International Journal of Geosynthetics and Ground Engineering (2024) 10:16
https://doi.org/10.1007/s40891-024-00533-7

ORIGINAL PAPER

Modeling the Mechanical Response of Cement-Admixed Clay Under Different Stress Paths Using Recurrent Neural Networks

Chana Phutthananon[1] · Praiya Ratanakijkul[1] · Sompote Youwai[1] · Warat Kongkitkul[1] · Pornkasem Jongpradist[1]

Received: 11 November 2023 / Accepted: 24 February 2024 / Published online: 9 March 2024
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2024

Abstract

Cement–admixed clay (CAC) is a widely-used soil stabilization technique for enhancing the strength and stiffness of soft clay. However, the stress–strain behavior of CAC is complex and nonlinear, and also depends on various factors such as mixing proportion, confining pressure, stress path, and shearing condition. In this study, we propose a novel approach for modeling the stress–strain behavior of CAC using recurrent neural networks (RNNs), which are a type of deep learning (DL) technique that can well capture the temporal dependencies and nonlinearities in sequential data. We compare three types of RNNs: traditional RNN, long short-term memory (LSTM) neural network, and gated recurrent unit (GRU) neural network, and evaluate their performance in simulating the strain- and stress-controlled triaxial test results of 25 CAC specimens with different mixing proportions and confining pressures. The results demonstrate that the LSTM model, incorporating a 2-time step backward, exhibits superior prediction accuracy and generalization capability compared to other evaluated models, achieving a mean absolute percentage error (MAPE) of 4%. This LSTM model is capable of capturing the stress–strain behavior of CACs across various loading conditions and mixing proportions within a unified framework. Therefore, we suggest that the LSTM model is a promising tool for modeling and analyzing the mechanical behavior of CAC in geotechnical engineering applications.

Keywords Deep learning · Long short-term memory · Prediction · Cement–admixed clay · Cemented clay stress–strain behavior

Introduction

Cement–admixed clay (CAC) has garnered significant attention as a soil stabilization technique in various geotechnical projects aimed at improving the mechanical properties of foundation soils to meet design requirements. It has been successfully employed in applications such as fill materials for land reclamations (e.g., [1, 2]), ground improvements for column-supported embankments (e.g., [3–7]), and retaining structures for excavations and slopes (e.g., [8–10]). One notable application is the soil–cement column, which utilizes CAC as a stronger component to enhance the stiffness and strength of weak foundation soils before bearing any structural load, i.e., vertical and lateral loads. This approach has demonstrated its effectiveness in minimizing displacement and enhancing the stability of geotechnical structures (e.g., [3–5, 9, 11–13]). Therefore, to obtain the realistic behavior of geotechnical structures when utilizing CAC as a reinforcement component, it is imperative to appropriately and precisely model and analyze the stress–strain characteristics of CAC.

Referring to the previous studies, the behaviors of CAC are often described using clayey soil mechanics-based constitutive models relying on the elasto-plasticity theory, including the Mohr–Coulomb model (e.g., [9, 11–15]), the hardening soil model (e.g., [8, 16, 17]), and the concrete model (e.g., [18–20]). They demonstrated that these models can moderately characterize the stress–strain behaviors of CACs. However, it is important to realize that when cement is mixed with natural clay, it forms a new structure called as soil–cement bonding. This bonding enhances the strength and reduces the compressibility of the clay and also exhibits

✉ Sompote Youwai
sompote.you@kmutt.ac.th

[1] Construction Innovations and Future Infrastructures Research Center, Department of Civil Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, 126 Pracha Uthit Rd., Bang Mod, Thung Khru, Bangkok 10140, Thailand

different behavior compared to original clay in its reconstituted or natural state. Consequently, the routinely used constitutive models for simulating the behavior of clayey soil seem to be inadequate in capturing the realistic characteristics of CAC.

To overcome the above-mentioned issues, the advancement in modeling clayey soils treated with cementitious materials has emerged as a significant research area within the field of soil stabilization techniques, and therefore, is attracting numerous researchers (e.g., [21–29]). Through a comprehensive review, it has been observed that various analytical constitutive models proposed in existing literature for simulating the behaviors of CACs commonly integrate the elasto-plastic theory within the frameworks of critical state or bounding surface plasticity. These models effectively incorporate rules for both hardening and softening, rendering them a prominent characteristic in capturing the complex behavior of CACs. As an instance, Kasama et al. [21] presented a constitutive model for simulating CAC responses that incorporated the cementation effect into the energy dissipation equation based on the critical state theory. They made adjustments to the critical state concept to accommodate the increase in strength of the CACs, resulting in an expanded stress range. Horpibulsuk et al. [24] utilized the Structured Cam Clay (SCC) model, which was developed by Liu and Carter [22], to simulate the behavior of CACs. They expanded the SCC model to account for CACs by modifying the mean effective stress and considering the cementation effect in which a failure envelope was assumed parallel to that of untreated clay, but with a specific shift to represent the cementation effect. In addition, Suebsuk et al. [26] made modifications to the SCC model to capture the strain-softening behavior typically observed when the soil–cement structure undergoes crushing. Furthermore, Nguyen et al. [27, 28] applied the fundamental principles of the critical state concept combined with the modified Cam-clay model to simulate CAC behaviors. This modification can effectively handle the degradation of cementation caused by the incremental mean effective stress and shear strain. Xiao et al. [29] proposed a novel constitutive model based on the bounding surface Cam-clay model for simulating CAC behavior, which takes into account the influence of cementitious bonding by incorporating a cohesion term into the Cam-clay flow rule. This model incorporates a mechanistic representation of bond breakage during plastic deformation, resulting in the reduction of cementitious bonding to simulate the loss of bonding. Referring to the above-mentioned literature, although various analytical constitutive models have been continuously developed and enhanced to accurately represent the stress–strain response of CACs, these models often have specific focuses, such as some types of elasto-plastic frameworks (e.g., critical state and bounding surface concepts), particular characteristics (e.g., hardening and softening behaviors), and also limited mixing proportions (e.g., cement and water contents). Furthermore, the existing constitutive models have also been formulated with various physical assumptions [30] and often require a considerable number of input parameters, making it challenging to define them accurately and optimally [31]. Besides, these models are unable to simulate the behavior of CACs with different mixing proportions and stress paths using a single set of model parameters. Hence, there is a need for a more general and robust constitutive model for CACs that can overcome these limitations.

