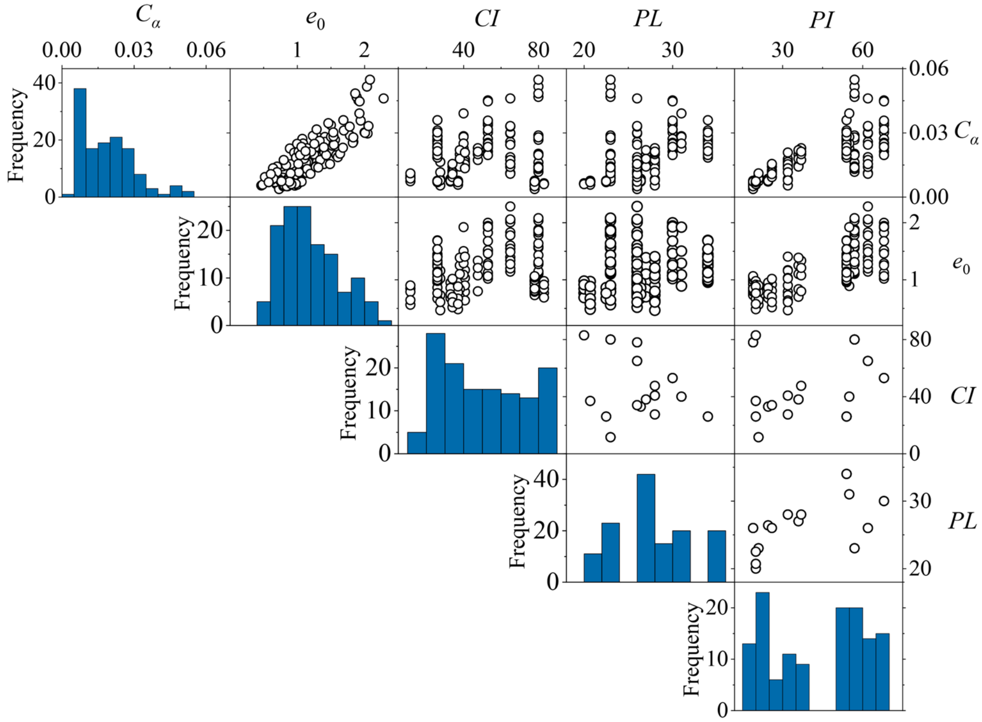


Fig. 9 Histograms and scatterplots of the database

5 Case Study

5.1 Data Source

The database used in this study is extracted from previous studies [13, 108–111] and includes 131 datasets. Each dataset comprises the C_α and four easily measured indices, i.e., initial void ratio e_0, clay content CI, plastic limit PL and plastic index PI. Fig. 9 presents their pairwise scatterplots in the upper triangular, in which dense and sparse clusters can be easily observed. In this context, uncertainty quantification becomes pivotal because accurate point estimates are elusive due to insufficient knowledge of sparse clusters. The distributions of variables are revealed by the histograms at the diagonal, with detailed statistics summarized in Table 3.

where n and n_i are the number of target quantities and those falling inside a certain predicted interval, typically taken as the common 95% CI. In statistics, a reliable 95% CI enables the RI value to be around 95% if sufficient datasets are available [99, 103, 115, 116].

Table 3 Statistics of variables

Variable	Min.	Max.	Mean	STD
C_α	0.004	0.055	0.019	0.011
e_0	0.466	2.284	1.186	0.418
CI	11.50	83.00	49.69	21.67
PL	20.00	34.00	27.14	4.025
PI	19.00	68.00	43.24	18.11

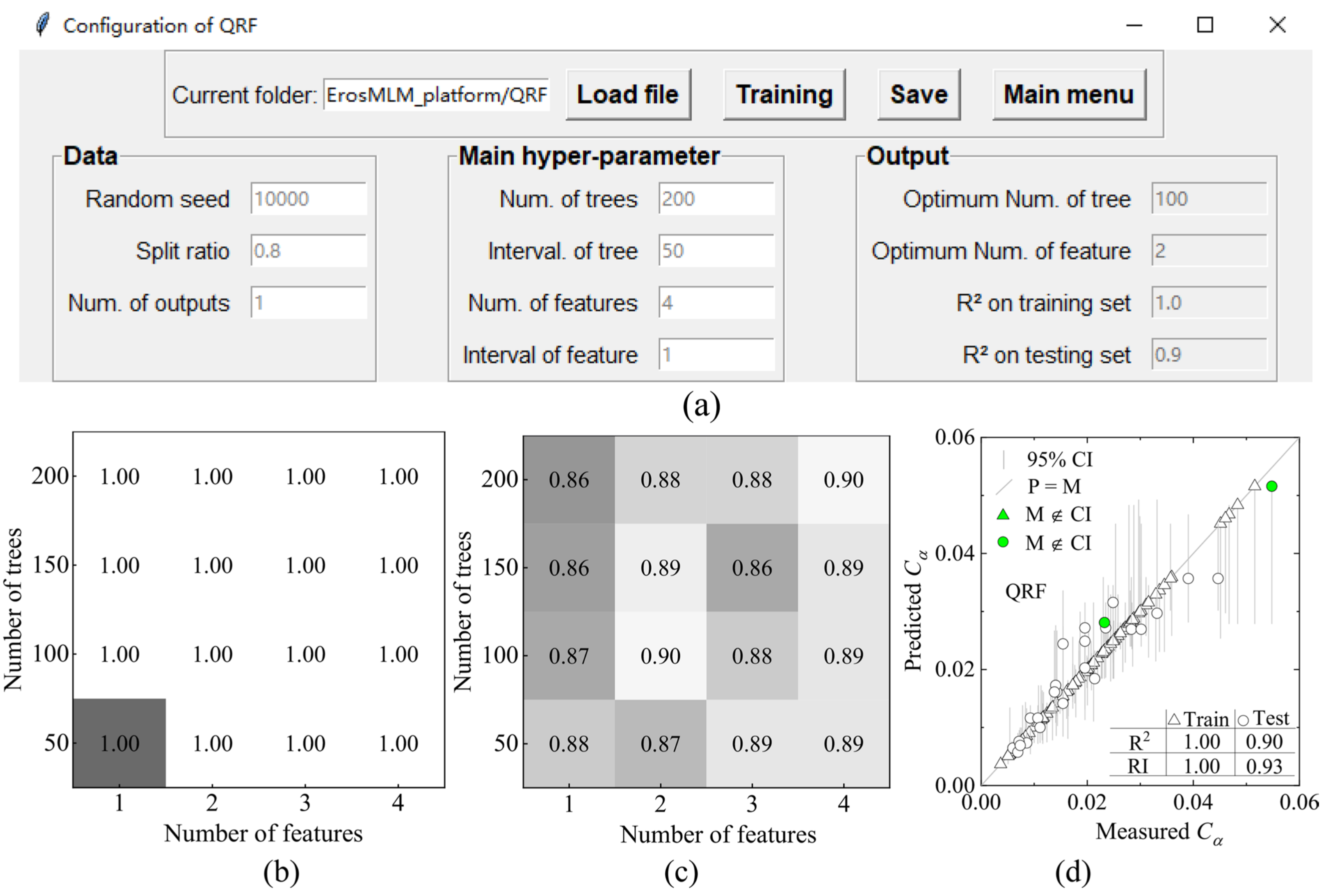
Note: STD = standard deviation

5.2 Development of ML-Based Models Using ErosMLM

5.2.1 Quantile Random Forest

The training of models was conducted by a baseline algorithm under a given configuration. Figure 10a presents the initial setting of QRF. Similar to previous works [95, 117], the number of trees (*ntree*) and features for split (*mtry*) varied from 50 to 200 and from 1 to 4 to search for an optimal configuration, since they significantly affect the size and shape of the forest. Since the features used to split each node are randomly sampled, all features can be used in building tree models, even though the value of *mtry* is 1.

Based on diverse *ntree* and *mtry* values, the performance of QRF on the training and testing sets was presented in Figs. 10b and 10c, in which all R^2 values were greater than 0.85 with minor fluctuations under different configurations, and demonstrated the strong fitting capability of tree structures. The optimal *ntree* and *mtry* values were found at 100 and 2 in Fig. 10a, in which the resulting R^2 values for the training and testing sets were 1.00 and 0.90, respectively, and showed excellent agreement. Figure 10d further compares the predicted and measured C_α values and finds that



Configuration of QRF

Current folder: ErosMLM_platform/QRF

Data

- Random seed: 10000
- Split ratio: 0.8
- Num. of outputs: 1

Main hyper-parameter

- Num. of trees: 200
- Interval. of tree: 50
- Num. of features: 4
- Interval of feature: 1

Output

- Optimum Num. of tree: 100
- Optimum Num. of feature: 2
- R² on training set: 1.0
- R² on testing set: 0.9

(a)

Number of trees	1	2	3	4
200	1.00	1.00	1.00	1.00
150	1.00	1.00	1.00	1.00
100	1.00	1.00	1.00	1.00
50	1.00	1.00	1.00	1.00

Number of features

(b)

Number of trees	1	2	3	4
200	0.86	0.88	0.88	0.90
150	0.86	0.89	0.86	0.89
100	0.87	0.90	0.88	0.89
50	0.88	0.87	0.89	0.89

Number of features

(c)

	△Train	○Test
R^2	1.00	0.90
RI	1.00	0.93

(d)

