

Enhancing deformation characteristics prediction of coarse-grained soils with time-series generative adversarial network-based data augmentation and pre-training

Ying ZHANG^a, Meng JIA^a, Xuedong ZHANG^{b*}, Liping CAO^a, Ziyang AN^a, Hongchao WANG^a, Jinyu WANG^a

^a School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China

^b The State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources and Hydropower Research, Beijing 100038, China

*Corresponding author. E-mail: zxerdo@126.com

© Higher Education Press 2025

ABSTRACT Coarse-grained soils are fundamental to major infrastructures like embankments, roads, and bridges. Understanding their deformation characteristics is essential for ensuring structural stability. Traditional methods, such as triaxial compression tests and numerical simulations, face challenges like high costs, time consumption, and limited generalizability across different soils and conditions. To address these limitations, this study employs deep learning to predict the volumetric strain of coarse-grained soils as axial strain changes, aiming to obtain the axial strain (ε_a)–volumetric strain (ε_v) curve, which helps derive key mechanical parameters like cohesion (c), and elastic modulus (E). However, the limited data from triaxial tests poses challenges for training deep learning models. We propose using a Time-series Generative Adversarial Network (TimeGAN) for data augmentation. Additionally, we apply feature importance analysis to assess the quality of the numerical augmented data, providing feedback for improving the TimeGAN model. To further enhance model performance, we introduce the pre-training strategy to reduce bias between augmented and real data. Experimental results demonstrate that our approach effectively predicts ε_a – ε_v curve, with the mean absolute error (MAE) of 0.2219 and the R^2 of 0.9155. The analysis aligns with established findings in soil mechanics, underscoring the potential of our method in engineering applications.

KEYWORDS coarse-grained soils, deformation characteristics, TimeGAN, data augmentation, pre-training

1 Introduction

In civil engineering, research on the deformation characteristics of coarse-grained soils is essential for understanding their behavior under external loading. It improves soil management in construction and reduces economic risks from geological disasters [1].

The triaxial compression test is crucial for examining the mechanical properties and deformation behavior of coarse-grained soils. It enables the calculation of key parameters like cohesion (c), internal friction angle (φ), and elastic modulus (E) [2–5], which characterize soil

response under various stress conditions. Specifically, cohesion (c) represents the shear strength of soil in kPa without external pressure, the internal friction angle (φ) quantifies soil resistance due to particle friction in degrees, and the elastic modulus (E) denotes the soil's capacity to deform elastically under stress, typically expressed in MPa. These parameters also support the development of constitutive models to simulate soil deformation and strength. However, traditional triaxial tests and constitutive models are often based on limited experimental data and fixed conditions, which restrict their broader applicability and generalizability in diverse real-world scenarios.

In computer science, machine learning methods such as

K-Nearest Neighbors, Support Vector Machines, and the Multilayer Perceptron neural network have been used to predict the strength and deformation characteristics of coarse-grained soils [6–9]. Additionally, Anitescu et al. [10] and Samaniego et al. [11] proposed neural network-based numerical methods for solving second-order boundary value problems and partial differential equations. These two studies provide valuable insight into applying machine learning techniques and knowledge from mathematical physics to soil mechanics problems.

The axial strain (ε_a)–volumetric strain (ε_v) curves of coarse-grained soils are influenced by complex nonlinear factors. Traditional machine learning methods, reliant on domain knowledge, struggle to capture these intricate relationships. In contrast, deep learning [12] effectively models these nonlinear interactions. However, deep learning requires large-scale data, which is often limited for coarse-grained soils due to the complexity and high cost of triaxial compression tests, resulting in insufficient real data samples.

Numerical simulation methods, such as the Discrete Element Method, can simulate coarse-grained soil behavior, partially addressing data scarcity [13]. However, factors like particle size, shape, and interactions result in discrepancies between simulations and real conditions. These discrepancies introduce systematic errors, leading to inaccurate model predictions. Besides, deep learning methods can also be used to augment training data samples. For image data, techniques like rotation, scaling, flipping, and color adjustments can create diverse data sets, improving model robustness and generalization [14,15]. For numerical data, such as data sets collected from coarse-grained soils, AutoEncoders (AEs) and Generative Adversarial Networks (GANs) can generate data samples. Models such as AE and their variant, variational autoencoder, have been extended for small-scale data sets to address the issue of insufficient training samples in regression problems [16–18]. However, AEs replicate learned patterns, limiting output diversity. GANs [19], based on a game-theoretic framework, continuously improve generator and discriminator networks, producing highly realistic data and effectively augmenting the training data set [20,21].

The deformation characteristics of coarse-grained soils exhibit temporal dependencies. To preserve these dependencies, it is necessary to consider both the distribution at a single time step, i.e., $p(x_t)$, and the conditional distribution between data points, i.e., $p(x_t|x_{1:t-1})$. Traditional GANs lack the capability to capture such temporal relationships. To address this, Yoon et al. [22] introduced Time-Series Generative Adversarial Networks (TimeGANs), designed to capture dependencies in time-series data. TimeGANs help

overcome the challenge of limited time-series data [23–25], enabling more effective application of deep learning models in soil mechanics through data augmentation.

For the generated samples, their quality can be evaluated visually for image data or semantically for text data. However, no unified standard exists for assessing the quality of numerical data, especially the data of coarse-grained soils. In this paper, we use feature importance analysis to evaluate the quality of generated data by calculating correlations between soil physical properties (e.g., confining pressure, dry density and so on) and mechanical behavior (e.g., volumetric strain). This method provides feedback for model training and guides the generation of high-quality data samples.

In studying the impact of generated data on model performance, Wang et al. [26] pointed out that generated data (even from a good diffusion model) can sometimes be detrimental to deep learning, such as in contrastive learning. Usually, it is inevitable that there will be biases between the generated data and the real data. To address this issue, this paper introduces the pre-training strategy [27] aimed at adjusting model parameters to better align with the distribution of real data. This approach seeks to reduce the adverse effects of bias on model training and enhance both the performance and robustness of the model. The proposed method involves initially training the model on a substantial amount of generated data, followed by fine-tuning on a smaller real data set.

The key contributions of this paper are outlined as follows.

The traditional approach uses triaxial compression tests to determine soil mechanical parameters. These parameters are sample-specific and require complex calculations, limiting their generalizability. Our deep learning method directly obtains the ε_a – ε_v curve for various types of coarse-grained soils with high accuracy, reducing the time and cost of data acquisition.

Due to limited experimental equipment and time constraints, data samples for coarse-grained soils are often inadequate. We propose using TimeGAN for data augmentation based on the analysis of soil data samples. Then we assess the quality of the augmented data through feature importance analysis and validate it against established civil engineering principles to enhance its interpretability.

Bias between real and augmented data can introduce systematic errors in prediction models. To mitigate this bias, we implement the pre-training strategy that enhances the performance of deep learning models, improving predictions of volumetric strain and the ε_a – ε_v curve, as well as achieving a closer fit between true and predicted curves.

2 Methodology

In this study, we aim to predict the deformation behavior of coarse-grained soils, specifically the volumetric strain (ε_v), to obtain the ε_a - ε_v curve, which helps in obtaining the mechanical parameters of the soil body related to the deformation characteristics of coarse-grained soils based on collected ε_a - ε_v relationship curves. The study is primarily divided into two stages. The first stage addresses the problem of the low quantity of data samples obtained from triaxial compression tests of coarse-grained soils. By considering the real data features and temporal dependencies of features, the TimeGAN is introduced to augment data samples. Additionally, the feature importance analysis method is employed to assess the quality of the augmented data, providing feedback to guide the training of the TimeGAN model. Subsequently, the data bias between real and enhanced data necessitates pre-training in the second stage. During the pre-training phase, the model is trained on augmented data to learn

general knowledge, then initialized with pre-trained weights, and finally fine-tuned on real data. By using this two-stage process, the model benefits from the general knowledge acquired during pre-training and adapts to the specific details of the real data during fine-tuning. This approach leads to better performance even with a limited amount of real data and mitigates the detrimental effects of augmented data on model training.