In recent years, the adoption of artificial intelligence (AI)-based modeling approaches for solving complex engineering challenges has captivated the interest of many scientists and researchers. AI is widely and popularly used in diverse engineering disciplines to capture the complex associations between inputs and outputs. Particularly, in the field of geotechnical engineering, numerous researchers have successfully developed machine learning (ML)-based methodologies to investigate the behavior of various geotechnical structures (e.g., [32–35]). Currently, data-driven soil-constitutive models developed through ML techniques are gaining prominence due to their inherent advantages. ML algorithms provide flexibility as they are not restricted by assumptions or specific model classes, thereby allowing for their applicability to expand alongside the growth of datasets. The pioneering work by Ghaboussi et al. [36] first proposed the application of artificial neural networks (ANNs) in mechanical behavior modeling, initially focusing on plain concrete under biaxial loading, with subsequent successful modeling of stress–strain behaviors in sands accomplished using ANNs as well [37, 38]. Although ML models excel in capturing the complex nonlinear relationships between influential factors and the desired responses, their effectiveness is often limited to extracting superficial features and may require intricate engineering judgments [39]. Additionally, the strain–stress characteristic in materials is significantly influenced by their loading history. Unfortunately, ML methods are unable to account for the stress history-dependent properties of materials due to the absence of interconnections between different time steps, error accumulation, and the substantial computational expenses involved [30, 31]. To ensure the accurate prediction of stress–strain increments, it is imperative to incorporate the variations in features during stress–strain increments into the analysis. Specifically, it becomes essential to include the feature values from the preceding time step within the analysis, a consideration that holds particular significance in forecasting the stress–strain behavior of soils.

A particularly noteworthy advancement is the emergence of recurrent neural networks (RNNs), along with deep learning (DL) techniques, which can effectively address the aforementioned challenges by incorporating

previous sequential information into their calculations [40]. RNNs utilize weights and values from previous time steps to calculate the current step, and the predictions from the current time step are then passed on to the next time step for further computation. However, a challenge arises when dealing with long-time sequences during backpropagation, as the gradient may become too small to effectively propagate back to the starting time step. To overcome this issue, advanced models have been developed that apply gates to control the weight between time steps. Two notable extensions of RNNs are the "long short-term memory (LSTM) neural network" [41–43] and the "gated recurrent unit (GRU) neural network" [44–46]. These extended models have gained widespread utilization in time series forecasting, owing to their well-established nature. In geotechnical engineering, the stress–strain behavior of geomaterials, such as soils and rocks, depends considerably on the stress history. Few studies have developed RNNs-based constitutive models that account for this dependency. According to previous studies [30, 31, 47, 48], they proposed LSTM models that used stress, strain and state parameters as inputs to the LSTM cells, but they did not consider the features from the previous stress–strain points. This is a limitation that should be addressed in future research. The stress–strain behavior of CAC is influenced by its stress history and the proportions of water, base clay, and cement. While RNNs represent a promising approach for modeling this complex behavior, there has been no prior application of RNNs to CAC. We propose an RNN model that utilizes previous stress–strain data and the engineering properties of CAC as inputs. The principal challenge for this model lies in assessing its generalizability across diverse stress paths and mixing ratios. This study introduces the inaugural application of RNNs for simulating the stress–strain behavior of clay-cement mixture under a variety of conditions.

This study investigated the stress–strain behavior of CACs using three types of RNNs: traditional RNN, LSTM neural network, and GRU neural network. We conducted isotropically consolidated undrained triaxial compression (CIU) tests on 27 CAC specimens with different mixing proportions tested under stress- and strain-controlled conditions and three confining pressures. The stress–strain data from these tests were used as the training dataset for the RNN models. The objective of the developed models was to predict the stress–strain behavior of CACs for various scenarios of mixing proportions, confining pressures, stress paths, and shearing paths (i.e., loading and unloading compressions) by using a unified model. We also evaluated the prediction performance of the RNN models by comparing the test results with the model outputs and assessed their accuracy, reliability, and effectiveness in capturing the stress–strain behavior of the CACs.

The current study consisted of three major parts. The first part described the stress-path undrained triaxial test, specimen preparation, mixing ratio, and testing details. The second part evaluated the RNN models (traditional RNN, LSTM, and GRU models) by comparing the mean absolute percentage error (MAPE) value using k-fold cross-validation method. The input features for the models were the stress state and the mixing ratio of CACs. The third part demonstrated the application of the trained models to simulate the stress–strain characteristics of CACs. The contribution of this study to the geotechnical field was to demonstrate the applicability of using RNNs to capture the stress–strain behavior of CACs. The trained models could be applied to CACs with mixing ratios within the range of the training dataset. The RNN model structure could also be adapted to other types of clay-cement mixtures in both strain- and stress-controlled triaxial tests.

Deep Learning Algorithms

A Brief Introduction of Recurrent Neural Networks

RNNs are a class of ANNs designed for processing sequential data, such as time series or natural language, that were initially developed in the late 1980s [40, 49, 50]. RNNs exhibit dynamic temporal behavior by using recurrent connections that allow information to be propagated from one-time step to the next. Figure 1 illustrates the typical structure of the traditional RNN and its expansion in a time sequence. Unlike the feedforward neural networks, which can propagate information solely in one direction, traditional RNN incorporates feedback connections that can enable it to maintain an internal state, referred to as "memory", that encapsulates information from the past. This memory capacity empowers the traditional RNN to capture dependencies and patterns over time, making it highly suitable for tasks involving sequences or time series. However, traditional RNN often suffers from the vanishing gradient problem, where gradients diminish exponentially over time, resulting

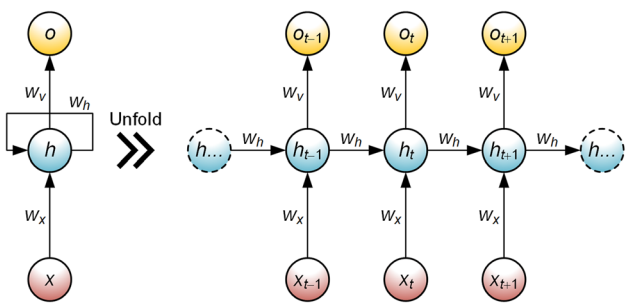

Fig. 1 Basic architecture of traditional RNN

in the limitation of their ability to capture long-term dependencies [39, 51, 52].