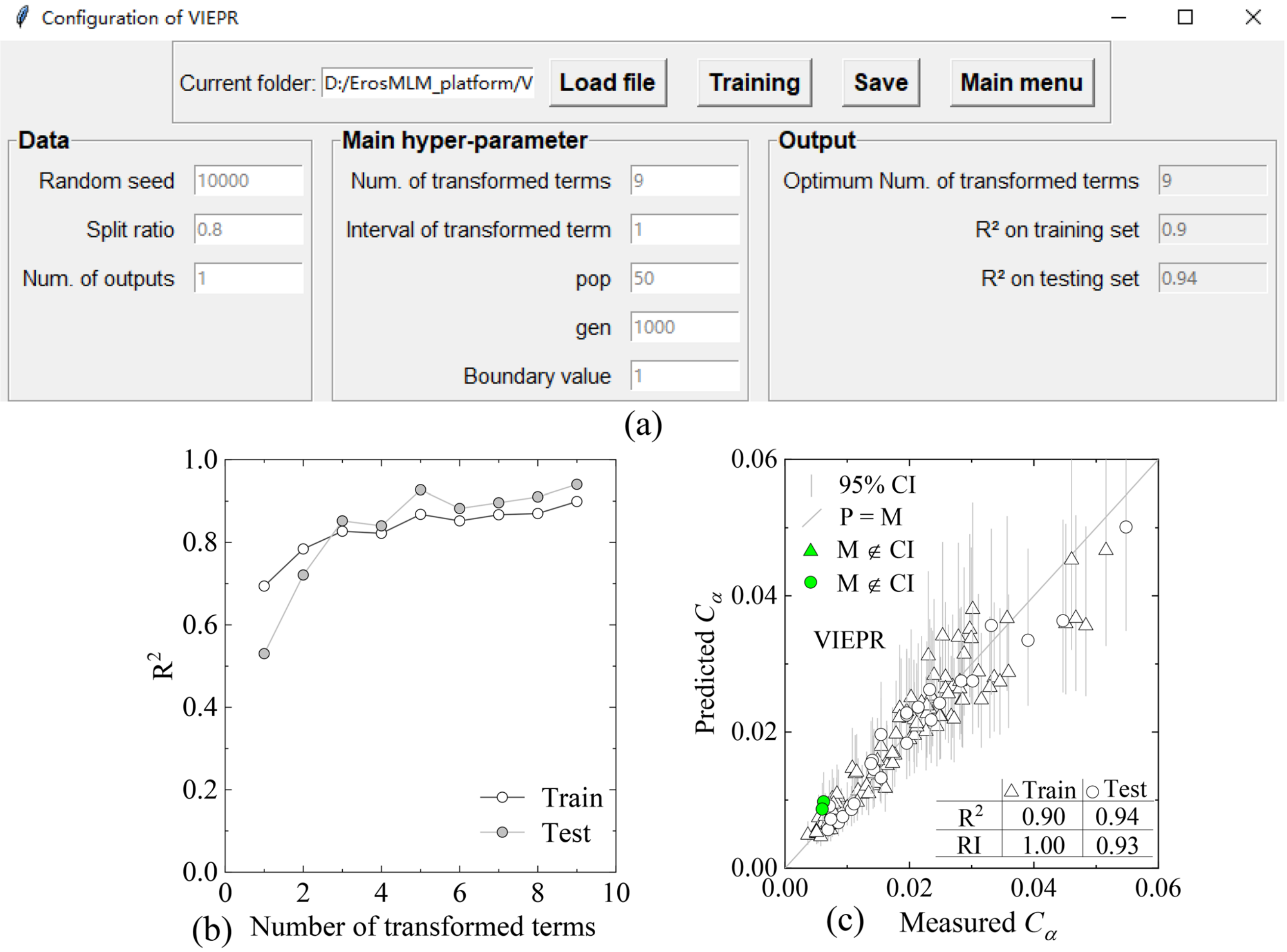
Fig. 10 Performance of QRF under various configurations: (**a**) initial settings of QRF; (**b**) on the training set; (**c**) on the testing set; (**d**) predictions based on the optimal configuration

only two measured values fell outside the predicted 95% CI. For better understanding, the measured C_α values outside 95% CI in the training and testing sets were plotted as green triangles and circles, of which the ratio can reveal model reliability because excessive measurements outside a 95% CI imply underestimated uncertainty. Meanwhile, the RI values on both training and testing sets reached around 0.95. These results demonstrate the excellent accuracy and reliability of QRF and the practicability of the platform.

5.2.2 Variational Inference Based Evolutionary Polynomial Regression

With a popular polynomial expression, the initial setting of VIEPR was illustrated in Fig. 11a. Specifically, the number of transformed polynomial terms varied within ten to explore various configurations, which can also be flexibly specified by users. The boundary value of the exponent matrix was set to one to prevent impractical formulations and overflow errors, similar to numerous empirical equations [3, 12]. The whole exponent matrix was searched and optimized by PSO with the baseline setting of population size (pop) and number of generations (gen) [95], as shown in Fig. 11a. Once the training process is done, the optimized exponent matrix can be directly stored as a visible file through the "Save" button to facilitate the understanding of VIEPR.

Fig. 11b presents the R^2 values generated by VIEPR models under various configurations. As the number of transformed terms increased, the model performance generally improved in both the training and testing sets, despite slight fluctuations in the local range. The optimal number of transformed terms was 9, generating the largest R^2 value as shown in 11a. Similar to the performance of QRF, Fig. 11c shows that only two measurements fell outside the predicted 95% CI, with the same RI values as QRF on both training and testing sets. Different from QRF with flexible trees and leaves to build the model, the VIEPR model was constrained by limited transformed polynomial terms, and observed a slightly smaller R^2 value and significantly wider CI. On the other hand, the improvement of model performance



Configuration of VIEPR

Current folder: D:/ErosMLM_platform/V

Data

Random seed: 10000

Split ratio: 0.8

Num. of outputs: 1

Main hyper-parameter

Num. of transformed terms: 9

Interval of transformed term: 1

pop: 50

gen: 1000

Boundary value: 1

Output

Optimum Num. of transformed terms: 9

R² on training set: 0.9

R² on testing set: 0.94

	△Train	○Test
R²	0.90	0.94
RI	1.00	0.93

Fig. 11 Performance of VIEPR under various configurations: (**a**) initial settings of VIEPR; (**b**) on the training and testing sets; (**c**) predictions based on the optimal configuration

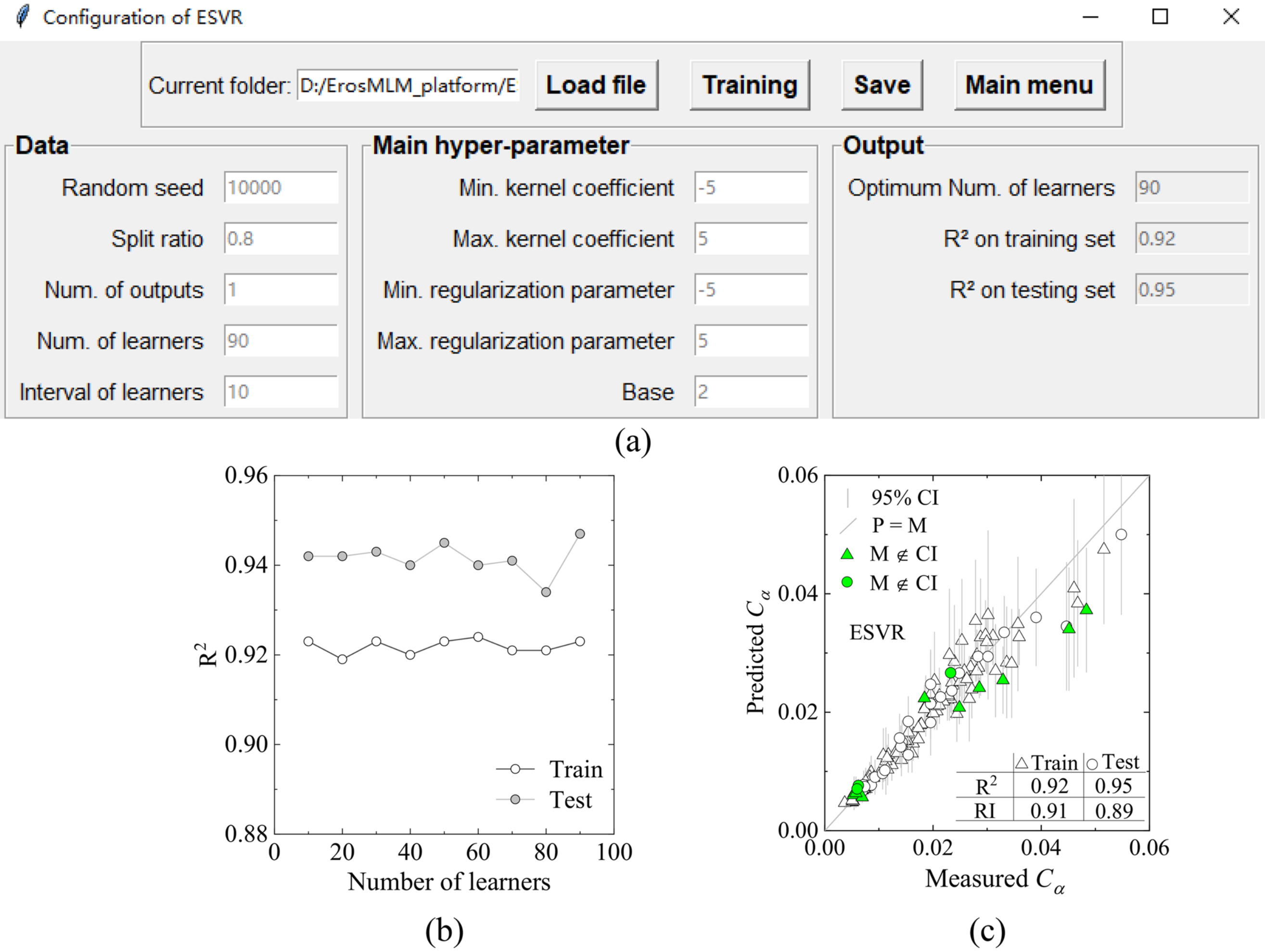
gradually decreases as the number of transformed terms in VIEPR increases, indicating that the subsequently added terms become trivial in practice. This finding suggests that the most informative terms are prioritized to build polynomials when the upper bound of transformed terms can be flexibly specified by users.

5.2.3 Ensemble Based Support Vector Regression

Similarly, the initial setting of ESVR was presented in Fig. 12a, in which common hyper-parameters, including kernel coefficient and regularization parameter, were optimized within a wide range $[2^{-5}, 2^5]$ to ensure an optimal configuration for each learner in ESVR. The Base in Fig. 12 represents the base of hyper-parameters during grid search and implies that they use this base as the multiplier to vary, which will not be further explained for brevity. A particular hyper-parameter in ESVR is the number of learners, which varies within 100 at a fixed interval of 10 to allow sufficient learners and explore various configurations.