The whole structure of the proposed approach is depicted in Fig. 1.

2.1 Data augmentation based on time-series generative adversarial network

As depicted in Fig. 2, TimeGAN comprises two main components: the Autoencoding Component and the Adversarial Component. The Autoencoding Component is designed to learn and capture the temporal dependencies in sequential data. This component uses an AutoEncoder architecture to encode and decode time

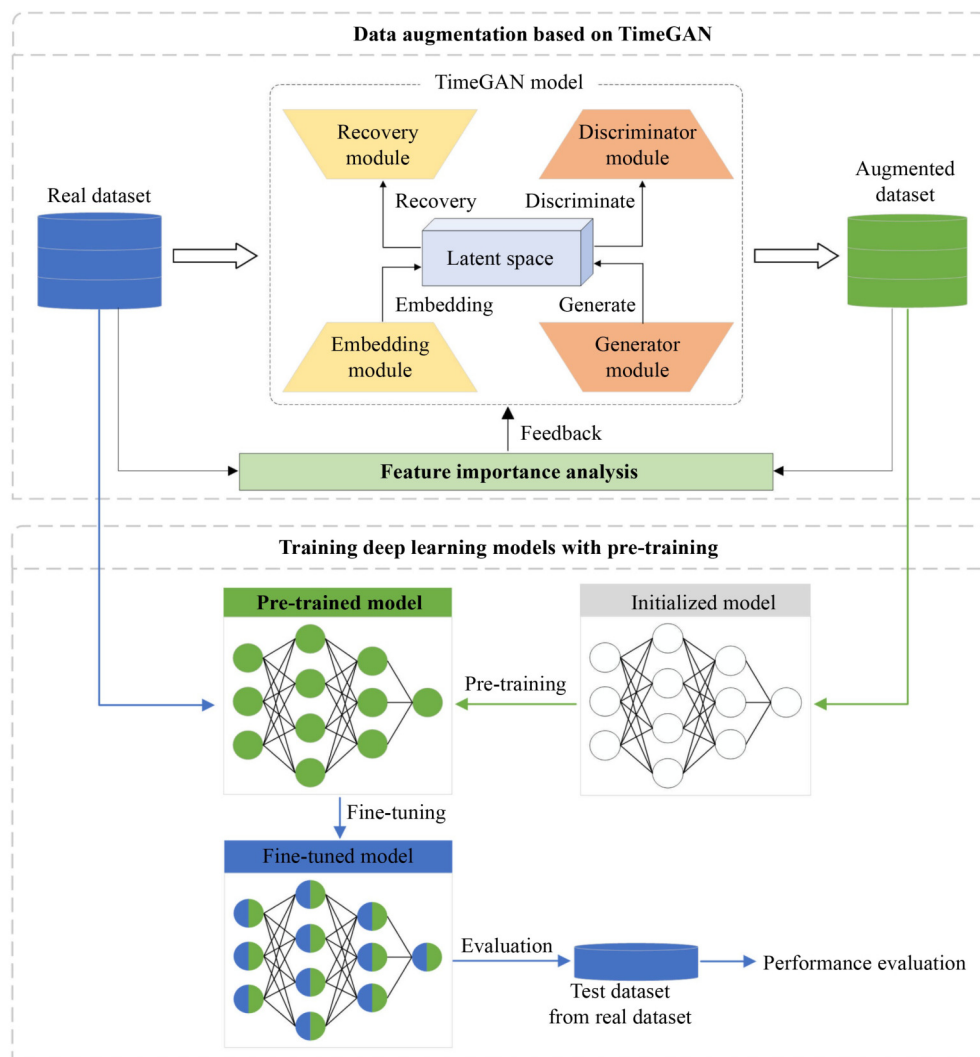


Fig. 1 The whole structure of the proposed approach.

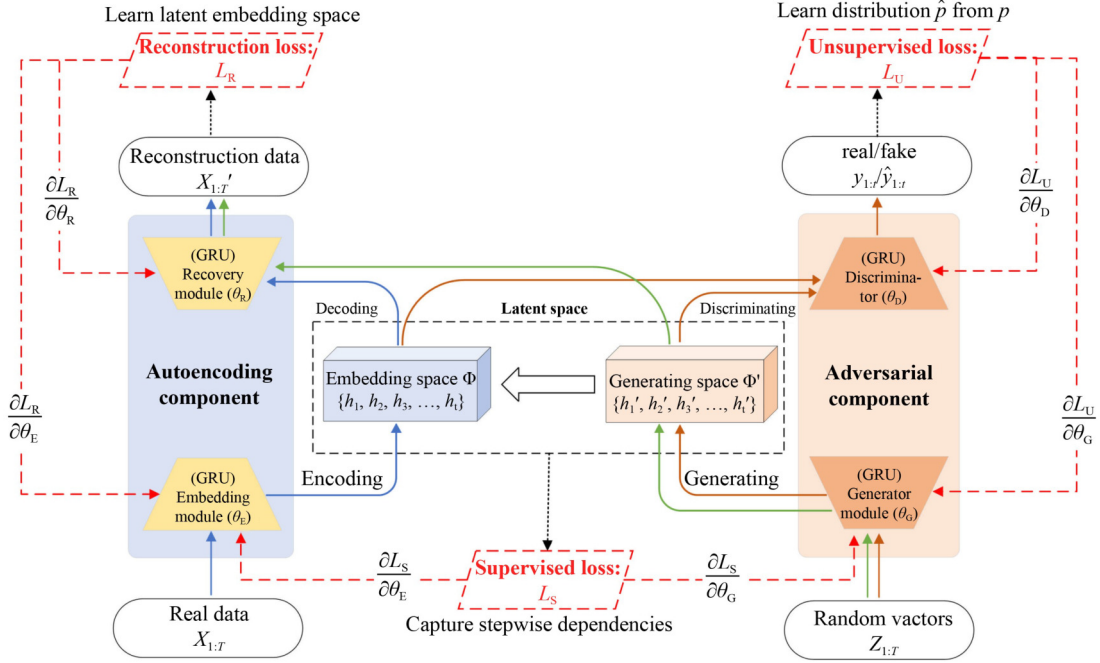


Fig. 2 TimeGAN network architecture.

series data, capturing both temporal and structural patterns within the data. The Adversarial Component introduces a GAN framework to enhance the quality and realism of the generated time series data. In Fig. 2, the blue and yellow solid lines represent the forward propagation of data, the green solid lines represent the data generation process, and the dashed lines represent the backward propagation of the gradient.

The overarching goal of the TimeGAN model is to learn a generative model capable of generating realistic synthetic time series data by capturing both the temporal dynamics and the inherent patterns of the real data, in order to generate large-scale, high-quality data and to enhance the data scale. It can be described as follows

$$\min_{\hat{p}} \{Distance(p(x_t|x_{1:t-1})||\hat{p}(\hat{x}_t|\hat{x}_{1:t-1}))\}. \quad (1)$$

In Eq. (1), p represents the distribution that real data follows, which is extracted from triaxial compression test of coarse-grained soils. \hat{p} represents the distribution that data generated by the TimeGAN model follows. x_t and \hat{x}_t represent data at time step t , obtained from real data set and generated data set, respectively. $p(x_t|x_{1:t-1})$ represents the conditional probability distribution of data x_t given the data from previous time steps $x_{1:t-1} = x_1, x_2, \dots, x_{t-1}$. This distribution reflects the temporal dependencies present in the real data samples. $\hat{p}(\hat{x}_t|\hat{x}_{1:t-1})$ represents the conditional probability distribution of data \hat{x}_t given the data from previous time steps $\hat{x}_{1:t-1} = \hat{x}_1, \hat{x}_2, \dots, \hat{x}_{t-1}$. $\hat{p}(\hat{x}_t|\hat{x}_{1:t-1})$ reflects the temporal dependencies in the generated data samples. *Distance* represents the distribution bias between the real data and the generated

data. By minimizing the bias between them, we can obtain generated data that closely approximates the distribution of real data.