To mitigate the above-stated issue, various variants of traditional RNNs, such as LSTM and GRU neural networks, have been introduced and developed to address the vanishing gradient problem and improve the modeling of long-term dependency in sequential data. Both LSTM and GRU neural networks incorporate gating mechanisms that enable the network to selectively retain or discard information. In the field of geotechnical engineering, LSTM and GRU neural networks have been widely adopted for various applications, e.g., land reclamation settlement prediction, advancement rate prediction of tunneling excavations, failure probability prediction of slopes, deformation prediction of excavation works, and prediction of soil stress–strain responses [37, 53–58].

LSTM neural network is another type of RNN, which was first introduced by Hochreiter and Schmidhuber [41] and Gers et al. [59] that is specifically designed to handle long-term dependencies and overcome the vanishing gradient problem often found in traditional RNN. LSTM neural network introduces a memory cell and three gating units: the input gate, the forget gate, and the output gate [42], which were different functions in the LSTM unit as shown in Fig. 2. The memory cell enables LSTM to selectively retain and update information over time, while the gating units control the flow of information within the network. The input gate regulates the flow of information into the memory cell, the forget gate determines what information should be discarded from the memory cell, and the output gate governs the flow of information to the next time step. The complex architecture of LSTM neural network allows it to effectively capture and propagate relevant information while selectively disregarding irrelevant or redundant information.

GRU neural network is an extension of traditional RNN which was proposed by Cho et al. [44, 45] that introduces gating mechanisms to selectively control the flow of information. Figure 2 also presents the typical structure of the GRU unit. GRU units have two gates: the reset gate and the update gate [57]. The reset gate determines how much of the previous hidden state should be forgotten, while the update gate controls the incorporation of new information. By adaptively updating the hidden state, GRU effectively captures and propagates relevant information over long sequences, mitigating the vanishing gradient problem. Furthermore, the simplicity of GRU allows for more efficient training and inference compared to more complex models like LSTM.

Upon extensive review, it is demonstrated that traditional RNNs are computationally efficient and simple, yet they struggle with learning long-term dependencies. This issue arises because the hidden state, which encodes information from each time step, is susceptible to the vanishing gradient problem during training, potentially leading to model collapse. LSTM and GRU neural networks, special kinds of RNNs, are designed to overcome the vanishing gradient problem, thereby enabling better modeling of long-term dependencies. These networks possess a more complex structure than traditional RNNs, comprising memory cells and gates. The gates regulate the flow of information into and out of the memory cells, allowing the network to selectively retain or discard previous states. Consequently, LSTM and GRU neural networks are more adept at processing sequential data of varying lengths and complexities. The LSTM neural network is particularly noted for its superior

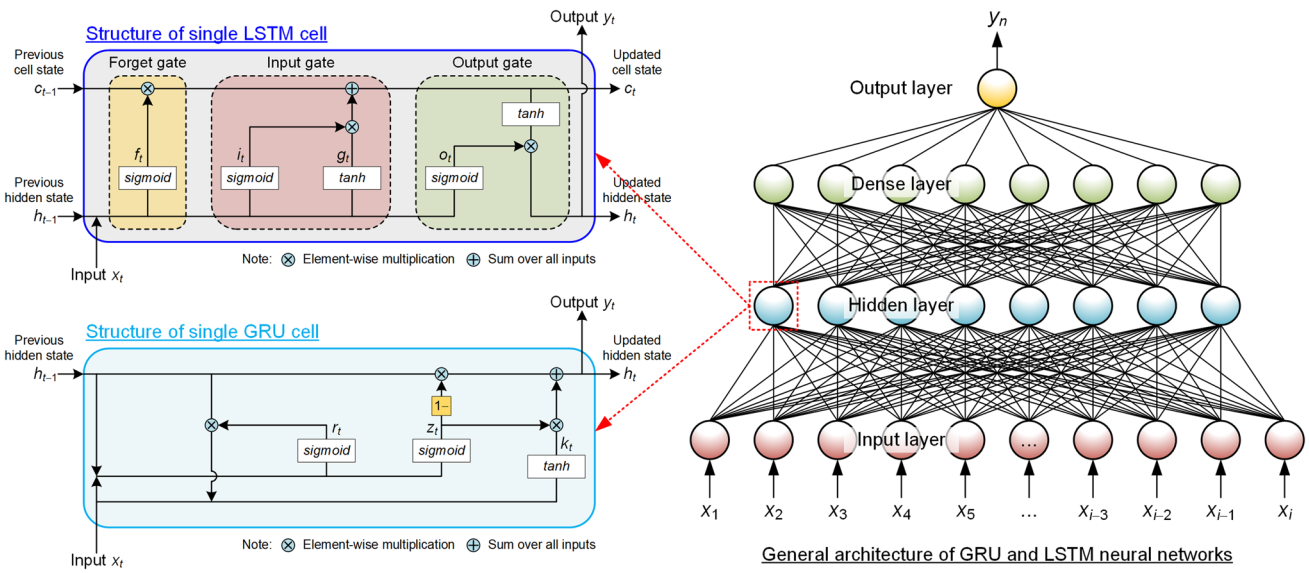

Fig. 2 Architecture of LSTM and GRU neural networks with the detailed structure of their single cells

performance in tasks requiring long-term memory, such as machine translation and speech recognition. Meanwhile, the GRU neural network has been shown to achieve performance comparable to that of the LSTM in many applications, while being computationally more efficient. Thus, the choice between these architectures depends on the specific requirements of the task at hand, including available computational resources and the desired balance between complexity and performance. It is important to note that the methodologies involving significant formulae related to these RNNs, namely traditional RNN, LSTM neural network, and GRU neural network, are detailed in the Appendix.