5.2.4 Bayesian Neural Network

To explore uncertainty in network structures, BNN has incorporated Monte Carlo dropout for Bayesian approximation, with an initial setting in Fig. 13a. Specifically, the number of hidden layers and the number of neurons in each layer vary within wide ranges to explore optimal configurations, since they significantly affect the performance of neural networks. The remaining configurations, such as the number of epochs and folds, can be arbitrarily specified by users and are assigned with commonly used settings [14, 55]. Specifically, the number of initial assignments and stochastic calculations represents the times of weight initializations and Monte Carlo dropout in neural networks. The dropout rate is expected to realize reliable uncertainty quantification, e.g., enabling the 95% CI to cover almost 95% exact values. Apart from dropout, *k*-fold cross-validation has also been used to prevent overfitting. Meanwhile, the activation function and optimizer can also be replaced by other options in



Configuration of ESVR

Current folder: D:/ErosMLM_platform/E

Load file | Training | Save | Main menu

Data

Random seed: 10000
Split ratio: 0.8
Num. of outputs: 1
Num. of learners: 90
Interval of learners: 10

Main hyper-parameter

Min. kernel coefficient: -5
Max. kernel coefficient: 5
Min. regularization parameter: -5
Max. regularization parameter: 5
Base: 2

Output

Optimum Num. of learners: 90
R² on training set: 0.92
R² on testing set: 0.95

(a)

	△Train	○Test
R^2	0.92	0.95
RI	0.91	0.89

Fig. 12 Performance of ESVR under various configurations: (**a**) initial settings of ESVR; (**b**) on the training and testing sets; (**c**) predictions based on the optimal configuration

the platform. As the Monte Carlo dropout adjusts the neural structure to generate interval estimates, negative log-likelihood (NLL) integrated with k-fold cross-validation is taken as the loss function to maximize the likelihood of observing measurements, expressed as follows:

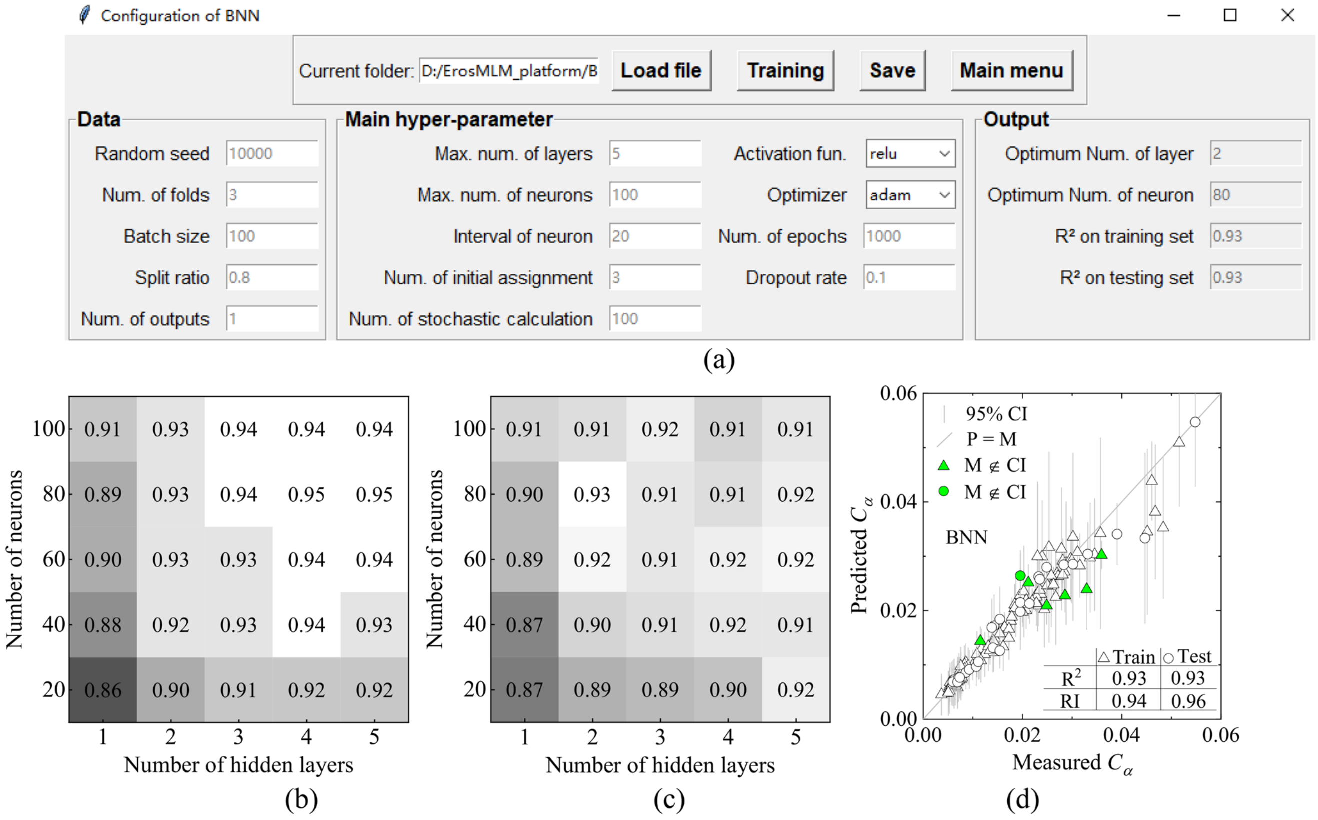
$$\mathrm{NLL} = -\frac{1}{kn_v}\sum_{i=1}^{n}\log[p(y_i^m|\boldsymbol{w},\boldsymbol{x})] \tag{20}$$

where n and n_v are the total number of training samples and the number of samples in each validation set. $p(y_i^m|\boldsymbol{w},\boldsymbol{x})$ represents the probability density of observing the ith measurement y_i^m given weights $\boldsymbol{w}$ and inputs $\boldsymbol{x}$.

Based on different configurations, Figs. 13b and 13c present the model performance of BNN on the training and testing sets, respectively. When it comes to more layers and neurons, the R^2 values significantly increased on the training set but decreased on the testing set. Such a result complies with common recognition that the fitting capability of neural networks improves with more complex structures, and meanwhile tends to overfit for unobserved data. The optimal numbers of layers and neurons were found at 2 and 80, respectively, observing the largest R^2 value on the testing set. Figure 13d shows that the resulting R^2 and RI values were approximately identical on the training and testing sets, reaching around 0.93 and 0.95, respectively, demonstrating excellent accuracy, reliability and generalization ability.

5.2.5 Gaussian Process Regression

Based on observations to generate posterior distributions, the initial setting of GPR is illustrated in Fig. 14a, in which sufficient ranges were assigned to search for the optimal configuration of function amplitude and length scale. Apart from the optimum amplitude, the resulting optimum length scale of each variable is presented in the panel, according to its order in the database. As hyper-parameters were iteratively optimized, the evolution of model performance was



Configuration of BNN

Current folder: D:/ErosMLM_platform/B

Data

Random seed	10000
Num. of folds	3
Batch size	100
Split ratio	0.8
Num. of outputs	1

Main hyper-parameter

Max. num. of layers	5
Max. num. of neurons	100
Interval of neuron	20
Num. of initial assignment	3
Num. of stochastic calculation	100
Activation fun.	relu
Optimizer	adam
Num. of epochs	1000
Dropout rate	0.1

Output

Optimum Num. of layer	2
Optimum Num. of neuron	80
R² on training set	0.93
R² on testing set	0.93

(a)

Number of neurons	1	2	3	4	5
100	0.91	0.93	0.94	0.94	0.94
80	0.89	0.93	0.94	0.95	0.95
60	0.90	0.93	0.93	0.94	0.94
40	0.88	0.92	0.93	0.94	0.93
20	0.86	0.90	0.91	0.92	0.92

Number of hidden layers

(b)

Number of neurons	1	2	3	4	5
100	0.91	0.91	0.92	0.91	0.91
80	0.90	0.93	0.91	0.91	0.92
60	0.89	0.92	0.91	0.92	0.92
40	0.87	0.90	0.91	0.92	0.91
20	0.87	0.89	0.89	0.90	0.92

Number of hidden layers

(c)

	△Train	○Test
R^2	0.93	0.93
RI	0.94	0.96

(d)

Fig. 13 Performance of BNN under various configurations: (**a**) initial settings of BNN; (**b**) on the training set; (**c**) on the testing set; (**d**) predictions based on the optimal configuration

illustrated in Fig. 14, in which the R^2 significantly improved with more iterations before reaching a plateau.

Fig. 14b shows that GPR obtained the optimal performance on the testing set when the number of iterations reached 10. The corresponding optimal amplitude and length scale of each input were presented in Fig. 14a, in which the resulting R^2 values on the training and testing sets were around 0.91 and 0.93, respectively. The corresponding RI values also reached 0.99 and 0.96 in Fig. 14c, with only one measurement outside the predicted 95% CI on the training and testing sets, respectively. These results demonstrate the desirable accuracy, reliability, and generalization ability of GPR for subsequent applications. Although hyper-parameters are optimized for GPR, a Bayesian treatment of them holds the potential to better reveal their effects in predictions.