2.1.1 Learning feature representation through the autoencoding component

In Fig. 2, the Autoencoding Component on the left comprises two modules: the Embedding Module f_E and the Recovery Module f_R . The f_E transforms the input time series into the latent representation, captures the key knowledge and temporal dynamics of the data. The f_R reconstructs the time series from the latent representation, ensuring that the reconstruction closely resembles the original input. Specifically, the input data $X_{1:T} = [x_1, x_2, x_3, \dots, x_t, \dots, x_T]$ is mapped to the latent embedding space Φ through the f_E , which is represented as $H_{1:T} = [h_1, h_2, h_3, \dots, h_t, \dots, h_T]$ in the latent space. The Recovery Module f_R generates the reconstructed data $X'_{1:T} = [x'_1, x'_2, x'_3, \dots, x'_t, \dots, x'_T]$ from $H_{1:T}$. $X'_{1:T}$ owns the same dimensions as the real data $X_{1:T}$.

To capture the temporal patterns and dependencies in the input time series data, the Embedding Module f_E and the Recovery Module f_R utilize the Gated Recurrent Unit (GRU) model. With GRU, we can preserve the long-term temporal dependencies existed in the input data samples. The network structure of the GRU is illustrated in Fig. 3.

As shown in Fig. 3, the GRU cell is a fundamental unit of the GRU model. Each cell receives the input x_t at time step t and the hidden state h_{t-1} from time step $t-1$, which contains the knowledge learned upto time step $t-1$. Then, the cell outputs the hidden state h_t at time step t . h_t also

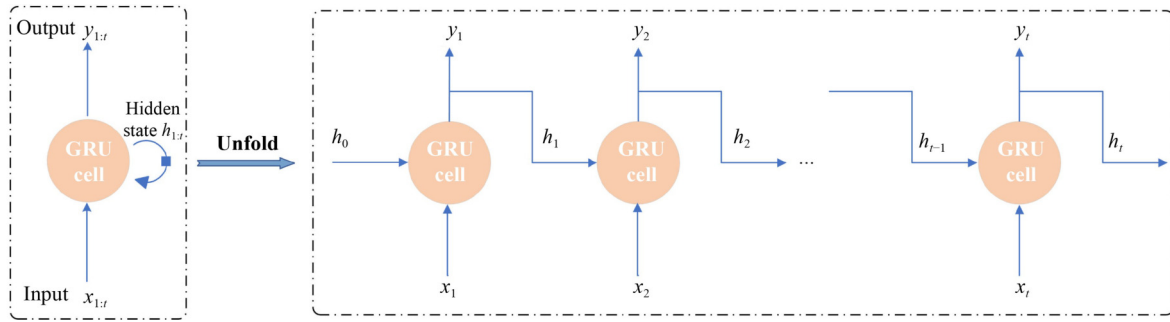


Fig. 3 The GRU network architecture.

represents the output y_t corresponding to input x_t at time step t , i.e., $h_t = y_t$. This process is described as follows

$$h_t = y_t = \text{GRU}(h_{t-1}, x_t). \quad (2)$$

Noting that the hidden state at time step $t-1$ also serves as the input for time step t .

The mappings of the Embedding Module f_E and the Recovery Module f_R are represented by the following equations.

$$h_t = f_E(h_{t-1}, x_t), \quad (3)$$

$$x'_t = f_R(x'_{t-1}, h_t). \quad (4)$$

2.1.2 Generating feature space approximating the embedding space via adversarial component

In Fig. 2, the Adversarial Component on the right comprises two modules: the Generator Module and the Discriminator Module (f_D). The Generator Module generates synthetic time series data from random noise, aiming to produce data that closely mimics the real time series data. The f_D evaluates and distinguishes between real and generated time series data. It provides feedback to the Generator Module by indicating how well the generated data matches the real data, thus driving improvements in the Generator Module's output. Specifically, the Generator Module includes the Generator Network (f_G) and the Supervisor Network (f_S). We configure the f_G , f_S and the f_D to utilize the GRU model. The input to f_G is a set of random noise vectors $Z_{1:T} = [z_1, z_2, z_3, \dots, z_t, \dots, z_T]$, typically sampled from a specific distribution (e.g., the Gaussian or Uniform distribution). f_G maps $Z_{1:T}$ to the latent generating space Φ' , which shares the same dimensionality as the embedding space Φ from the Autoencoding Component. f_G is represented by Eq. (5). The f_D receives latent representations from both space Φ and Φ' , and distinguishes the source of data samples, whether from a real data set or a generated data set. f_D is represented by Eq. (6).

$$h'_t = f_G(h'_{t-1}, z_t), \quad (5)$$

$$\tilde{u}_t = f_D(\tilde{u}_{t-1}, \tilde{h}_t), \tilde{y}_t = \text{Sigmoid}(\tilde{u}_t), \quad (6)$$

h'_{t-1} represents the hidden state of the f_G at time step $t-1$, z_t represents the random noise vector fed into f_G at time step t , and h'_t represents the latent representation generated by f_G at time step t . The f_D receives the latent representations \tilde{h}_t . \tilde{h}_t represents h_t from embedding space Φ or h'_t from generating space Φ' . \tilde{u}_{t-1} represents the hidden state of the f_D at time step $t-1$. \tilde{u}_t represents the output by f_D at time step t . By applying the Sigmoid activation function after the output \tilde{u}_t of the f_D , the discriminator outputs a probability value within the range of $[0, 1]$, represented by \tilde{y}_t . The \tilde{y}_t close to 1 indicates that the input latent representation is classified as originating from real data, expressed as y_t , while the \tilde{y}_t close to 0 signifies that it is classified as generated data, expressed as y'_t .

However, relying solely on the discriminator's binary adversarial feedback may not provide adequate motivation for the generator to accurately capture the stepwise conditional distributions within the data. To align the generated data more closely with the temporal dependencies of real data, we introduce the f_S . This network provides an additional supervisory mechanism. It guides the learning process and helps capture long-term temporal dependencies.

As shown in Fig. 4, the Supervisor Network takes the latent representations of real data h_{t-1} at the time step $t-1$ as input to predict next time step h_t , and the predicted value is \hat{h}_t . It is represented by Eq. (7).

$$\hat{h}_t = f_S(\hat{h}_{t-1}, h_{t-1}). \quad (7)$$

2.1.3 Joint training of autoencoding and adversarial components

In Fig. 2, TimeGAN utilizes three loss functions: the reconstruction loss L_R , the unsupervised adversarial loss L_U , and the supervised loss L_S .

The reconstruction loss L_R calculates the error between the real data $X_{1:T}$ and the reconstructed data $X'_{1:T}$ obtained through the Autoencoding Component. The mean squared

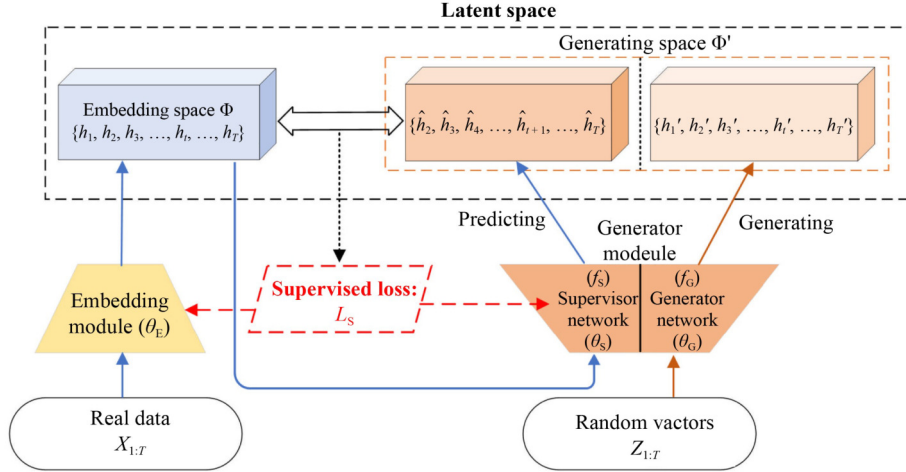


Fig. 4 Capturing the stepwise conditional distribution.

error (MSE) is used to compute L_R , which is then used to train the Embedding Module and Recovery Module, updating the parameters θ_E and θ_R of both modules, as described in Eq. (8).

$$L_R = E_{X_{1:T} \sim p} \sum_t \|x_t - x'_t\|_2, \quad (8)$$

where p represents the data distribution that $X_{1:T}$ follows, $\sum_t \|x_t - x'_t\|_2$ indicates the sum of squared Euclidean distances between the real data sample and the reconstructed data sample, and $E_{X_{1:T} \sim p}$ denotes the expectation.