Distinctive Features of the Study Compared to Previous Research

- In this study, a comparison was conducted between different types of RNNs. The baseline model used in the comparison was a traditional RNN. The comparison and subsequent discussion of the findings are presented in this paper.
- This study encompasses the simulation of stress–strain relationships using various stress paths derived from the CIU tests on CACs. These stress paths include both compression loading and compression unloading stress increments. This simulation can be further extended for use in the applications of geotechnical engineering projects.
- The study investigates the impact of incorporating features from previous time steps in the analysis. Specifically, the features from the preceding time step are added as an additional dimension in the input tensor of the LSTM and GRU perceptron. This inclusion may influence the prediction errors, as the stress–strain increments of geotechnical materials are dependent on their stress history.

Experimental Tests

In the current study, a series of CIU tests on CACs were performed under both stress- and strain-controlled conditions. The purpose of these tests was to gather extensive stress–strain data, which would be used as the training dataset in RNN models. Under the stress-controlled condition, two shearing paths controlled by different ratios of the incremental deviator stress over the confining pressure ($\Delta q/\Delta p$) were considered in this study: (1) loading compression path with $\Delta q/\Delta p = +3.0$ and (2) unloading compression path with $\Delta q/\Delta p = -1.5$. The CAC preparation for the entire CIU testing program involved three different C_w/A_w mixing proportions and each C_w/A_w mixing proportion was subjected to three confining pressures (p_c) of 50, 100, and 200 kPa.

Materials

This study utilized the Pa Sak clay as the base clay for preparing CAC mixtures. The clay was extracted from a depth of 3 – 5 m below the ground surface, situated within the soft clay layer along the Pa Sak River in Ayutthaya province, located on the central plain of Thailand. The collection site was part of the Pasak River wall project, approximately 100 km north of Bangkok [10]. The natural water content of the Pa Sak clay was found to be approximately 45%, with a liquid limit (LL) of 83%, classifying it as a high-plasticity clay (CH) according to the Unified Soil Classification System (USCS). Furthermore, the clay exhibited a total unit weight of 15 kN/m^3, an initial void ratio of 2.24, and a specific gravity of 2.71. The raw materials employed in the preparation of the tested specimens were Ordinary Portland Cement Type I and untreated clay.

Preparation of Tested Specimens

Before introducing the cement slurry for the mixing process, the untreated clay sample was remolded to obtain the desired remolding water content (w^*) for achieving the flowability and workability of soil–cement mixture. In this study, w^* of 125% was adopted. To attain the desired w^*, an additional amount of water (ΔW_W, kg) was added to the untreated clay sample, which can fundamentally be calculated by the following equation [60, 61]:

$$\Delta W_W = \frac{W_S\left(w^* - w_0\right)}{1 + w_0}, \tag{1}$$

where W_S and w_0 are the total weight (kg) and the initial water content (%) of untreated clay sample in each batch before remolding. To prepare the remolded clay, the untreated clay sample, along with the calculated ΔW_W, was placed inside a portable mechanical soil mixer and blended thoroughly for a few hours. Once the remolded clay sample is well-blended, a water content test was conducted. The remolded clay sample was then poured into a container and sealed with the plastic film layers before being placed in a humidity-controlled room overnight. If the water content of the remolded clay sample was within 1% of the desired w^*, this sample was considered ready for admixture mixing.

In the mixing process, remolded clay with the desired w^* of 125% was blended with the cement slurry, which was prepared using a water-to-cement ratio (w/c) of 1 according to the previous studies [61, 62]. Consequently, the required cement content (A_w) for each mixing proportion can be directly calculated through Eq. (2) [61]. Noted that A_w was determined as the percentage ratio of the cement weight to the dry weight of clay and also varied depending on the

prescribed C_w/A_w ratio. CAC mixture was then carefully blended using a portable mechanical mixer until a uniform mixture was achieved. Due to the additional water provided by the cement slurry, the total amount of water included in each CAC mixing proportion was thus referred to as the total clay-water content (C_w).

$$A_w = \frac{w^*}{\left(C_w/A_w\right) - (w/c)}. \tag{2}$$

For the molding process, the CAC slurry was carefully poured into a PVC mold with a diameter of 70 mm and a height of 150 mm, in four layers. In each layer, a small tamping rod was periodically used to compact the CAC slurry and minimize the presence of air bubbles [61]. Following the molding process, the CAC specimens within the molds were wrapped with polyethylene films and placed in a humidity-controlled room at a maintained humidity of 97% and an ambient temperature of 25 °C for 5 days. Subsequently, the CAC specimens were gently extruded from the molds and re-wrapped with polyethylene film for further curing within a humid room until reaching the target curing period of 28 days. Prior to the triaxial testing, both ends of each CAC specimen were carefully trimmed to achieve smooth and perpendicular end surfaces in the axial direction.

Testing Method and Program

The CIU tests were conducted on CAC specimens under the curing period of 28 days, following the ASTM D4767 - 11 [63]. Prior to the isotropic compression, the specimens were saturated under a back-pressure of 300 kPa until achieving a B-value greater than 0.9 to ensure full saturation [61]. Subsequently, the consolidation process can be initiated in which p_c values of 50, 100, and 200 kPa were used in this study. Following consolidation, the specimens underwent shearing by the prescribed stress paths as stated above. In the current study, a total of 27 CACs specimens were utilized for CID tests, with variations in C_w/A_w ratios, p_c values, and stress paths. To accurately reflect actual construction practices, the cement mixing ratio was determined based on the real case history of a river wall reinforced with soil–cement columns [10]. The w^* was set at 125% of the LL. This value of water content was chosen because it facilitated the pre-jetting of water to disrupt the natural clay structure and ensured homogeneous mixing with the cement. Additionally, the confining stress was selected to encompass the potential stress range encountered in actual field conditions. Furthermore, the stress path was designed to reflect possible stress increments in the real-field condition of a river wall structure stabilized by soil–cement columns, attributable to both compression loading and unloading [10]. The specific details of these specimens are provided in Table 1.