5.3 Computational Time

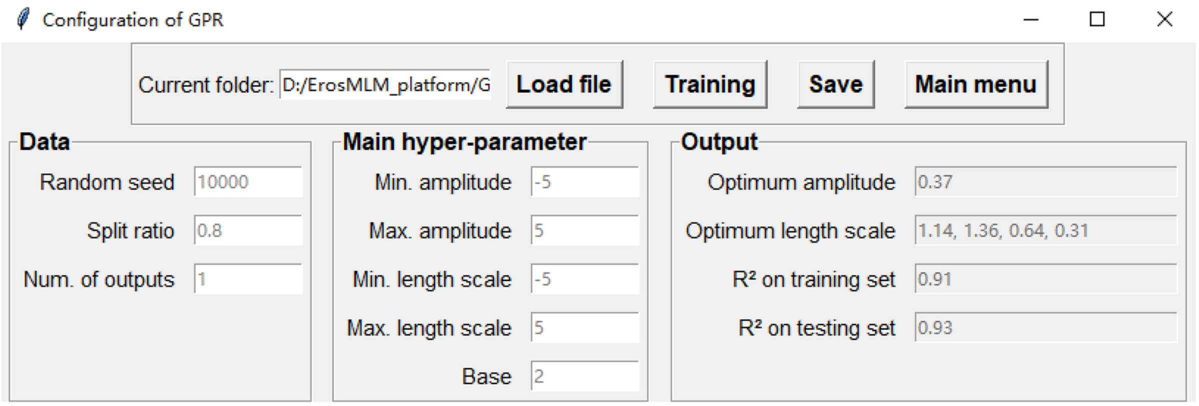
Once all ML-based models have been developed, the corresponding computational times can be obtained and are presented in Fig. 15. The results reveal that GPR is the most time-saving, followed by QRF, ESVR, VIEPR and BNN. Note that the whole training process is conducted at a computer with an Intel Core i9–14900K CPU running at 3.20 GHz and 32 GB of RAM.

5.4 Prediction on New Data

Once ML-based models have been built by the GUI platform, they can be directly saved and applied to make predictions on other data. This can be conducted by anyone using the "Prediction on new data" module, since click is all users need to load data and models to directly generate predictions, as shown in Figs. 7b and 7c. Therefore, 20 datasets from another study were used to examine the feasibility of this module and the practicability of all developed models [92], which are compared with several empirical correlations [13, 118]. The predicted results on new data are presented in Fig. 16, and the performance of all models is compared in Table 4. Apart from empirical equations, the R^2 and RI values of all ML-based models are larger than 0.8 and demonstrate desirable accuracy, reliability and practicability.

5.5 Feature-Importance Analysis

To enhance the interpretability of developed models and understand the impact of input variables on predictions, the



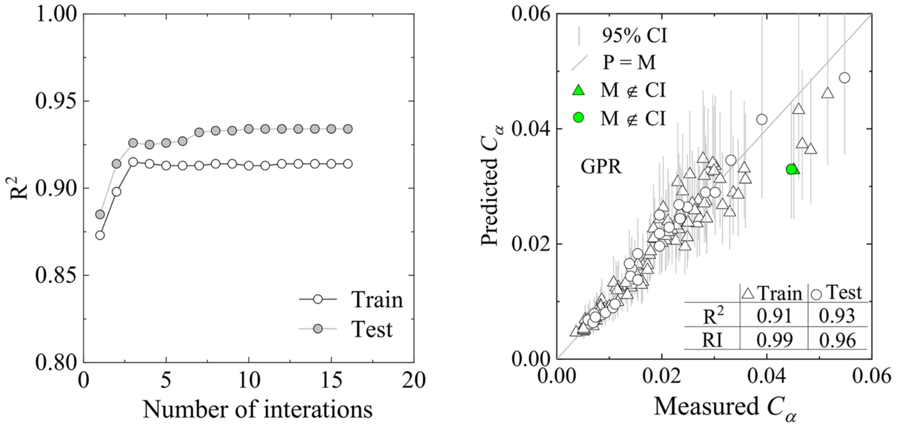
Current folder: D:/ErosMLM_platform/G

Data
Random seed: 10000
Split ratio: 0.8
Num. of outputs: 1

Main hyper-parameter
Min. amplitude: -5
Max. amplitude: 5
Min. length scale: -5
Max. length scale: 5
Base: 2

Output
Optimum amplitude: 0.37
Optimum length scale: 1.14, 1.36, 0.64, 0.31
R² on training set: 0.91
R² on testing set: 0.93

(a)



(b) (c)

Fig. 14 Performance of GPR under various configurations: (**a**) initial settings of GPR; (**b**) on the training and testing sets; (**c**) predictions based on the optimal configuration

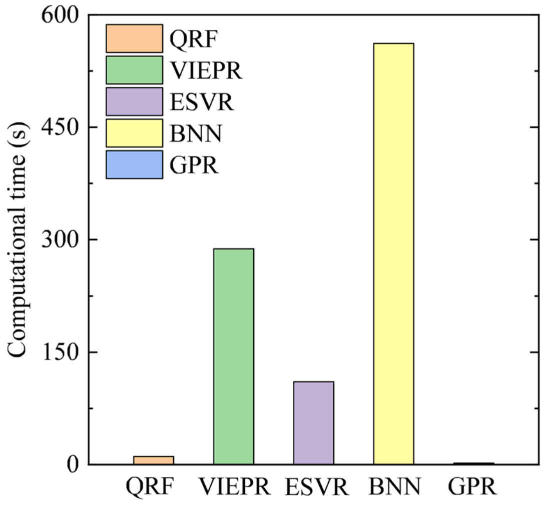


Fig. 15 Computational time for model development

SHapley Additive exPlanations (SHAP) method is used to conduct feature-importance analysis for each model [119]. It utilizes the varying model predictions on different datasets to generate SHAP values for each input variable to represent its importance. A greater SHAP value implies a higher importance and a larger impact on model prediction. Details of applying this method can be found in the previous study [120]. Herein, for each model developed by the platform, the importance of each input variable is quantified by the mean absolute SHAP value on the testing set. Fig. 17 presents the SHAP values of input variables generated by each model and their ranks. The results reveal that each model has identical ranks for importance of input variables, despite diverse SHAP values. Specifically, e_0 is the most influential variable, followed by *PI*, *PL* and *CI*.

5.6 Parametric Analysis

Based on the previous introduction, quantiles of each studied input variable ranging from 20% to 80% with an interval of 10% are first generated using its statistics in the original

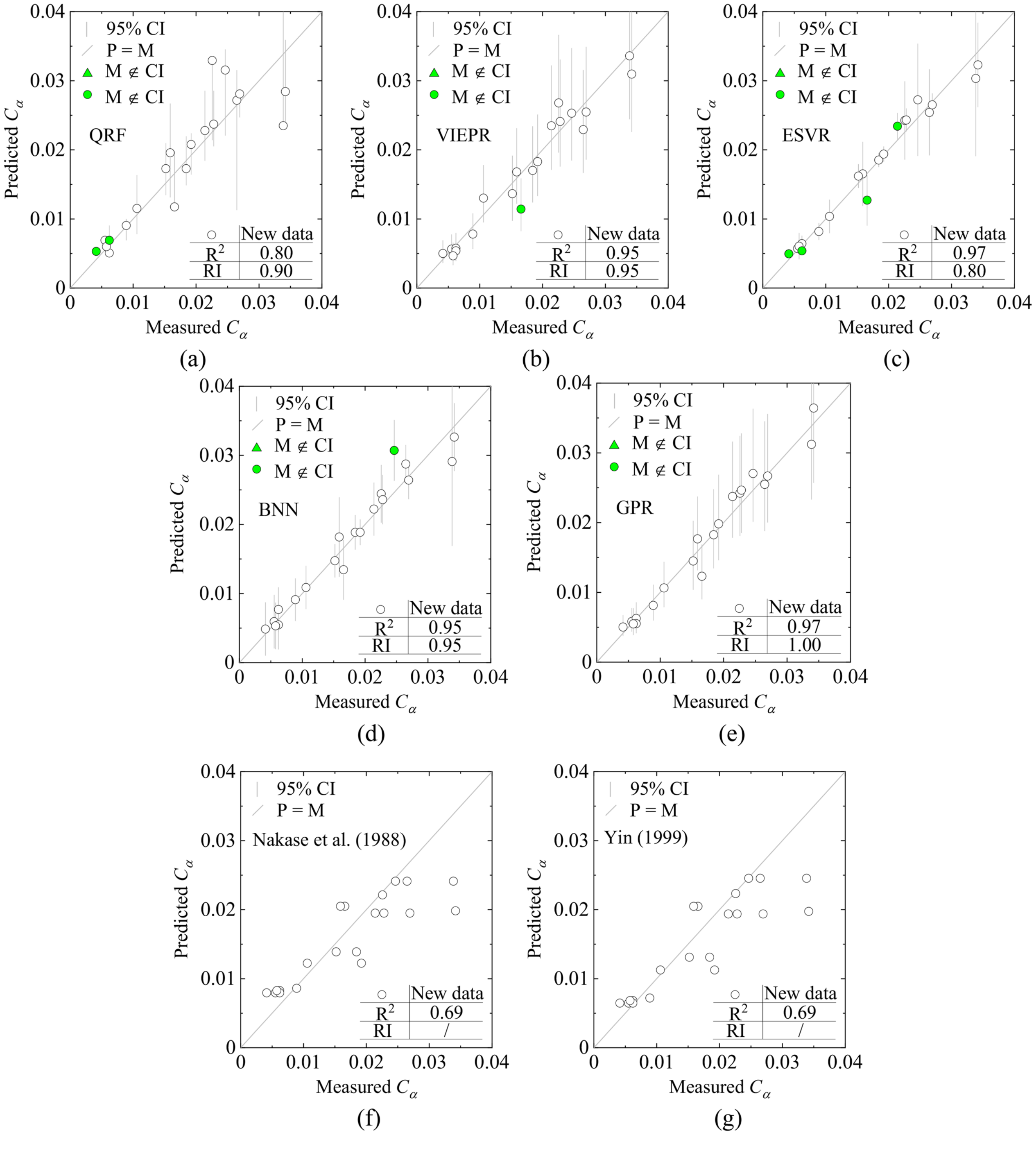


Fig. 16 Prediction on new data: (**a**) QRF; (**b**) VIEPR; (**c**) ESVR; (**d**) BNN; (**e**) GPR; (**f**) Equation [118]; (**g**) Equation [13]

database. The remaining input variables then take their mean values to reveal the captured correlations between the input and output variables of all developed models. The results of parametric analysis on each studied variable were illustrated in Fig. 18, where all inputs and outputs were normalized by their respective maximum to illustrate the impact of input variables and compare captured correlations behind each model.