The unsupervised adversarial loss L_U quantifies the classification error between the representations of real data $H_{1:T}$ and the generated data $H'_{1:T}$ in latent space. L_U employs the binary cross-entropy loss to train the Adversarial Component, updating the parameters θ_G and θ_D of f_G and f_D , as stated in Eq. (9).

$$\begin{aligned} L_U &= E_{X_{1:T} \sim p} \sum_t \log y_t + E_{X'_{1:T} \sim \hat{p}} \sum_t \log(1 - y'_t) \\ &= E_{X_{1:T} \sim p} \log D(x_t) + E_{X'_{1:T} \sim \hat{p}} \log(1 - D(G(z_t))). \end{aligned} \quad (9)$$

The supervised loss L_S computes the discrepancy between predicted \hat{h}_t at time step t and actual value h_t to guide the Supervisor Network. MSE is used to compute L_S , which is then used to train the Supervisor Network and Embedding Module, updating the parameters θ_S and θ_E , as described in Eq. (10).

$$L_S = E_{X_{1:T} \sim p} \sum_t \|h_t - \hat{h}_t\|_2. \quad (10)$$

2.2 Training deep learning models with pre-training

A large amount of augmented data can be generated using real coarse-grained soil data through data augmentation. At this stage, we implement a pre-training strategy to train deep learning models on both augmented and real data. Although the augmented data samples are produced

through adversarial training and generally follows the pattern of the real data, it is not an exact replication. Instead, the augmented data exhibits diversity. During pre-training, we can apply a strategy of parameter freezing or overall parameter fine-tuning for pre-trained models. In this paper, we aim for the pre-training strategy to mitigate systematic errors caused by data bias in the augmented data. We hope that the model can better adapt to the data characteristics of the target task while retaining the general knowledge embedded in the augmented data. Therefore, the parameter tuning strategy we adopt in this paper is overall fine-tuning.

As shown in Fig. 5, the augmented data D_S is included in the source domain, while the real data D_T is included in target domain. The deep learning models are initially trained on the source domain with randomly initialized weights. In the pre-training process, the model learns general knowledge from the large augmented data set D_S in the source domain to capture general patterns, features, and relationships. The resulting pre-trained model f_{θ_s} , with parameters denoted as θ_s , is then refined and adjusted through fine-tuning on the target domain real data D_T . Fine-tuning adjusts the parameters of the pre-trained model to better fit the nuances and specifics of the target domain. After that, the target model f_{θ_t} is obtained, where the final model parameters are denoted as θ_t .

3 Experimental results and analysis

3.1 Data collection and feature selection

In this study, we collected real data from triaxial compression experiments reported in 56 relevant literatures from China, the USA, and Japan. For the specific sources, please refer to the Electronic Supplementary Material. Generally, to gather the experimental data, coarse-grained soil specimens are first prepared and then tested using triaxial compression

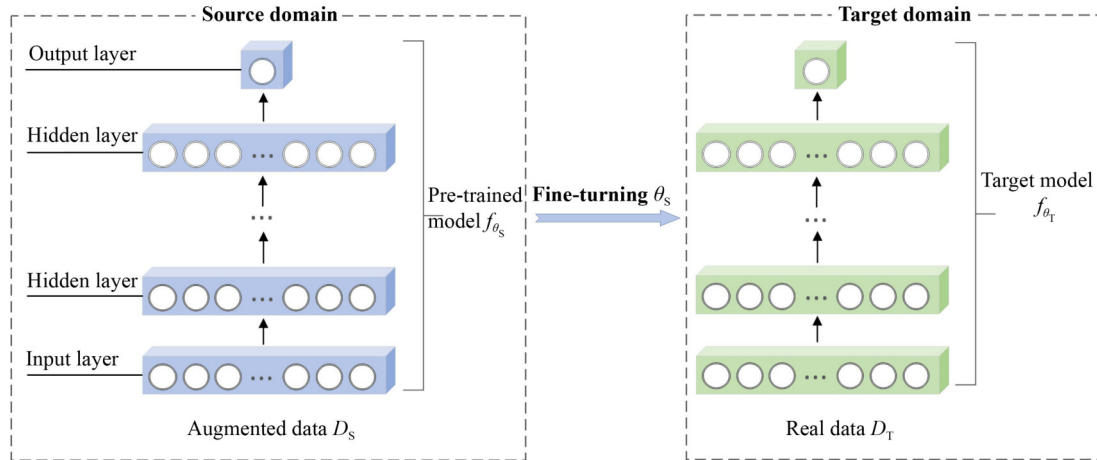


Fig. 5 Pre-training and fine-tuning process of the model.

equipment under laboratory conditions. Changes in the specimen are measured using transducers in the test setup, and these changes are continuously recorded until the predetermined stress or strain conditions are reached, or the specimen fails. The recorded data represents the actual measurements obtained from the experiments documented in these literatures.

In all the literature, although the soils are not identical, they are all coarse-grained soils with more than 50% of the total soil mass consisting of particles larger than 0.075 mm. Additionally, these soils exhibit a similar particle size distribution and gradation, leading to similar physical properties. There are only a few data sets available for coarse-grained soils with a maximum particle size greater than 60 mm. These soils also exhibit significant differences in physical properties compared to those with a maximum particle size of 60 mm. As a result, the larger particle size soil samples (80, 160, 200 mm) were not included in this study.

We collected a total of 335 sets of ε_a – ε_v curves (with a maximum particle size of 60 mm or less) from the literature. These curves served as experimental samples to construct a database for developing deep learning-based prediction models. In civil engineering, triaxial compression experiments are essential for studying the deformation characteristics of coarse-grained soils. The data set used in this paper consists of results from triaxial compression experiments found in the literature. Each curve collected typically describes the deformation behavior of soils under stress.

In triaxial compression experiments on coarse-grained soils, the deformation behavior of the soil sample is directly influenced by its physical properties, such as particle size distribution, dry density and so on. Additionally, the confining pressure and stress applied to the soil samples simulate real-world environmental conditions and also affect its deformation behavior. Therefore, we select particle size distribution, dry density, confining pressure and others as input features, with the

soil's deformation behavior (volumetric strain) as the target value for prediction, as detailed in Table 1.

Table 1 The input features and model output

Categorie	Symbol	Measurement unit	Description
Input feature	d_{\max}	mm	maximum particle size
	d	mm	container diameter
	h	mm	container height
	p	kPa	confining pressure
	ρ_d	g/cm ³	dry density
	e	%	void ratio
	PSD	%	particle size distribution
Output	ε_a	%	axial strain
	ε_v	%	volumetric strain

In Table 1, the feature particle size distribution (PSD) refers to the particle size distribution of coarse-grained soil. It describes the distribution of grading curves for coarse-grained soil. Specifically, PSD is a key feature affecting the deformation characteristics of coarse-grained soil. It can be discretized for particle sizes, including 5, 10, 20, 40, and 60 mm. Each particle size is denoted as P_i , representing the proportion of particles smaller than i mm. PSD consists of five features: $P_5, P_{10}, P_{20}, P_{40}$, and P_{60} . In this paper, we uniformly use PSD to represent the gradation distribution feature.