Test Apparatus and Measuring Devices

Figure 3 presents the system of triaxial compression apparatus utilized for testing CAC specimens. In this system, a cell pressure pipe was connected with a cell pressure transducer, which measured the cell pressure value within the triaxial cell. This pressure was then applied to the top of the triaxial test cell. To precisely control the cell pressure for the testing under unloading stress path, the electro-pneumatic transducer along with the air pressure regulator was used to serve the purpose of regulating and controlling the cell pressure. The ASCII code programming was adopted to accurately control the cell pressure for performing the compression unloading tests. For measuring the pore pressure value, a distinct pressure transducer was connected to a burette located below it. Throughout the test, several data were continuously monitored and recorded automatically. These parameters included changes in axial loading, axial displacement, pore pressure, and cell pressure. The measurements were carried out with the aids of various instruments including a load cell, a linear variable differential transformer, a pore pressure transducer, and a cell pressure transducer. The integration of these above-mentioned instruments with a computer and a data logger system can achieve efficient data acquisition and recording in which the data was recorded at intervals of 1 s. Therefore, during the test, the CIU tests also export stepwise axial stress (σ_1), cell pressure (σ_3), pore pressure (u), and axial strain (ε_a). For the triaxial test in the principle stress space, the deviator stress (q) and the mean principle stress (p) can be derived from the triaxial results as expressed in the Eqs. (3) and (4), respectively.

Table 1 A summary of the CAC specimens used for CIU tests in this study

Mix No.	C_w/A_w (–)	w^* (%)	A_w (%)	C_w (%)	p_c (kPa)	No. of specimens
1	2.92	125	65.1	190.1	50, 100, 200	9
2	6.92	125	21.1	146.0	50, 100, 200	9
3	10.92	125	12.6	137.6	50, 100, 200	9

Notes: For each testing condition (i.e., stress-controlled condition with $\Delta q/\Delta p = +3.0$, stress-controlled condition with $\Delta q/\Delta p = -1.5$, and strain-controlled condition), 3 specimens were used

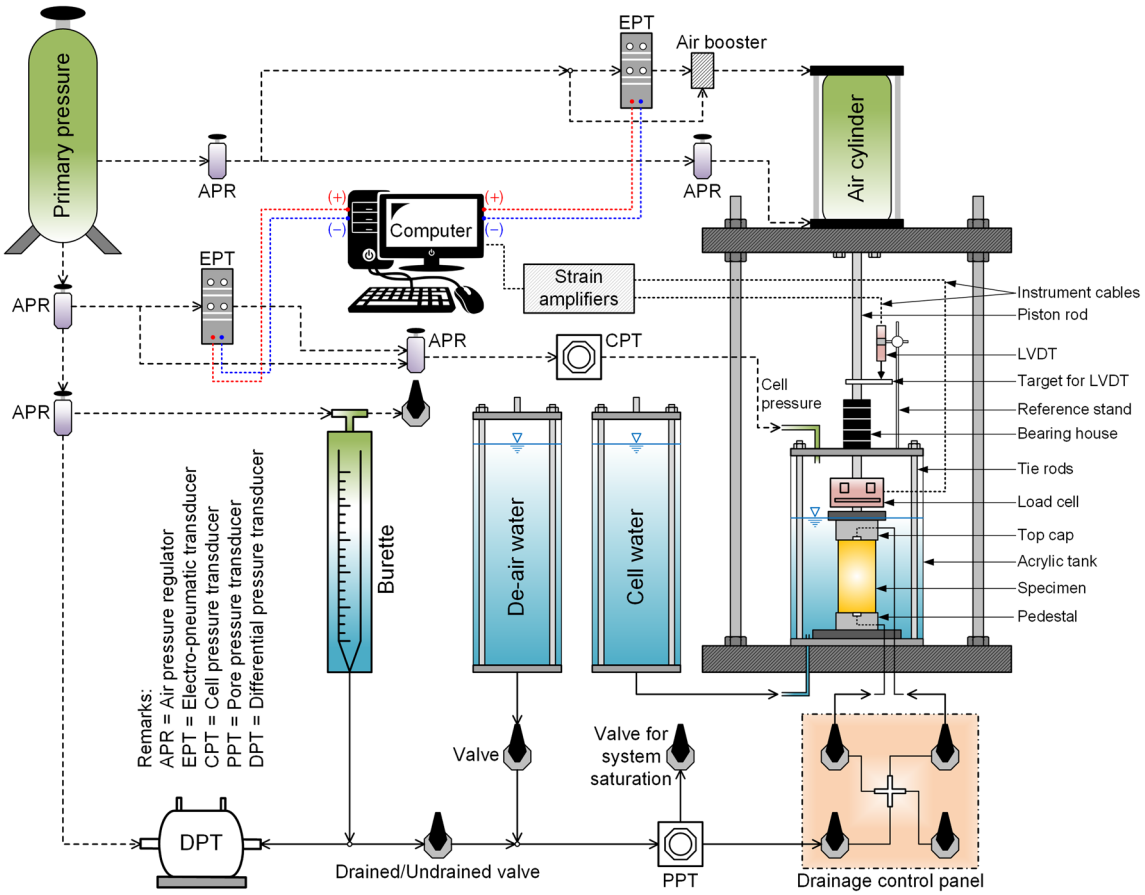

Fig. 3 Diagram of the triaxial machine system used for CIU tests under both strain- and stress-controlled conditions

$$q = (\sigma_1 - u) - (\sigma_3 - u) = \sigma'_1 - \sigma'_3, \tag{3}$$

$$p = (\sigma'_1 + \sigma'_2 + \sigma'_3)/3 = (\sigma'_1 + 2\sigma'_3)/3. \tag{4}$$

Modeling Details

The stress–strain responses of CACs were predicted using the RNN models implemented within the Tensorflow framework [64]. The coding of the models was developed with the aid of the Keras library [65], which is a Python-based DL application programming interface. To facilitate efficient matrix operations and computations, the Numpy library, a fundamental tool for scientific computing, was employed to construct a tensor as input for the RNN models. The data was partitioned into training, testing, and evaluation sets using Numpy arrays. The analysis was performed on a cloud-based server using the Google Colaboratory platform, harnessing its powerful GPU capabilities for computational tasks.