At first glance, all models captured similar input-output correlations, apart from GPR showing noticeable discrepancies and fluctuations. This is mainly because the merit of kernel-based similarity measure allows GPR to flexibly

Table 4 Comparison of model performance

Model	R^2	RI
QRF	0.80	0.90
VIEPR	0.95	0.95
ESVR	0.97	0.80
BNN	0.95	0.95
GPR	0.97	1.00
Empirical equation [118]	0.69	/
Empirical equation [13]	0.69	/

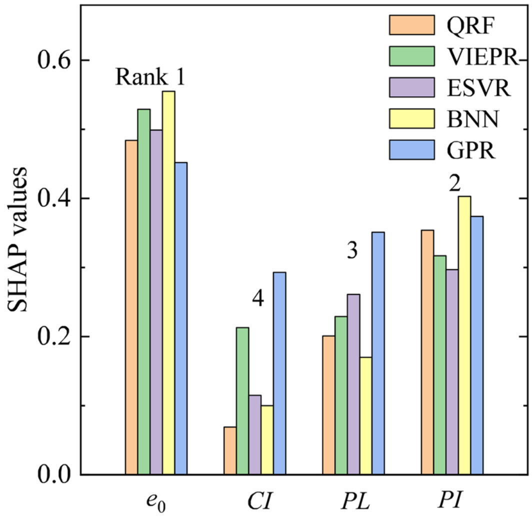


Fig. 17 Results of feature-importance analysis

fit known data and perform better on new data than other models, as shown in Fig. 16. Accordingly, the captured correlation of GPR largely relies on known data and may have higher fluctuations in local space. In contrast, BNN, QRF and ESVR are built on networks, trees, and submodels to learn general correlations and saw relatively smaller fluctuations, with the smoothest curve found at VIEPR. These varying curves indicate various possibilities of underlying correlations and underscore the significance of uncertainty quantification in modelling soil properties.

On the other hand, C_α exhibits strongly positive relationships with e_0 regardless of models, which aligns with the recognition of existing empirical equations [68, 109]. Such an observation also complies with soil mechanics because larger e_0 implies looser soil particles that can be compressed and also a broader space that allows for long-term deformation, thus resulting in larger C_α. In contrast, C_α has significant differences in the relationships with CI, PL and PI among diverse models. Specifically, GPR and VIEPR saw clearly positive correlations with e_0 and CI in predicting C_α, which aligns with their larger impact on making predictions, as shown in Fig. 17. In contrast, the remaining models obtained generally positive correlations with e_0, PL and PI, but no prominent tendency with CI. This is because apart from e_0 directly relating deformation space, larger PL and PI also indicate wide ranges of water contents that can cause

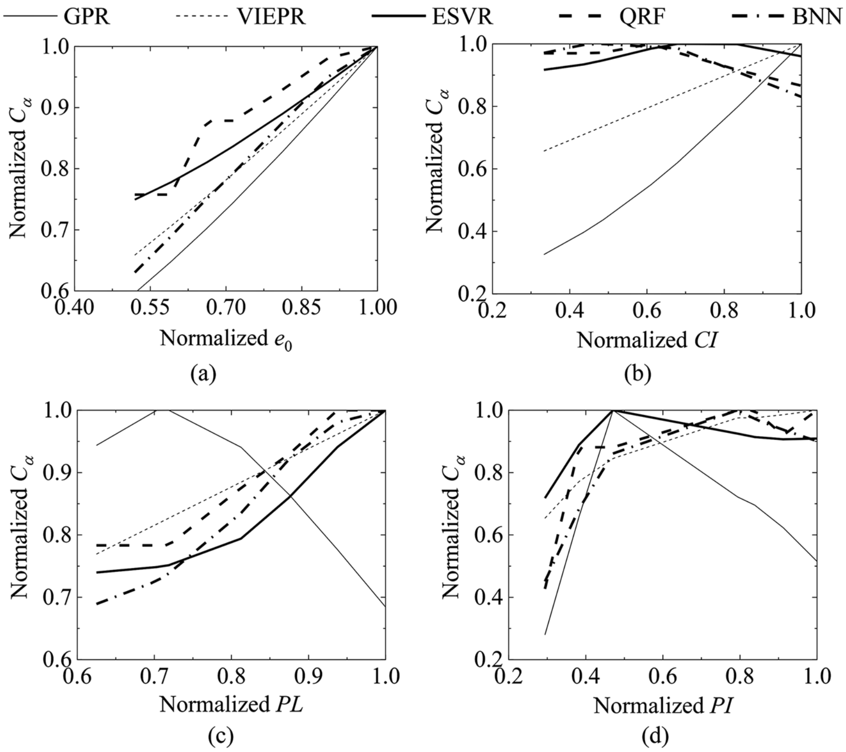


Fig. 18 Captured correlations behind ML-based models: (**a**) $C_a - e_0$; (**b**) $C_a - CI$; (**c**) $C_a - PL$; (**d**) $C_a - PI$

soil elastic and plastic deformation, which somewhat contribute to creep behaviours.

5.7 Active Learning Guided Data-Efficient Modelling

The aforementioned contents demonstrate how to develop and apply models based on existing data, but lack explicit principles for data acquisition once data are insufficient to ensure model performance. To reduce data demand and realize data-efficient modelling, the proposed uncertainty based AL strategy is used to guide data acquisition and applied to cases with sparse data, in comparison with RS. In this section, only 20% of all available data is employed as candidate training data and gradually acquired to develop models, while model performance is compared on the remaining 80% testing data.

Among the five probabilistic methods, GPR has less data demand and can use a single dataset to build a model. In contrast, the remaining methods, such as QRF, cannot construct a tree through a single dataset and need more datasets to build models. Hence, initial datasets for AL based modelling can be case-by-case in practice. Herein, a single dataset is randomly sampled as initial observed data for GPR and is subsequently added one by one through AL or RS until all candidate training data are used. The other four methods use ten randomly sampled datasets as initial observed data before subsequent data acquisition through AL or RS. To prevent randomness and explore different initialisations, the initially observed data are randomly sampled 30 times before subsequent data acquisition. Through the five probabilistic methods, Figs. 19, 20, 21, 22 and 23 present and compare the evolution of model accuracy and reliability, when data were gradually added via AL and RS (red and blue lines). Those lines based on different initializations are displayed in light colour, while their average performance is highlighted by dark colour. Overall, AL-based models achieved a faster increase in R^2 and RI values than RS based models, and obtained larger values when the same number of datasets were observed. This is because AL prioritizes the datasets where the model has less knowledge with larger uncertainty to build models, while the remaining datasets have a limited contribution, thus saving data demand and realizing data-efficient modelling [106]. Nonetheless, the

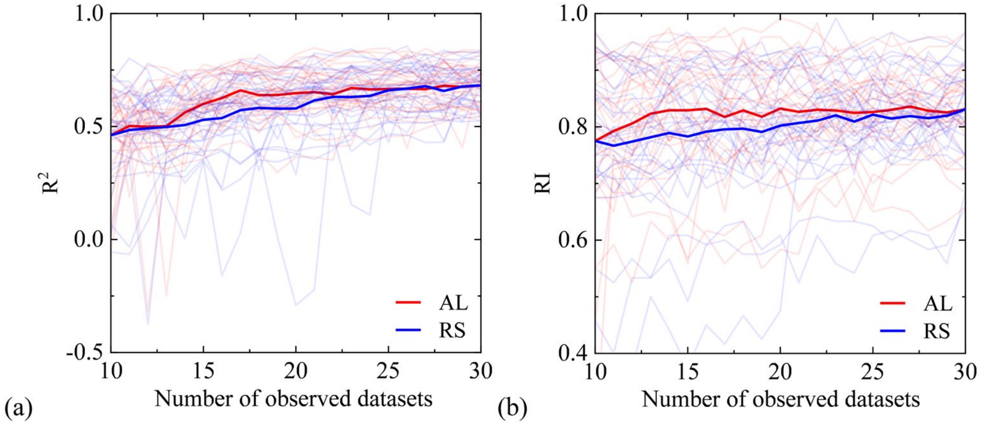


Fig. 19 Evolution of QRF performance in predicting C_α when data are added by active learning and random sampling: (**a**) accuracy; (**b**) reliability

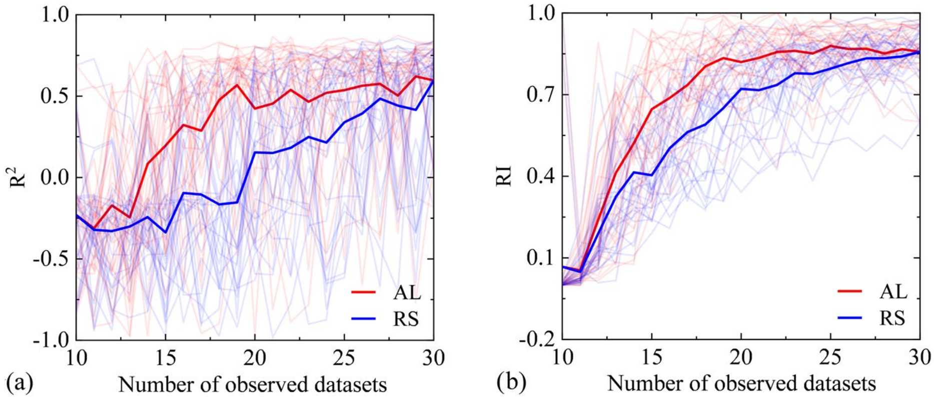


Fig. 20 Evolution of VIEPR performance in predicting C_α when data are added by active learning and random sampling: (**a**) accuracy; (**b**) reliability

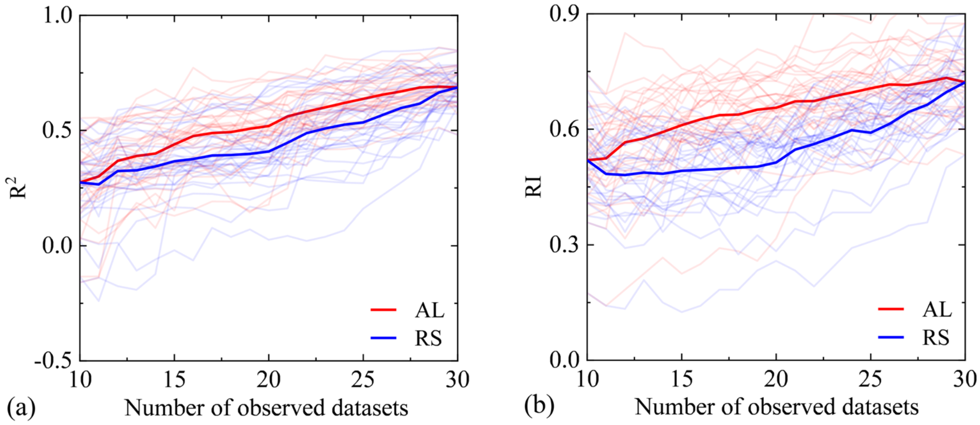


Fig. 21 Evolution of ESVR performance in predicting C_α when data are added by active learning and random sampling: (**a**) accuracy; (**b**) reliability