In the collected data set, some features had incomplete information and missing values. To address this issue, we used empirical formulas from geotechnical studies. These formulas were applied to estimate the missing values based on the relevant features in the collected data. In addition, in triaxial compression tests, the volumetric strain of coarse-grained soils is initially measured as a continuous variable. We discretized this variable for analysis. Specifically, volumetric strain values are sampled at intervals of 0.3% of axial strain, up to a maximum axial strain of 15%. As a result, each curve in

the data set comprises 50 discrete points, as shown in Fig. 6. Since these data are obtained from real experimental results and the discretized points show no abrupt jumps or inconsistencies, the collected data samples are coherent and accurate enough.

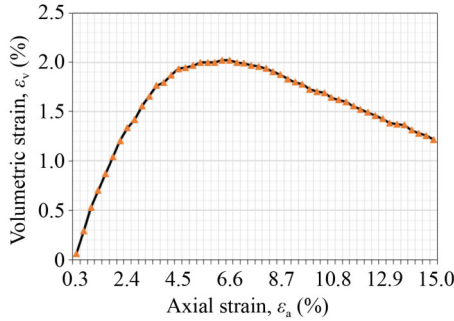


Fig. 6 Discretized ε_a - ε_v curve of coarse-grained soils.

3.2 Model settings and evaluation metrics

We use 80% of the collected data for training and 20% for testing. Specifically, 268 curves were designated for the training set, comprising a total of 13400 points. The remaining 67 curves were allocated to the test set, totaling 3350 points. The divided training set is used to train the TimeGAN model to generate augmented data. Then, this augmented data serves as the source domain data set for model pre-training. During the pre-training process, we used 268 curves as the target data set. This data set was employed for fine-tuning the model. To prevent rapid overwriting of the pre-trained weights, we set a lower learning rate than that used for the pre-trained model. Finally, the test set is used to evaluate the performance of the deep learning-based models.

We used four deep learning models for training on the training set and testing on the test set. These models served as baseline models. We performed a Grid Search to optimize the parameters of the four models. The final parameters for the baseline models are shown in Table 2.

In this study, mean absolute error (MAE) and the coefficient of determination (R^2) are set as evaluation metrics. MAE measures the average magnitude of errors in a set of predictions. A smaller MAE value indicates better model performance. R^2 is a statistical measure that assesses how well a regression model explains and predicts the outcome variable. It takes values in the range [0,1]. A larger R^2 value indicates better model performance. The calculation equations for both evaluation metrics are provided below.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (11)$$

$$R^2 = \frac{1}{N/n} \sum \left(1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \right), \quad (12)$$

where N denotes the total number of data samples, n denotes the number of samples per curve, which is equal to 50. This represents the number of points obtained after discretizing each axial strain–volumetric strain curve. y_i is the actual value of the i th observation, \hat{y}_i is its predicted value, and \bar{y} signifies the mean of the actual values for each curve. R^2 indicates the degree of fit between the true and predicted curves. Each curve is calculated separately. The results are then summed and averaged to obtain the R^2 value for the entire test set.

3.3 Evaluation of the proposed approach

3.3.1 Evaluating the performance of the proposed data augmentation method

To evaluate the quality of augmented data, feature importance analysis is employed. This analysis assesses the distribution of feature importance and examines whether the augmented data replicates the feature relationships observed in real data. Additionally, it checks if the features consistently contribute to target decisions, specifically the mechanical behavior of volumetric strain. This approach incorporates insights from both deep learning and civil engineering. The results of the feature importance analysis guide adjustments to the TimeGAN model. These results inform the necessary improvements. The final parameters for the TimeGAN model are listed in Table 3.

During the training process of the TimeGAN model, updating the generator and discriminator with the same frequency can often lead to one component overpowering the other, resulting in training instability or mode collapse. To maintain a balance between the generator and discriminator, it is typical to increase the parameter update frequency for the generator. In this study, the generator's parameter update frequency is set to be twice that of the discriminator. Additionally, there are a few input features for the study of deformation characteristics of coarse-grained soils. So, we increase the dimensionality of the latent space to achieve more detailed and comprehensive feature representations, as shown in the table above.

1) Introduction to feature importance analysis

Two common methods are used to explain the decision-making process of machine learning models: Local Interpretable Model-Agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP).

The interpretation process of LIME involves selecting a

Table 2 Architectures and hyper-parameters of four baseline models

Model	Name of architecture and hyper-parameter	Value
Deep neural network (DNN)	number of network layers	5
	number of neurons per layer	128, 256, 64, 32, 1
	learning rate	0.001
	batch size	200
	activation function	ReLU
	optimizer	Adam
Convolutional neural network (CNN)	number of network layers	5
	number of convolution kernels per layer	64, 128, 64
	convolutional layer	Conv1D
	convolution kernel size	5
	learning rate	0.001
	batch size	200
	activation function	ReLU
	optimizer	Adamax
Residual network (ResNet)	number of residual blocks	5
	number of neurons per layer	128, 256, 64, 32, 1
	learning rate	0.001
	batch size	200
	activation function	ReLU
	optimizer	Adam
Autoencoder (AE)	number of encoder network layers	5
	number of decoder network layers	5
	number of neurons per layer of the encoder	128, 256, 64, 32, 12
	number of neurons per layer of the decoder	32, 64, 256, 128, 12
	learning rate	0.001
	batch size	200
	activation function	ReLU
	optimizer	Adam

Table 3 Hyper-parameters of the TimeGAN

No.	Name	Value
1	sequence steps	5
2	latent space dimension	64
3	input dimensions of random vectors	128
4	learning rate	0.0005
5	batch size	32
6	activation function	ReLU
7	optimizer	Adam

specific instance from the data set to interpret the prediction of a complex machine learning model. Perturbations are generated around this instance by creating similar samples, which are then used to predict outcomes using the black-box model. Next, a simpler, interpretable model (such as linear regression) is trained

on these perturbed instances, aiming to approximate the behavior of the black-box model locally. Feature importance is assessed through the coefficients of the interpretable model, providing insights into which features significantly influence the prediction for the chosen instance. This approach enables LIME to offer explanations for individual predictions. However, LIME does not provide global explanations that cover the entire model's behavior [28]. SHAP [29] is employed to calculate the Shapley values, which quantify the contribution of each feature to the prediction across all possible combinations of features. These values are then integrated to determine the overall impact of each feature on the model's output for a specific instance. Subsequently, these impacts are summarized to provide a clear and comprehensive explanation of how each feature affects the output. The explanations are presented in a way that allows for both local interpretations of

individual predictions and global insights into the model's behavior across the entire data set. SHAP can explain the overall behavior of the model and quantify each feature's contribution to the outputs. In this paper, we employed SHAP to analyze feature importance.

SHAP is a unified approach to explain the output of any machine learning model. It is based on Shapley values from cooperative game theory, which fairly distribute the "payout" among the players based on their contribution. The equation for calculating the Shapley value is shown below

$$\Phi_i(f) = \sum_{S \subseteq U \setminus \{i\}} \frac{|S|! * (|U| - |S| - 1)!}{|U|!} [f(S \cup \{i\}) - f(S)]. \quad (13)$$

In Eq. (13), $\Phi_i(f)$ represents the contribution, or the importance of feature i in the model f . U represents the set of all features, and $|U|$ denotes the number of features. S is a subset of U excluding feature i , and $|S|$ is the number of features in S . The term $f(S)$ represents the output of the model with S as input. The expression $f(S \cup \{i\}) - f(S)$ indicates the marginal contribution of feature i . The coefficient $\frac{|S|! * (|U| - |S| - 1)!}{|U|!}$ represents the weight of the marginal contribution of feature i .

2) Evaluating TimeGAN using feature importance analysis

We use SHAP to get the feature importance rank for each baseline model. The results are shown in Fig. 7.