Raw Dataset

Following the CIU tests, each soil sample, characterized by a predetermined C_w/A_w proportion, exhibited a distinct stress–strain relationship (stress–strain sample) under a specific confining pressure (p_c = 50, 100, and 200 kPa) and stress path ($\Delta q/\Delta p = +3.0$ and -1.5). The raw data set comprised 17 sets of training samples for stress-controlled conditions and 8 sets of training samples for strain-controlled conditions. It is important to note that, due to instrument malfunctions during testing, one tested sample from each of stress- and strain-controlled conditions could not be obtained for the training data.

The first set of input data contains six variables derived from each tested sample, including q, p, ε_a, u, and initial mean principle stress (p_i), and the mixing proportion variable (C_w/A_w). According to the study of Miura et al. [66], they implied that the stress–strain behavior of CACs is remarkably affected by C_w/A_w and this parameter should be

thus included in the prediction model. Additionally, another set of input data consists of two variables representing the physical properties of CAC specimens after curing, namely the after-curing unit weight (γ_{ot}) and after-curing water content (w_{ot}). These two variables were identified as crucial parameters influencing the stress–strain responses of CACs [60, 61, 67, 68]. Another parameter included in the prediction model was the state parameter (ψ), proposed by the recent study of Phutthananon et al. [61], which was developed based on critical state theory. This ψ parameter effectively characterizes the mechanical properties of CACs with varying mixing proportions and curing states, tested under different stress conditions. The input parameters mentioned earlier were identical for both stress- and strain-controlled prediction models proposed in this study. Notably, the feature related to the applied stress path was not included as input data, as the RNN models inherently incorporate the effects of increasing stress and strain from the previous step. It is worth mentioning that the selected input data in this study align with the features used in previous studies [30, 31, 47].

Data Preprocessing

To reduce the effect of different scales and speed up the convergence process, we applied the min–max scaling technique to the data. This technique can effectively address concerns about feature dominance, enhance convergence, accommodate dissimilar input distributions, improve generalization capabilities, and also facilitate feature comparison. By normalizing the input data to a unified range, the RNN models are capable of attaining more equitable learning dynamics, achieving swifter convergence, and exhibiting heightened performance on the previously unseen data. The application of the min–max scaler, as delineated by Eq. (5), ensures the normalization of all data within the confines of a − 1 to 1 range, while preserving the original distribution unaltered [69]:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}\left(\bar{x}_{\text{max}} - \bar{x}_{\text{min}}\right) + \bar{x}_{\text{min}} \tag{5}$$

where x_{norm}, x_{max} and x_{min} refer to the normalized, maximum and minimum values of the input data, respectively; $\bar{x}_{\text{max}}$ and $\bar{x}_{\text{min}}$ are respectively − 1 and 1. By using the min–max scaling approach, the statistical characteristics for all input data are summarized in Tables 2 and 3 for the cases of CIU tests under strain- and stress-controlled conditions, respectively.

Performance Evaluation

To assess the capability of the prediction model and identify potential problems such as underfitting or overfitting, a k-fold cross-validation approach was implemented [53, 54, 70]. This approach was adopted to reduce the impact of high variance and ensure a fair evaluation of the model's performance. It is widely recognized that relying solely on a single set of validation data may lead to unreliable validation loss estimates. To overcome this limitation, for this study, the training data was partitioned into ten equal-sized folds. Subsequently, the model was trained on nine of the folds, while the remaining fold was utilized for evaluation. This process was repeated ten times at every epoch, resulting in the model being trained and evaluated ten times on different training folds. In this study, the mean absolute percentage error (MAPE) was employed as an indicator to evaluate the prediction performance, which was determined by calculating the average MAPE across the ten validation folds, as denoted by Eq. (6).

$$\text{MAPE} = \frac{1}{10}\sum_{j=1}^{m}\left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^p - y_i^m}{y_i^m}\right| \times 100\%\right)_j \tag{6}$$

Table 2 Statistical characteristics of input variables used for the prediction model under strain-controlled condition

Statistical value	q	p	ε_a	u	γ_{ot}	w_{ot}	C_w/A_w	p_i	ψ
Mean	–0.154	–0.212	–0.110	–0.339	–0.041	0.071	–0.074	–0.259	–0.260
Std	0.420	0.464	0.552	0.451	0.634	0.569	0.777	0.793	0.793
Min	–1.000	–1.000	–1.000	–1.000	–1.000	–1.000	–1.000	–1.000	–1.000
Max	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 3 Statistical characteristics of input variables used for the prediction model under stress-controlled condition

Statistical value	q	p	ε_a	u	γ_{ot}	w_{ot}	C_w/A_w	p_i	ψ
Mean	–0.580	–0.378	–0.949	–0.345	–0.105	0.055	0.043	–0.119	0.040
Std	0.462	0.488	0.102	0.555	0.546	0.722	0.882	0.766	0.600
Min	–1.000	–1.000	–1.000	–1.000	–1.000	–1.000	–1.000	–1.000	–1.000
Max	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

where m is the total number of folds; n denotes the total number of data within one validation set; y_i^p and y_i^m are respectively the predicted and measured values at the ith data of the jth fold.

Model Configuration

The optimal configuration of DL models is intractable due to the combinatorial explosion of possible configurations. Therefore, we followed a trial-and-error approach to find the best configuration, as commonly done in previous studies [31, 53, 54, 57]. Our DL model consists of two main components: LSTM and Dense layers, as shown in Table 4. The model architecture has two LSTM layers and a multilayer perceptron (MLP). The first LSTM layer passes the outputs of all time steps to the second LSTM layer. The second LSTM layer then feeds into the MLP. The number of LSTM units decreases from 50 in the input layer to 25 in the hidden layer, reducing the dimensionality of the tensor gradually before entering the MLP. The MLP layer has a Dense layer, which is a standard fully connected perceptron in neural networks. It uses the tanh activation function, which maps input values to the range $(-1, 1)$ using a hyperbolic tangent. Unlike the sigmoid function, the tanh function is symmetric around the origin and has steeper gradients, alleviating the vanishing gradient problem of the sigmoid function. Moreover, the tanh function centers output values around zero, easing the optimization process. Finally, the DL network outputs an array with a dimension of five.