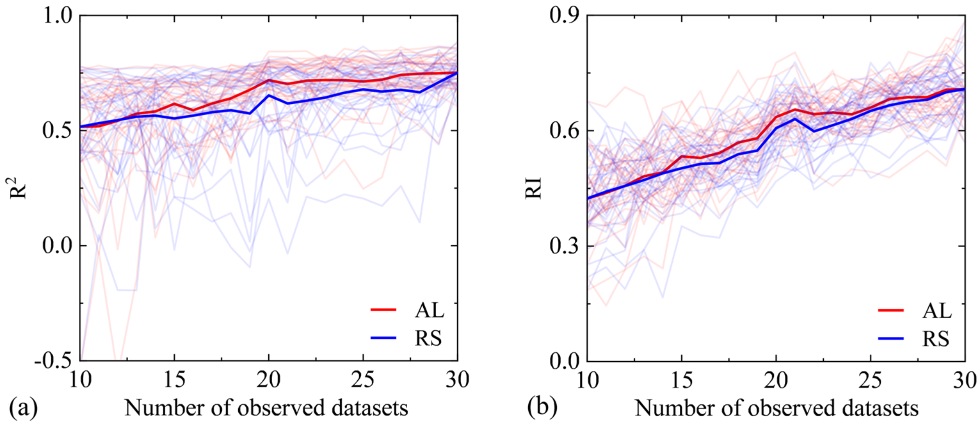


Fig. 22 Evolution of BNN performance in predicting C_α when data are added by active learning and random sampling: (**a**) accuracy; (**b**) reliability

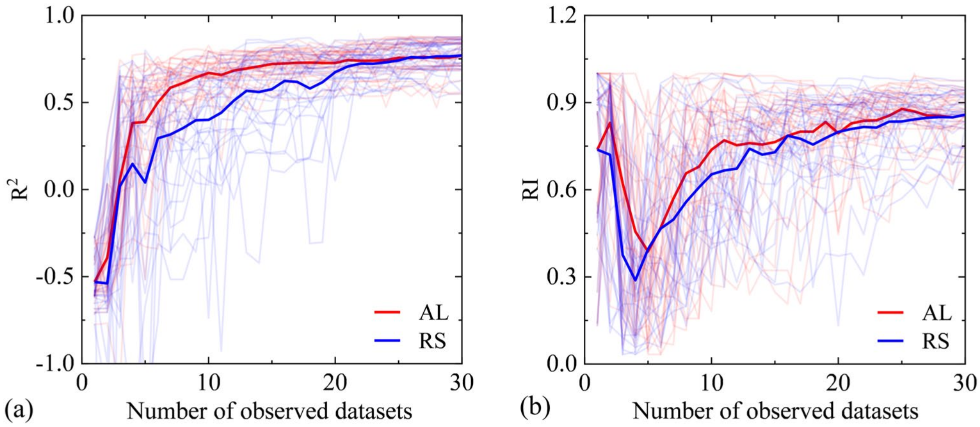


Fig. 23 Evolution of GPR performance in predicting C_α when data are added by active learning and random sampling: (**a**) accuracy; (**b**) reliability

merit of AL diminishes as more and more identical datasets are acquired, because they gradually become sufficient to fill the parametric space and capture target correlations regardless of data acquisition strategies. Another noteworthy phenomenon is that RS based models exhibited notably larger fluctuations on different initializations due to the lack of instructions for data acquisition, while AL based models showed strong robustness.

A different trend was observed for GPR in Fig. 23b, in which both AL and RS based models had a drop at the RI values initially before subsequently continuous growth. This is because models merely observed a single dataset at the beginning, such that captured correlations are only applicable to local areas. Nonetheless, such biased correlations can be calibrated as more data arrive. Overall, the proposed data acquisition strategy achieves highly promising performance for data-efficient modelling of soil properties and indicates substantial potential to be extended to broader applications in geotechnical engineering.

6 Discussion

Although the proposed platform characterizes a user-friendly GUI, probabilistic and data-efficient modelling, some limitations are worth noting to point out valuable extensions for future studies. For example, models developed via the platform have inevitably limited extrapolation ability at the space outside the training domain. To mitigate this issue, effective fusion of big indirect data or pre-trained models with limited site-specific data through hierarchical modelling [121–123], transfer learning [124] or multi-fidelity learning [103] represents promising research areas. Additionally, all compiled probabilistic methods can be integrated through model averaging to build hybrid models [53] and can also be configured to generate multiple outputs. Moreover, the current platform primarily focuses on capturing cross-correlation and can further characterize autocorrelation to improve its practicality in broader engineering scenarios. More specifically, the proposed uncertainty-based AL strategy can be extended to data acquisition site by site if unlabelled data could be distinguished by sites. Furthermore, if the distances between unexplored sites and sources of known data were available, such distances could also enrich the strategy for efficient data acquisition.

7 Conclusions

This paper reviewed the application of machine learning algorithms in modelling soil properties, categorizing them by their underlying mechanisms and highlighting their respective strengths and limitations. Five probabilistic algorithms, including quantile random forest (QRF), variational inference based evolutionary polynomial regression (VIEPR), ensemble based support vector regression (ESVR), Bayesian neural network (BNN) and Gaussian process regression (GPR), were introduced and integrated into the ErosMLM platform. This enables users to develop, optimize, evaluate and apply models with minimal effort. Practicality of the platform was demonstrated through a case study on soil creep index prediction, where all models exhibited high accuracy and reliability. Additionally, an uncertainty-based active learning strategy was proposed to guide efficient data acquisition, significantly reducing data requirements while maintaining model performance. The combination of versatile algorithms, a user-friendly interface, and an intelligent data acquisition framework positions ErosMLM as a powerful tool for advanced probabilistic and data-efficient modelling in geotechnics and beyond.

Acknowledgements This research was financially supported by the National Natural Science Foundation of China, Shenzhen Science and Technology Program, the Research Grants Council (RGC) of the Hong Kong Special Administrative Region Government (HKSARG) of China and the Hong Kong Polytechnic University. Moreover, the authors would like to thank Dr. Pin Zhang and Ms. Jia-Ning Yin for their efforts.

Author Contributions Geng-Fu He: Conceptualization; Methodology; Software; Validation; Formal Analysis; Investigation; Data Curation; Writing—Original Draft; Writing—Review & Editing. Zhen-Yu Yin: Writing—Review & Editing; Supervision; Project Administration; Funding Acquisition. Jidong Zhao: Writing—Review & Editing; Project Administration. Yin-Fu Jin: Writing—Review & Editing; Supervision; Project Administration; Funding Acquisition.

Funding Open access funding provided by The Hong Kong Polytechnic University. This research was financially supported by the National Natural Science Foundation of China (52578420), the Research Grants Council (RGC) of Hong Kong Special Administrative Region Government (HKSARG) of China (E-PolyU501/24, T22-607/24-N, 15220423, 15220221) and the State Key Laboratory of Climate Resilience for Coastal Cities at the Hong Kong Polytechnic University. Open access funding is provided by the Hong Kong Polytechnic University.

Data Availability Data used during the study are available upon reasonable request. (The GUI platform is available at: https://github.com/Gengfu-He/ErosMLM-2025).

Declarations

Competing Interests The authors declare no competing interests.