In Fig. 7, the y -axis represents the features of coarse-grained soils, and the x -axis represents the importance of these features. As depicted in Fig. 7, it can be observed that the feature importance distributions of the real data are similar to those of the augmented data. In the DNN, CNN, and ResNet baseline models, the importance ranking of the corresponding features is entirely consistent, with only slight variations in the contribution value of each feature to the model outputs. In the AE baseline model, the top six ranked features maintain their order in terms of contribution, with slight changes in their contribution values. The order of the last two features has changed, but their importance is very low, resulting in a minimal impact on the model. This indicates that there are biases between the augmented data and the real data, but the biases remain within an acceptable range.

Upon a comprehensive review of Fig. 7, it is evident that the ranking of feature importance in the augmented data aligns with that in the real data, indicating that the augmented data closely mirrors the real data. Furthermore, employing SHAP to analyze the feature importance across different models reveals that PSD has the most significant impact on the deformation behavior of coarse-grained soils. Confining pressure (p), dry density (ρ_d) and void ratio (e) also have critical effects on deformation characteristics, while the impact of

maximum particle size (d_{\max}), container diameter (d) and height (h) is relatively minor. There is still a minor effect of the container on the mechanical behavior of coarse-grained soils. But this effect is often overlooked in soil mechanics research. At the same time, d_{\max} has a minimal effect on the deformation behavior of coarse-grained soils. This is because nearly all values of d_{\max} in the collected data are 60 mm. Only a few values are less than 60 mm. The results of the feature importance analysis are consistent with the findings of literature on the mechanical properties of coarse-grained soils [30–32]. These findings indicate that the mechanical behavior of coarse-grained soils is governed by inherent particle properties, with density and confining pressure being important factors affecting the mechanical properties.

3.3.2 Evaluating the performance of models implementing pre-training strategies

Although the feature importance distributions of real data and augmented data are consistent, combining them as a training set for deep learning-based models does not yield optimal results, as shown in Table 4. Table 4 provides the performance of baseline models with various training sets.

The results indicate that training with real data consistently yields the best prediction performance. At once, there is a downward trend in the prediction performance of models when using augmented data alone or when combining augmented and real data for training. It shows that there are biases between the real and augmented data as shown in Fig. 7. And these biases adversely affect the performance of the model. To overcome this challenge, we introduce the pre-training strategy. The randomly initialized model is pre-trained using augmented data to acquire general knowledge and then fine-tuned with real data. The results are shown in Table 5.

The results in Table 5 indicate that the introduction of the pre-training strategy significantly improves the models' performance. In four models, results with pre-training consistently outperform those without pre-training. Particularly, the ResNet model shows the most significant improvement, with a 0.03 decrease in MAE , representing an 11.75% decrease, and an increase in R^2 of 0.033. Among all models, the AE model performs the best, with the MAE of 0.2219 and the R^2 of 0.9155. The improvement demonstrates the feasibility and rationality of our proposed method, which is a new approach for studying the deformation characteristics of coarse-grained soils. Moreover, pre-training can also bring the distribution of feature importance of the combined real and augmented data closer to that of the real data across all four baseline models, as shown in Fig. 8.

In Fig. 8, the y -axis represents the features influencing

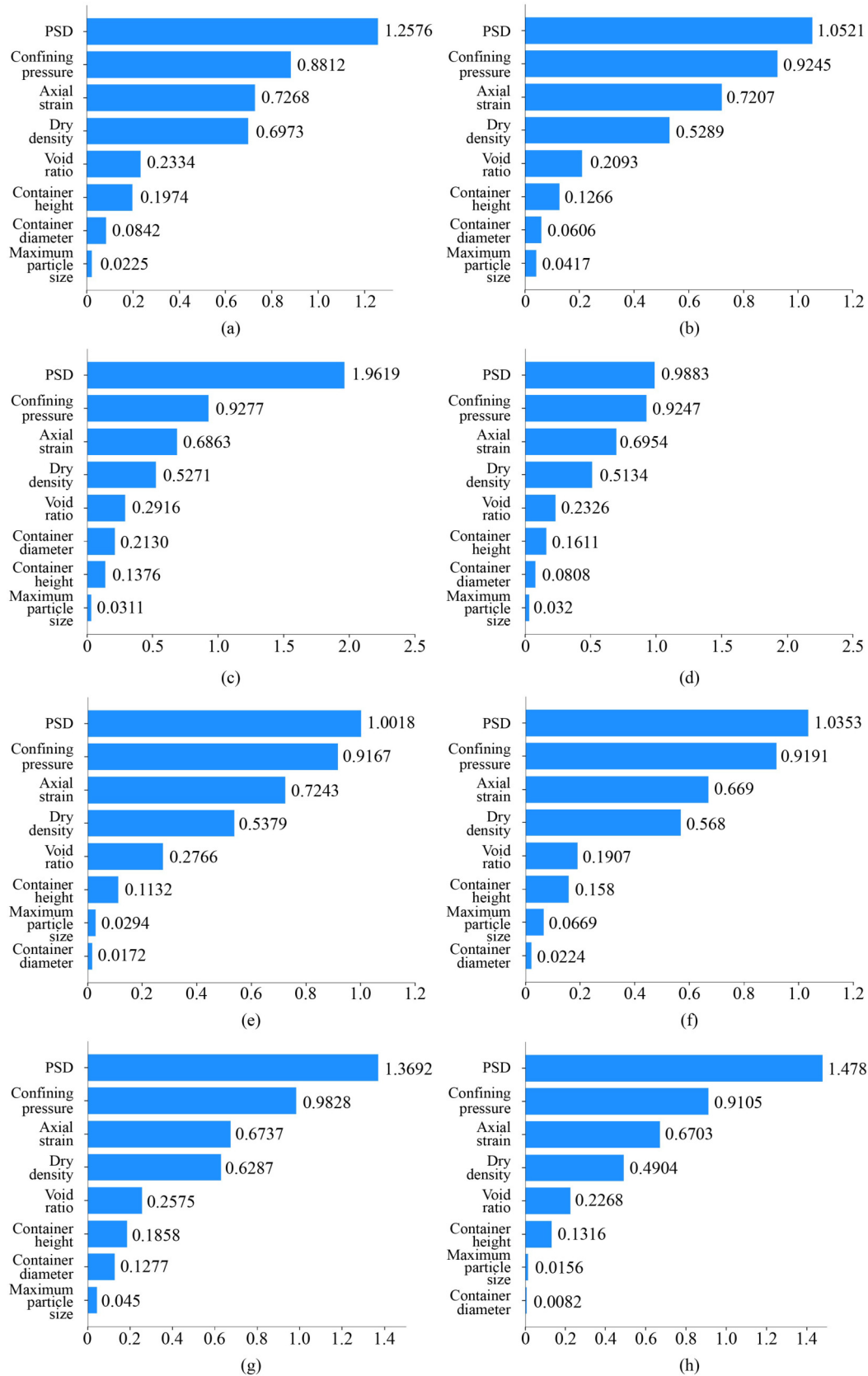


Fig. 7 Feature importance analysis using SHAP: (a) DNN (training with real data); (b) DNN (training with augmented data); (c) CNN (training with real data); (d) CNN (training with augmented data); (e) ResNet (training with real data); (f) ResNet (training with augmented data); (g) AE (training with real data); (h) AE (training with augmented data).

the deformation characteristics of coarse-grained soils, and the x -axis represents the importance of each feature, i.e., the contribution value. Additionally, the blue bar represents the feature importance distribution of baseline models trained with real data, the orange bar represents the feature importance distribution of models trained directly with combined real and augmented data, and the

Table 4 Performance of baseline models with various training sets

Model	Real data		Augmented data		Real and augmented data	
	MAE	R^2	MAE	R^2	MAE	R^2
DNN	0.2718	0.8520	0.4121	0.2778	0.2901	0.7651
CNN	0.2582	0.8630	0.3891	0.2116	0.2792	0.8388
ResNet	0.2569	0.8682	0.4038	0.2881	0.2928	0.8308
AE	0.2470	0.8747	0.4161	0.2430	0.2880	0.8199

Table 5 Model evaluation results

Model	Without pre-training (real data)		With pre-training (real and augmented data)	
	MAE	R^2	MAE	R^2
DNN	0.2718	0.8520	0.2518	0.8802
CNN	0.2582	0.8630	0.2376	0.9007
ResNet	0.2569	0.8682	0.2267	0.9008
AE	0.2470	0.8747	0.2219	0.9155

green bar represents the feature importance distribution of pre-trained and fine-tuned models trained with combined real and augmented data using the pre-training strategy.