Based on the trial-and-error approach and the trade-off between prediction accuracy and training time, several factors were considered to determine the optimal configuration for the predicted models under stress- and strain-controlled conditions, as outlined in Table 5. In this study, the Adam algorithm was employed to optimize the back-propagation procedure. This algorithm is a widely-utilized algorithm in DL that dynamically adjusts neural network parameters in real time, enhancing both accuracy and speed. It achieves this by adapting the learning rate based on historical gradients and momentum [54–56]. Default values of dropout rate, learning rate, and regularization determined in Keras library were used in this study. In this study, the loss function of mean square error (L_{MSE}) is chosen to evaluate the error between the predicted and actual values, as expressed in Eq. (7):

$$L_{\mathrm{MSE}} = \frac{1}{k}\sum_{i=1}^{k}\left(y_i^p - y_i^m\right)^2 \tag{7}$$

where k refers to the number of samples; y_i^p and y_i^m stand for the same definitions previously mentioned. The MSE calculates the average of the squared differences between the predicted and actual values. Notably, MSE assigns greater weight to larger errors, making it particularly sensitive to outliers and extreme values. Given its differentiability and ease of optimization, MSE is frequently and commonly used as a loss function for training DL (e.g., [30, 48, 57]). A smaller L_{MSE} value indicates a more accurate prediction by the constructed model. The number of epochs was selected by monitoring the training process for significant improvement or deterioration. Overfitting and underfitting were evaluated by examining the L_{MSE} value and the training number. Convergence between the training and test data trends was considered desirable, indicating a certain number of epochs where the training process reaches stability. Therefore, the process of model training aims to determine

Table 4 The model configuration

Configuration (Activation function)	Strain-controlled condition	Stress-controlled condition
LSTM (ReLU)	50	50
LSTM (ReLU)	20	25
Flatten	–	–
Dense (tanh)	10	25
Dense (tanh)	5	5
Parameters	16,345	18,780

Table 5 Optimal configurations of the prediction model adopted in this study

Configuration	Strain-controlled condition	Stress-controlled condition
Hidden layer	4	4
Architecture*	50 – 20 – 10 – 5	50 – 25 – 25 – 5
Epoch	50	50
Batch size	32	32
Activation function	ReLU, tanh	ReLU, tanh
Optimizer	Adam	Adam
Loss function	MSE	MSE
Learning rate	0.001	0.001

*Architecture configuration as presented in Table 4

the model parameters that minimize the L_{MSE}, reflecting the nature of the data.

Figure 4 depicts the configuration of the array utilized as input for the DL model. This array is implemented as a three-dimensional Numpy array designed for the LSTM perceptron. The first dimension signifies the number of training data samples. The second dimension encompasses the sequential parameters, denoted as q, p, ε_a, u, γ_{ot},w_{ot}, C_w/A_w, p_i, and ψ, (see detailed selections in the above "Raw dataset"). The third dimension specifies the time step, which denotes the values of the aforementioned parameters at various time lags. For this study, the time step was varied from 1 to 3, and the subsequent results are presented in the subsequent section. The predictive value, or rather the predicted label, in this study is composed of q, p, ε_a, and u. The output array is a two-dimensional array, with the first dimension indicating the number of training samples and the second dimension representing the number of expected outputs. Consequently, the complete dataset, comprising different mixing ratios and stress paths, was combined and shuffled. Both the data and the model were evaluated using the k-fold cross-validation method to assess the effectiveness of each model type and structure.

Results and Discussion

Comparative Results of Traditional RNN, GRU, and LSTM

To evaluate the accuracy of the RNN models in predicting the stress–strain responses of CACs, we adopted the k-fold cross-validation method suggested by Jung and Hu [71], utilizing MAPE as the performance criterion. This method aimed to validate the model and prevent overfitting. We divided the data into ten subsets, using one subset as the test set and the remaining nine subsets as the training set in each iteration. This process was repeated for each subset, and the results were averaged. We computed the average MAPE across the iterations and plotted them as shown in Fig. 5. The MAPE results based on k-fold cross-validation showed satisfactory outcomes for both stress- and strain-controlled conditions, with small MAPE deviations ranging from approximately 4 to 10% depending on the model types and time steps. The prediction performance of traditional RNN was inferior to those of other predicting models, especially when we increased the number of time steps backward in the training array. This is because RNNs suffer from the vanishing gradient problem, which means that they struggle to learn long-term dependencies in the data, as the gradients