References

1. Li J, Yin Z-Y (2021) Time integration algorithms for elasto-viscoplastic models with multiple hardening laws for geomaterials: enhancement and comparative study. Arch Comput Methods Eng 28:3869–3886
2. Phoon KK, Cao ZJ, Ji J, Leung YF, Najjar S, Shuku T, Tang C, Yin ZY, Ikumasa Y, Ching J (2022) Geotechnical uncertainty, modeling, and decision making. Soils Found 62:101189
3. Tiwari B, Ajmera B (2012) New correlation equations for compression index of remolded clays. J Geotech Geoenviron Eng 138:757–762
4. Nagaraj T, Srinivasa Murthy B (1986) A critical reappraisal of compression index equations. Geotechnique 36:27–32
5. Phoon K-K, Kulhawy FH (1999) Characterization of geotechnical variability. Can Geotech J 36:612–624
6. Löfman MS, Korkiala-Tanttu LK (2021) Transformation models for the compressibility properties of Finnish clays using a multivariate database. Georisk: Assess Manage Risk Eng Syst Geohazards 16:330–346
7. Ching J, Phoon K-K (2014) Transformations and correlations among some clay parameters-the global database. Can Geotech J 51:663–685
8. Ching J, Phoon K-K, Li K-H, Weng M-C (2019) Multivariate probability distribution for some intact rock properties. Can Geotech J 56:1080–1097
9. Hicher PY (2016) Experimental study of viscoplastic mechanisms in clay under complex loading. Géotechnique 66:661–669
10. Sridharan A, Nagaraj H (2000) Compressibility behaviour of remoulded, fine-grained soils and correlation with index properties. Can Geotech J 37:712–722
11. Phoon K-K, Kulhawy FH (1999) Evaluation of geotechnical property variability. Can Geotech J 36:625–639
12. Yoon GL, Kim BT, Jeon SS (2004) Empirical correlations of compression index for marine clay from regression analysis. Can Geotech J 41:1213–1221
13. Yin JH (1999) Properties and behaviour of Hong Kong marine deposits with different clay contents. Can Geotech J 36:1085–1095
14. Zhang P, Yin ZY, Jin YF (2021) Machine learning-based modelling of soil properties for geotechnical design: review, tool development and comparison. Arch Comput Methods Eng 29:1229–1245
15. Leng Y, Tac V, Calve S, Tepole AB (2021) Predicting the mechanical properties of biopolymer gels using neural networks trained on discrete fiber network data. Comput Methods Appl Mech Eng 387:114160
16. Zhang P, Yin ZY, Jin YF (2021) State-of-the-art review of machine learning applications in constitutive modeling of soils. Arch Comput Methods Eng 28:3661–3686
17. Wang ZZ, Xiao C, Goh SH, Deng M-X (2021) Metamodel-based reliability analysis in spatially variable soils using convolutional neural networks. J Geotech Geoenviron Eng 147:04021003
18. Goh A (1995) Empirical design in geotechnics using neural networks. Geotechnique 45:709–714
19. Song G, He Y, Sheil B, Morris J (2024) Ground settlement prediction for open caisson shafts in sand using a neural network constrained by empiricism. Comput Geotech 166:106001
20. Rangel RL, Gimenez JM, Oñate E, Franci A (2024) A continuum-discrete multiscale methodology using machine learning for thermal analysis of granular media. Comput Geotech 168:106118
21. Huang D, Fuhg JN, Weißenfels C, Wriggers P (2020) A machine learning based plasticity model using proper orthogonal decomposition. Comput Methods Appl Mech Eng 365:113008
22. Qu T, Zhao J, Feng YT (2025) Artificial intelligence for computational granular media. Comput Geotech 185:107310
23. Qu T, Zhao J, Guan S, Feng YT (2023) Data-driven multiscale modelling of granular materials via knowledge transfer and sharing. Int J Plast 171:103786
24. Breiman L (2001) Random forests. Mach Learn 45:5–32
25. Were K, Bui DT, Dick ØB, Singh BR (2015) A comparative assessment of support vector regression, artificial neural networks, and random forests for predicting and mapping soil organic carbon stocks across an Afromontane landscape. Ecol Indic 52:394–403
26. Szabó B, Szatmári G, Takács K, Laborczi A, Makó A, Rajkai K, Pásztor L (2019) Mapping soil hydraulic properties using random-forest-based pedotransfer functions and geostatistics. Hydrol Earth Syst Sci 23:2615–2635
27. Zhang P, Yin ZY, Jin YF, Chan THT, Gao FP (2021) Intelligent modelling of clay compressibility using hybrid meta-heuristic and machine learning algorithms. Geosci Front 12:441–452
28. Salazar F, Toledo M, Oñate E, Morán R (2015) An empirical comparison of machine learning techniques for dam behaviour modelling. Struct Saf 56:9–17
29. Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2:1–27
30. Samui P (2008) Prediction of friction capacity of driven piles in clay using the support vector machine. Can Geotech J 45:288–295
31. Achieng KO (2019) Modelling of soil moisture retention curve using machine learning techniques: artificial and deep neural networks vs support vector regression models. Comput Geosci 133:104320
32. Ballabio C (2009) Spatial prediction of soil properties in temperate mountain regions using support vector regression. Geoderma 151:338–350
33. Kovačević M, Bajat B, Gajić B (2010) Soil type classification and estimation of soil properties using support vector machines. Geoderma 154:340–347
34. Das SK, Samui P, Sabat AK (2012) Prediction of field hydraulic conductivity of clay liners using an artificial neural network and support vector machine. Int J Geomech 12:606–611
35. Kohestani V, Hassanlourad M (2016) Modeling the mechanical behavior of carbonate sands using artificial neural networks and support vector machines. Int J Geomech 16:04015038
36. Rezania M, Javadi AA, Giustolisi O (2010) Evaluation of liquefaction potential based on CPT results using evolutionary polynomial regression. Comput Geotech 37:82–92
37. Rezania M, Faramarzi A, Javadi AA (2011) An evolutionary based approach for assessment of earthquake-induced soil liquefaction and lateral displacement. Eng Appl Artif Intell 24:142–153
38. Ghorbani A, Hasanzadehshooiili H (2018) Prediction of UCS and CBR of microsilica-lime stabilized sulfate silty sand using ANN and EPR models; application to the deep soil mixing. Soils Found 58:34–49
39. Cheng Z-L, Zhou W-H, Garg A (2020) Genetic programming model for estimating soil suction in shallow soil layers in the vicinity of a tree. Eng Geol 268:105506
40. Yin ZY, Jin YF, Huang HW, Shen SL (2016) Evolutionary polynomial regression based modelling of clay compressibility using an enhanced hybrid real-coded genetic algorithm. Eng Geol 210:158–167
41. Yin ZY, Jin YF, Shen SL, Huang HW (2016) An efficient optimization method for identifying parameters of soft structured clay by an enhanced genetic algorithm and elastic-viscoplastic model. Acta Geotech 12:849–867
42. Zohdi T (2022) A digital-twin and machine-learning framework for precise heat and energy management of data-centers. Comput Mech 69:1501–1516
43. Sanikhani H, Deo RC, Yaseen ZM, Eray O, Kisi O (2018) Non-tuned data intelligent model for soil temperature estimation: a new approach. Geoderma 330:52–64

44. Asteris PG, Mamou A, Ferentinou M, Tran T-T, Zhou J (2022) Predicting clay compressibility using a novel Manta ray foraging optimization-based extreme learning machine model. Transp Geotech 37:100861
45. Dong S, Li Z (2021) A modified batch intrinsic plasticity method for pre-training the random coefficients of extreme learning machines. J Comput Phys 445:110585
46. Bardhan A, Kardani N, Alzo'ubi AK, Roy B, Samui P, Gandomi AH (2022) Novel integration of extreme learning machine and improved Harris hawks optimization with particle swarm optimization-based mutation for predicting soil consolidation parameter. J Rock Mech Geotech Eng 14:1588–1608
47. Zhang R, Li Y, Goh ATC, Zhang W, Chen Z (2021) Analysis of ground surface settlement in anisotropic clays using extreme gradient boosting and random forest regression models. J Rock Mech Geotech Eng 13:1478–1484
48. Wang ZZ (2022) Deep learning for geotechnical reliability analysis with multiple uncertainties. J Geotech Geoenviron Eng 148:06022001
49. Zhang J, Zhang L, Tang WH (2009) Bayesian framework for characterizing geotechnical model uncertainty. J Geotech Geoenviron Eng 135:932–940
50. Zhang J, Tang WH, Zhang LM, Huang HW (2012) Characterising geotechnical model uncertainty by hybrid Markov chain Monte Carlo simulation. Comput Geotech 43:26–36
51. Phoon K-K, Tang C (2019) Characterisation of geotechnical model uncertainty. Georisk: Assess Manage Risk Eng Syst Geohazards 13:101–130
52. Cao Z-J, Zheng S, Li D-Q, Phoon K-K (2019) Bayesian identification of soil stratigraphy based on soil behaviour type index. Can Geotech J 56:570–586
53. Wang ZZ, Hu Y, Guo X, He X, Kek HY, Ku T, Goh SH, Leung CF (2023) Predicting geological interfaces using stacking ensemble learning with multi-scale features. Can Geotech J 60:1036–1054
54. Scillitoe A, Seshadri P, Girolami M (2021) Uncertainty quantification for data-driven turbulence modelling with mondrian forests. J Comput Phys 430:110116
55. Zhang P, Yin ZY, Jin YF (2022) Bayesian neural network-based uncertainty modelling: application to soil compressibility and undrained shear strength prediction. Can Geotech J 59:546–557
56. Gal Y, Ghahramani Z (2015) Dropout as a Bayesian approximation: representing model uncertainty in deep learning. arXiv: 1506. 02142
57. He G-F, Zhang P, Yin Z-Y, Goh SH (2024) Multifidelity-based Gaussian process for quasi-site-specific probabilistic prediction of soil properties. Can Geotech J 61:2304–2322
58. Sharma A, Wang H, Zhang J, Lu M, Wu C (2024) Constructing multivariate distribution of rainfall characteristics: a Bayesian vine algorithm. J Hydrol 637:131392
59. Ching J, Phoon K-K (2012) Modeling parameters of structured clays as a multivariate normal distribution. Can Geotech J 49:522–545
60. Meinshausen N, Ridgeway G (2006) Quantile regression forests. J Mach Learn Res 7:983–999
61. Collico S, Spagnoli G, Monforte L (2025) Automating site characterization from pile field data. Comput Geotech 187:107498
62. Jin Y-F, Yin Z-Y, Zhou W-H, Horpibulsuk S (2019) Identifying parameters of advanced soil models using an enhanced transitional Markov chain Monte Carlo method. Acta Geotech 14:1925–1947
63. Li R, Yin Z-Y, He S-H (2025) 3D reconstruction of arbitrary granular media utilizing vision foundation model. Appl Soft Comput 169:112599
64. Wang H, Liu L, Shi M, Yang J, Song X, Zhang C, Tao D (2024) Active learning framework for tunnel geological reconstruction based on TBM operational data. Automation Construct 158:105230
65. Rahimi M, Wang Z, Shafieezadeh A, Wood D, Kubatko EJ (2020) Exploring passive and active metamodeling-based reliability analysis methods for soil slopes: a new approach to active training. Int J Geomech 20:04020009
66. Zhang W, Zhang Y, Goh ATC (2017) Multivariate adaptive regression splines for inverse analysis of soil and wall properties in braced excavation. Tunn Undergr Space Technol 64:24–33
67. Rashed A, Bazaz JB, Alavi AH (2012) Nonlinear modeling of soil deformation modulus through LGP-based interpretation of pressuremeter test results. Eng Appl Artif Intell 25:1437–1449
68. Jin YF, Yin ZY, Zhou WH, Yin JH, Shao JF (2019) A single-objective EPR based model for creep index of soft clays considering L2 regularization. Eng Geol 248:242–255
69. Mahmoodzadeh A, Nejati HR, Mohammadi M, Ibrahim HH, Rashidi S, Ibrahim BF (2022) Forecasting face support pressure during EPB shield tunneling in soft ground formations using support vector regression and meta-heuristic optimization algorithms. Rock Mech Rock Eng 55:6367–6386
70. Zhang J, Lin C, Tang H, Wen T, Tannant DD, Zhang B (2024) Input-parameter optimization using a SVR based ensemble model to predict landslide displacements in a reservoir area - a comparative study. Appl Soft Comput 150:111107
71. Singh VK, Kumar D, Kashyap PS, Singh PK, Kumar A, Singh SK (2020) Modelling of soil permeability using different data driven algorithms based on physical properties of soil. J Hydrol 580:124223
72. Liu C, Macedo J (2022) Machine learning-based models for estimating seismically-induced slope displacements in subduction earthquake zones. Soil Dyn Earthquake Eng 160:107323
73. Zhang P, Yin ZY, Chen Q (2022) Image-based 3D reconstruction of granular grains via hybrid algorithm and level set with convolution kernel. J Geotech Geoenviron Eng 148:04022021
74. Boubou R, Emeriault F, Kastner R (2010) Artificial neural network application for the prediction of ground surface movements induced by shield tunnelling. Can Geotech J 47:1214–1233
75. Zhang P, Yin ZY (2021) A novel deep learning-based modelling strategy from image of particles to mechanical properties for granular materials with CNN and BiLSTM. Comput Methods Appl Mech Eng 382:113858
76. Wang M, Wang E, Liu X, Wang C (2023) Topological graph representation of stratigraphic properties of spatial-geological characteristics and compression modulus prediction by mechanism-driven learning. Comput Geotech 153:105112
77. Waqas U, Ahmed MF (2022) Investigation of strength behavior of thermally deteriorated sedimentary rocks subjected to dynamic cyclic loading. Int J Rock Mech Min Sci 158:105201
78. Yoshida I, Tomizawa Y, Otake Y (2021) Estimation of trend and random components of conditional random field using Gaussian process regression. Comput Geotech 136:104179
79. Zheng H, Mooney M, Gutierrez M (2023) Surrogate model for 3D ground and structural deformations in tunneling by the sequential excavation method. Comput Geotech 154:105142
80. Pinto F, Torres C, Birrell M, Li Y, Fayaz J, Astroza R (2025) Probabilistic characterization of inherent and epistemic geotechnical uncertainty in soil constitutive models using polynomial chaos expansion and monotonic drained triaxial tests. Comput Geotech 186:107361
81. Mentani A, Govoni L, Bourrier F, Zabatta R (2023) Metamodelling of the load-displacement response of offshore piles in sand. Comput Geotech 159:105490
82. Yoon H, Jun S-C, Hyun Y, Bae G-O, Lee K-K (2011) A comparative study of artificial neural networks and support vector