As shown in Fig. 8, we can observe that in four models, the feature importance distribution from the pre-trained and fine-tuned models is closer to that from the baseline models without pre-training. Specifically, for the two most significant features, PSD and Confining Pressure, the contribution values in the pre-trained and fine-tuned models, though not identical to those in the baseline models, are noticeably closer. For the remaining features, the fine-tuned models show a trend of feature importance that more closely aligns with the baseline models. It is evident that introducing pre-training does not change the ranking of feature importance but alleviated the negative effects of biases between augmented data and real data in the model training process.

The true and predicted $\varepsilon_a-\varepsilon_v$ curves for coarse-grained soils are compared, with the results presented below. In Fig. 9, the black curve represents the results obtained from the triaxial compression test, regarded as ground truth values; the red curve represents the predicted values from the Baseline model; the green curve represents the predicted values from the proposed model with data augmentation and the pre-training strategy. We can see that, compared to the baseline model, the proposed method achieves significantly closer alignment with the

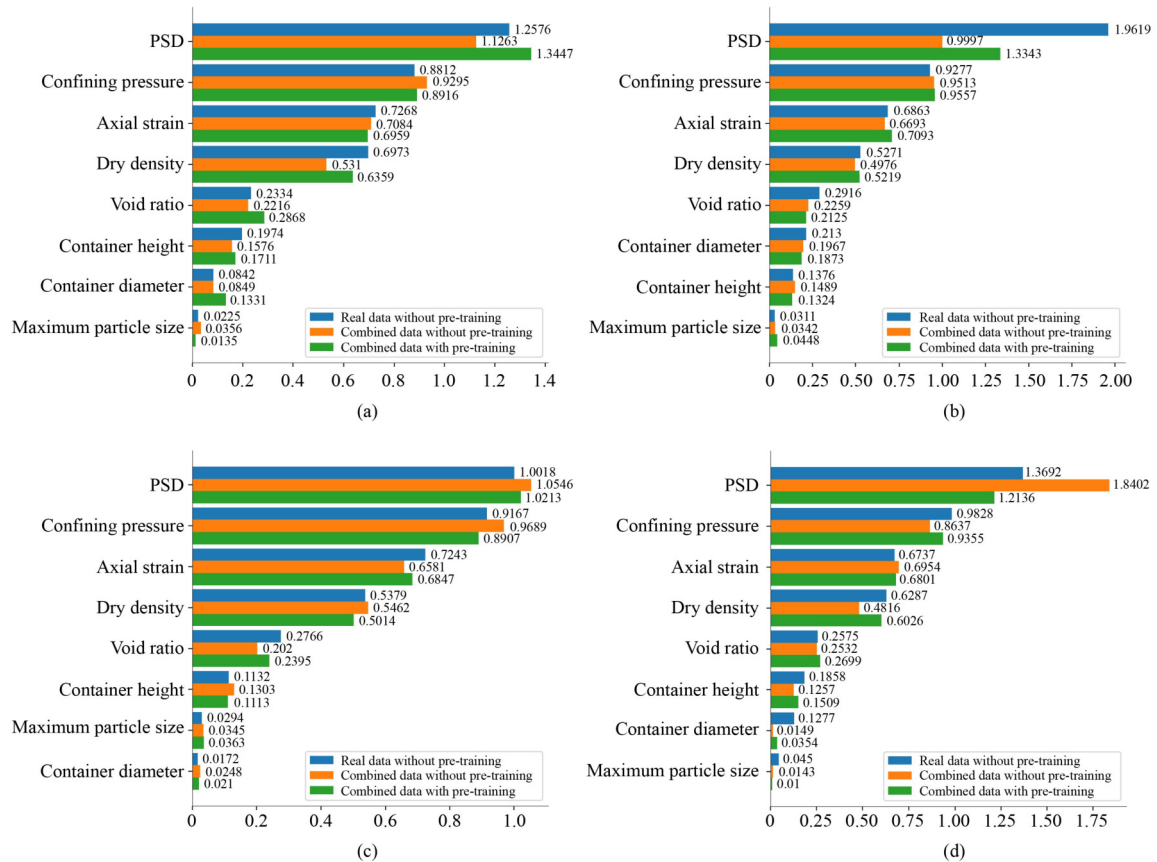


Fig. 8 Feature importance distribution with/without pre-training strategy: (a) DNN; (b) CNN; (c) ResNet; (d) AE.

ground truth values, demonstrating higher fidelity in fitting the real curve.

For further evaluation of the proposed method, scatter plots are plotted to compare the true values of the entire test set with the model's predicted values. As shown in Fig. 10, the x -axis represents the true values, and the y -axis represents the predicted values. The black diagonal line indicates perfect prediction accuracy, where the predicted values equal the actual values. Blue scatter points denote the predicted values from the baseline model, and red scatter points denote the predicted values from the proposed model. It is worth noting that the closer the scatter points are to the black diagonal line, the more accurate the predictions. It is evident that the red scatter points are more closely distributed around the diagonal compared to the blue scatter points, indicating that the proposed method's prediction results are closer to the true values and demonstrating higher model accuracy.

4 Conclusions and future work

In this paper, we proposed a deep learning-based method for predicting the deformation behavior of coarse-grained soils. The method aimed to explore the relationship between the deformation behavior and the physical properties of coarse-grained soils. It focused on obtaining the axial strain–volumetric strain curve for these soils.

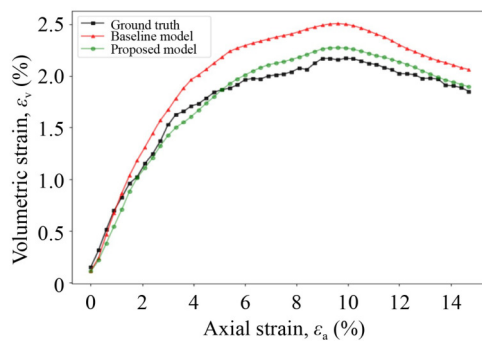


Fig. 9 Comparison of the predicted ε_a – ε_v curves.

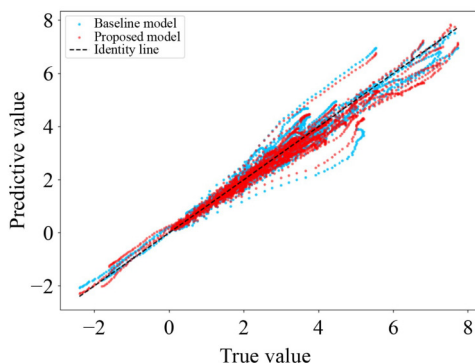


Fig. 10 The scatter plots of predicted volumetric strain values.

Once trained, the model could be applied to various soil types and conditions. This approach enabled the rapid generation of axial strain–volumetric strain curves, significantly reducing both time and economic costs associated with physical experiments. The use of these prediction curves allows us to quickly derive mechanical parameters for soils in various environments, such as c , ϕ and E . This capability aids in exploring the mechanical properties of different soils. It also contributes to the optimization of engineering design and improves decision-making processes.

We employed the TimeGAN model for data augmentation to expand the data set and address the common issue of insufficient training data in deep learning-based methods. Additionally, we utilized the SHAP method to assess the quality of generated data and guide the training of TimeGAN to address the challenge of evaluating training progress in traditional GAN models when generating numerical data. The experimental results indicated that the data generated through TimeGAN learned the feature importance distribution of real data. To mitigate the detrimental impact of biases between augmented and real data, we introduced the pre-training strategy. The experimental findings demonstrated that this strategy aligns the feature importance distribution of the combined real and augmented data more closely with that of the real data, thereby improving the model's performance in studying the deformation characteristics of coarse-grained soils. At the same time, the results of the proposed method aligned with findings in geotechnical engineering research and hold practical value for addressing real-world issues in civil engineering.

The future work is to introduce domain knowledge and prior information related to the mechanical properties of coarse-grained soils and embed the information into the model training process. This will improve the model's ability to solve practical problems and enhance its interpretability. Besides, a major challenge is the extreme scarcity of experimental data sets for soils with large grain sizes. To address this, we plan to use deep learning methods, such as transfer learning, to conduct an in-depth study of the mechanical properties of large-grain coarse-grained soils.

Electronic supplementary material Supplementary material is available in the online version of this article at <https://doi.org/10.1007/s11709-025-1161-z> and is accessible for authorized users.

Acknowledgements This work was jointly supported by the National Natural Science Foundation of China (Grant No. 52078212), the State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, and China Institute of Water Resources and Hydropower Research, China (No. IWHR-SKL-202003).

Competing interests The authors declare that they have no competing interests.

References

1. He Z M, Xiang D, Liu Y X, Gao Q F, Bian H B. Deformation behavior of coarse-grained soil as an embankment filler under cyclic loading. *Advances in Civil Engineering*, 2020, 2020(1): 4629105
2. Zhang X T, Gao Y Z, Wang Y, Yu Y Z, Sun X. Experimental study on compaction-induced anisotropic mechanical property of rockfill material. *Frontiers of Structural and Civil Engineering*, 2021, 15(1): 109–123
3. Liu X Y, Zou D G, Liu J M, Zheng B W. Predicting the small strain shear modulus of coarse-grained soils. *Soil Dynamics and Earthquake Engineering*, 2021, 141: 106468
4. Chen J H, Zhang Y J, Yang Y X, Yang B, Huang B C, Ji X P. Influence of coarse grain content on the mechanical properties of red sandstone soil. *Sustainability*, 2023, 15(4): 3117
5. Li S Y, Wang T C, Wang H, Jiang M J, Zhu J A. Experimental studies of scale effect on the shear strength of coarse-grained soil. *Applied Sciences—Basel*, 2022, 12(1): 447
6. Le Q H, Nguyen D H, Sang-To T, Khatir S, Le-Minh H, Gandomi A H, Cuong-Le T. Machine learning based models for predicting compressive strength of geopolymer concrete. *Frontiers of Structural and Civil Engineering*, 2024, 18(7): 1028–1049
7. Khuntia S, Mujtaba H, Patra C, Farooq K, Sivakugan N, Das B M. Prediction of compaction parameters of coarse grained soil using multivariate adaptive regression splines (MARS). *International Journal of Geotechnical Engineering*, 2015, 9(1): 79–88
8. Ikeagwuani C C, Nwonu D C. Statistical analysis and prediction of spatial resilient modulus of coarse-grained soils for pavement subbase and base layers using MLR, ANN and Ensemble techniques. *Innovative Infrastructure Solutions*, 2022, 7(4): 273
9. Verma G, Kumar B. Multi-layer perceptron (MLP) neural network for predicting the modified compaction parameters of coarse-grained and fine-grained soils. *Innovative Infrastructure Solutions*, 2022, 7(1): 78
10. Anitescu C, Atroshchenko E, Alajlan N, Rabczuk T. Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials & Continua*, 2019, 59(1): 345–359
11. Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh V M, Guo H, Hamdia K, Zhuang X, Rabczuk T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 2020, 362: 112790
12. Shrestha A, Mahmood A. Review of deep learning algorithms and architectures. *IEEE Access: Practical Innovations, Open Solutions*, 2019, 7: 53040–53065
13. Qu T M, di S C, Feng Y T, Wang M, Zhao T T, Wang M Q. Deep learning predicts stress–strain relations of granular materials based on triaxial testing data. *CMES—Computer Modeling in Engineering and Sciences*, 2021, 128(1): 129–144
14. Shorten C, Khoshgoftaar T M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019, 6(1): 60
15. Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 2017, 60(6): 84–90
16. Islam Z, Abdel-Aty M, Cai Q, Yuan J H. Crash data augmentation using variational autoencoder. *Accident Analysis and Prevention*, 2021, 151: 105950
17. Ohno H. Auto-encoder-based generative models for data augmentation on regression problems. *Soft Computing*, 2020, 24(11): 7999–8009
18. Delgado J M D, Oyedele L. Deep learning with small datasets: using autoencoders to address limited datasets in construction management. *Applied Soft Computing*, 2021, 112: 107836
19. Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence N D, Weinberger K Q, eds. *Advances in Neural Information Processing Systems 27 (NIPS 2014)*. Quebec: Institute of Electrical and Electronics Engineers (IEEE), 2014, 2672–2680
20. Jiang X, Ge Z. RAGAN: Regression attention generative adversarial networks. *IEEE Transactions on Artificial Intelligence*, 2023, 4(6): 1549–1563
21. Ma Z R, Wang J J, Feng Y S, Wang R K, Zhao Z H, Chen H W. Hydrogen yield prediction for supercritical water gasification based on generative adversarial network data augmentation. *Applied Energy*, 2023, 336: 120814
22. Yoon J, Jarrett D, van der Schaar M. Time-series generative adversarial networks. In: Wallach H, Larochelle H, Beygelzimer A, d’Alche-Buc F, Fox E, Garnett R, eds. *Advances in Neural Information Processing Systems 32 (NIPS 2019)*. Vancouver: Neural Information Processing Systems Foundation (NeurIPS), 2019: 5508–5518
23. Zhang Y F, Zhou Z H, Liu J W, Yuan J J. Data augmentation for improving heating load prediction of heating substation based on TimeGAN. *Energy*, 2022, 260: 124919
24. Wang Y, Song M, Jia M, Shi L, Li B. TimeGAN based distributionally robust optimization for biomass-photovoltaic-hydrogen scheduling under source-load-market uncertainties. *Energy*, 2023, 284: 128589
25. Liang T, Wang F L, Wang S, Li K, Mo X L, Lu D. Machinery Health Prognostic with Uncertainty for Mineral Processing using TSC-TimeGAN. *Reliability Engineering & System Safety*, 2024, 246: 110055
26. Wang Y, Zhang J, Wang Y. Do generated data always help contrastive learning? 2024, arXiv:2403.12448
27. Hendrycks D, Lee K, Mazeika M. Using pre-training can improve model robustness and uncertainty. In: Chaudhuri K, Salakhutdinov R, eds. *Proceedings of the 36th International Conference on Machine Learning*. Long Beach, CA: Association for the Advancement of Artificial Intelligence, 2019, 2712–2721
28. Zafar M R, Khan N. Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, 2021, 3(3): 525–541
29. Lundberg S M, Lee S I. A unified approach to interpreting model predictions. In: Guyon I, Luxburg U V, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. Long Beach,

- CA: Neural Information Processing Systems Foundation (NeurIPS), 2017: 1705.07874
30. Meng F, Zhang J S, Chen X B, Wang Q Y. Deformation characteristics of coarse-grained soil with various gradations. *Journal of Central South University*, 2014, 21(6): 2469–2476
 31. Ahmed S S, Martinez A, DeJong J T. Effect of gradation on the strength and stress-dilation behavior of coarse-grained soils in drained and undrained triaxial compression. *Journal of Geotechnical and Geoenvironmental Engineering*, 2023, 149(5): 04023019
 32. Jiang J S, Liu H L, Cheng Z L, Ding H, Zou Y Z. Influences of density and confining pressure on mechanical properties for coarse-grained soils. *Journal of Changjiang River Scientific Research*, 2009, 26(8): 46–50 (in Chinese)