machines for predicting groundwater levels in a coastal aquifer. J Hydrol 396:128–138

83. Zhang W, Wu C, Li Y, Wang L, Samui P (2019) Assessment of pile drivability using random forest regression and multivariate adaptive regression splines. Georisk: Assess Manage Risk Eng Syst Geohazards 15:27–40
84. Zhang P, Chen R-P, Wu H-N (2019) Real-time analysis and regulation of EPB shield steering using random forest. Automation Construct 106:102860
85. Qi C, Tang X (2018) A hybrid ensemble method for improved prediction of slope stability. Int J Numer Anal Methods Geomech 42:1823–1839
86. D'Ignazio M, Phoon K-K, Tan SA, Länsivaara TT (2016) Correlations for undrained shear strength of Finnish soft clays. Can Geotech J 53:1628–1645
87. Ching J, Lin G-H, Chen J-R, Phoon K-K (2017) Transformation models for effective friction angle and relative density calibrated based on generic database of coarse-grained soils. Can Geotech J 54:481–501
88. Bardhan A, Kardani N, Alzo'ubi AK, Samui P, Gandomi AH, Gokceoglu C (2022) A comparative analysis of hybrid computational models constructed with swarm intelligence algorithms for estimating soil compression index. Arch Comput Methods Eng 29:4735–4773
89. Park HI, Lee SR (2011) Evaluation of the compression index of soils using an artificial neural network. Comput Geotech 38:472–481
90. Pham K, Nguyen K, Lim K, Kim Y, Choi H (2024) A generalized formula for predicting soil compression index using multi-evolutionary algorithm. Eng Geol 343:107789
91. Bardhan A, GuhaRay A, Gupta S, Pradhan B, Gokceoglu C (2022) A novel integrated approach of ELM and modified equilibrium optimizer for predicting soil compression index of subgrade layer of dedicated freight corridor. Transp Geotech 32:100678
92. Zhang P, Yin ZY, Jin YF, Chan THT (2020) A novel hybrid surrogate intelligent model for creep index prediction based on particle swarm optimization and random forest. Eng Geol 265:105328
93. Padarian J, Minasny B, McBratney AB (2022) Assessing the uncertainty of deep learning soil spectral models using Monte Carlo dropout. Geoderma 425:116063
94. He G-F, Yin Z-Y, Zhang P (2025) Uncertainty quantification in data-driven modelling with application to soil properties prediction. Acta Geotech 20:843–859
95. He G-F, Zhang P, Yin Z-Y (2025) Uncertainty quantification in tree structure and polynomial regression algorithms toward material indices prediction. Data-Centric Eng 6:e20
96. Li KQ, Yin ZY, Liu Y (2023) A hybrid SVR-BO model for predicting the soil thermal conductivity with uncertainty. Can Geotech J 61:258–274
97. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22:79–86
98. Olivier A, Shields MD, Graham-Brady L (2021) Bayesian neural networks for uncertainty quantification in data-driven materials modeling. Comput Methods Appl Mech Eng 386:114079
99. Ching J, Phoon KK, Wu CT (2022) Data-centric quasi-site-specific prediction for compressibility of clays. Can Geotech J 59:2033–2049
100. Wang HL, Yin ZY, Zhang P, Jin YF (2020) Straightforward prediction for air-entry value of compacted soils using machine learning algorithms. Eng Geol 279:105911
101. Zhang P, Yin Z-Y, Sheil B (2023) A physics-informed data-driven approach for consolidation analysis. Géotechnique 74:620–631
102. Williams CK, Rasmussen CE (2006) Gaussian processes for machine learning. MIT Press, Cambridge, MA
103. He G-F, Zhang P, Yin Z-Y (2024) Active learning inspired multi-fidelity probabilistic modelling of geomaterial property. Comput Methods Appl Mech Eng 432:117373
104. Di Fiore F, Nardelli M, Mainini L (2024) Active learning and Bayesian optimization: a unified perspective to learn with a goal. Arch Comput Methods Eng 31:2985–3013
105. Zhou T, Peng Y (2022) Ensemble of metamodels-assisted probability density evolution method for structural reliability analysis. Reliab. Eng. Syst. Saf 228:108778
106. Li K, Persaud D, Choudhary K, DeCost B, Greenwood M, Hattrick-Simpers J (2023) Exploiting redundancy in large materials datasets for efficient machine learning with less data. Nat Commun 14:7283
107. Zhang P, Sheil B, Girolami M (2025) Active learning informed proper orthogonal decomposition for reduced order modelling of heat transfer in porous medium. Comput Methods Appl Mech Eng 444:118174
108. Zeng L, Hong Z, Liu S, Chen F (2012) Variation law and quantitative evaluation of secondary consolidation behavior for remolded clays. Chin J Geotech Eng 34:1496–1500
109. Yin Z-Y, Xu Q, Yu C (2015) Elastic-viscoplastic modeling for natural soft clays considering nonlinear creep. Int J Geomech 15:A6014001
110. Li Q, Ng C, Liu G-B (2012) Low secondary compressibility and shear strength of Shanghai clay. J Cent South Univ (Engl Ed) 19:2323–2332
111. Zhu Q-Y, Jin Y-F, Yin Z-Y, Hicher P-Y (2013) Influence of natural deposition plane orientation on oedometric consolidation behavior of three typical clays from southeast coast of China. J Zhejiang Univ, Sci, A 14:767–777
112. Agbaje S, Zhang X, Patelli E, Ward D, Dhimitri L (2024) Random field failure and post-failure analyses of vertical slopes in soft clays. Comput Geotech 166:106037
113. Kannangara KPM, Su L-J, Zhou W-H (2024) Analysis of post-ground settlement induced during twin tunnelling in silty sand. Tunn Undergr Space Technol 152:105949
114. Zhu Q-Y, Jin Y-F, Yin Z-Y (2020) Modeling of embankment beneath marine deposited soft sensitive clays considering straightforward creep degradation. Mar Georesour Geotechnol 38:553–569
115. Zhao T, Wang Y (2018) Simulation of cross-correlated random field samples from sparse measurements using Bayesian compressive sensing. Mech Syst Sig Process 112:384–400
116. Lyu B, Hu Y, Wang Y (2023) Data-driven development of three-dimensional subsurface models from sparse measurements using Bayesian compressive sampling: a benchmarking study. ASCE-ASME J Risk Uncertain Eng Syst Part A: Civ Eng 9:04023010
117. Zhang P (2019) A novel feature selection method based on global sensitivity analysis with application in machine learning-based prediction model. Appl Soft Comput 85:105859
118. Nakase A, Kamei T, Kusakabe O (1988) Constitutive parameters estimated by plasticity index. J Geotech Eng 114:844–858
119. Li X, Wang H, Qin S, Lin L, Wang X, Cornelis W (2024) Evaluating ensemble learning in developing pedotransfer functions to predict soil hydraulic properties. J Hydrol 640:131658
120. Ou J, Luo X, Liu J, Huang L, Zhou L, Yuan Y (2023) Predicting microbial extracellular electron transfer activity in paddy soils with soil physicochemical properties using machine learning. Sci China Technol Sci 67:259–270
121. Collico S, Spagnoli G, Romero E, Fraccica A (2024) A Bayesian clustered-multilevel updating for local undrained shear strength prediction of fine-grained soils. Appl Clay Sci 257:107444
122. Sharma A, Ching J, Phoon K-K (2023) A spectral algorithm for quasi-regional geotechnical site clustering. Comput Geotech 161:105624

123. Ching J, Wu S, Phoon K-K (2021) Constructing quasi-site-specific multivariate probability distribution using hierarchical Bayesian model. J Eng Mech 147:04021069
124. Rossel RAV, Shen Z, Lopez LR, Behrens T, Shi Z, Wetterlind J, Sudduth KA, Stenberg B, Guerrero C, Gholizadeh A (2024) An imperative for soil spectroscopic modelling is to think global but fit local with transfer learning. Earth Sci Rev 254:104797

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.