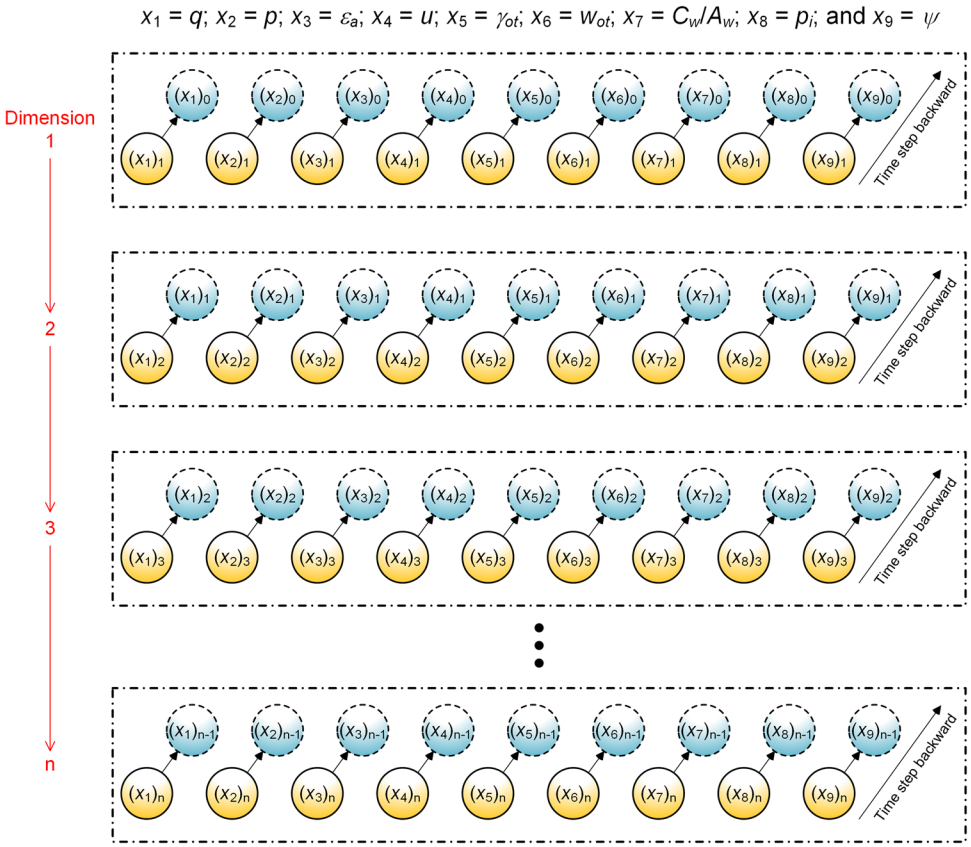

Fig. 4 The structure of input array for deep leaning model

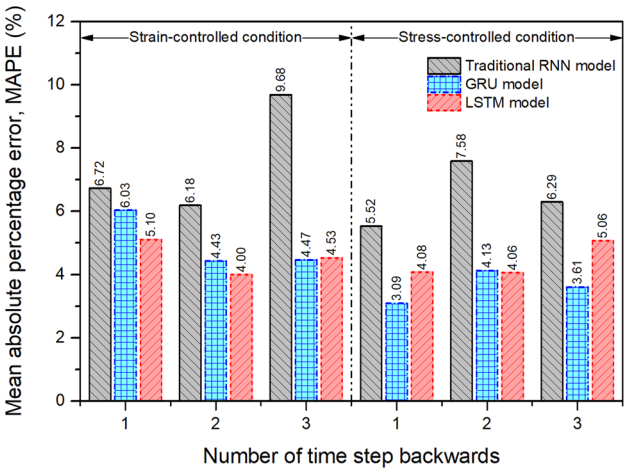

Fig. 5 Comparison of MAPE results among the traditional RNN, LSTM neural network, and GRU neural network using k-fold cross-validation approach with varying time steps

tend to diminish or vanish as they propagate back through time [72].

We compared the predictive performance of the LSTM model with that of other models for both stress- and strain-controlled triaxial tests (Fig. 5), utilizing a two-time step backward scheme. The LSTM model achieved a low MAPE of approximately 4% in both stress- and strain-controlled conditions, demonstrating its high accuracy and reliability in simulating the stress–strain behavior of CACs (see details in the next subsection). Conversely, the stress-controlled triaxial simulation indicated that the GRU model with a 1-time step backward was the most effective. However, it is important to note that increasing the time step backward for the GRU model led to higher errors. Furthermore, the GRU model performed less accurately in predicting strain-controlled triaxial tests compared to the LSTM model, particularly when using 1- or 2-time steps backward for the strain-controlled condition. These findings highlight the superior performance of the predicting model-based LSTM neural network over traditional RNN and GRU models in this context. However, it is worth mentioning that training the LSTM model is quite time-consuming due to its complex structure. The involvement of intricate equations necessitates significant computational resources and extended training durations. Nevertheless, the LSTM neural network, based on the predictive model, can effectively forecast the responses of CACs, i.e., stress and strain. Moreover, the LSTM model was capable of reproducing complex features that conventional constitutive models cannot capture, such as the softening effect caused by bond breakage of cementation during shearing (to be discussed in a subsequent subsection). Considering the influence of stress history on CAC behavior, the optimal backward time step for the predictive model-based DL was determined to be 2-time steps. Therefore, it is recommended to employ the LSTM neural network with a 2-step backward configuration as the preferred model structure for predicting stress–strain responses of CACs under both stress- and strain-controlled conditions. Furthermore, based on this finding, we propose extending this research to develop a constitutive model that characterizes CAC behavior accurately and implementing it in finite element analysis to investigate the stress–strain relationship more thoroughly.

Prediction Results of LSTM Models

This section presents the predicted results of the trained LSTM network model with two backward time steps, which exhibit the best performance in simulating strain and stress-controlled triaxial tests. Figures 6 and 7 depict the satisfactory performance of the LSTM model in predicting the $q-p$ and $q-\varepsilon_a$ relationships under strain-controlled triaxial tests, respectively. Notably, the prediction error escalated as C_w/A_w increased (see Fig. 6). Moreover, the model can effectively capture the softening behavior observed after the peak stress value, as shown in Fig. 7. However, there were deviations in predicting pore water pressure as compared to the actual test results, resulting in an inaccurate prediction of the stress path, particularly for the case of $C_w/A_w = 10.92$. Nevertheless, based on overall predicted results, the LSTM model demonstrated the ability to predict the stress–strain relationship for low cement content and the relationship between deviatoric stress and distortional strain.

As for stress-controlled triaxial tests, the LSTM model exhibited strong predictive performance in forecasting the behaviors of CACs during testing with various stress paths, as indicated in Figs. 8 and 9. The applied stress paths consisted of the compression loading with $\Delta q/\Delta p = +3.0$ path (see Fig. 8) and compression unloading with $\Delta q/\Delta p = -1.5$ path (see Fig. 9). Remarkably, the predicted stress path closely aligned with the experimental results, indicating the model's ability to forecast the changes of pore water pressure under different stress paths without requiring an incremental stress path as an input feature for analysis. Interestingly, the model's predictions demonstrated greater conformity to the compression loading scenario as compared to the compression unloading scenario. Furthermore, the LSTM model also exhibited excellent predictive accuracy for q and ε_a relations with different mixing ratios, as depicted in Fig. 10.

This study evaluates the performance of RNN models, especially LSTM and GRU neural networks, for predicting the stress–strain behavior of CACs under various stress paths using a single set of model parameters. The model incorporates the key features of CACs, such as the mixing ratio, initial stress state, index properties and state parameter. The model is trained on a limited dataset of 25 samples,

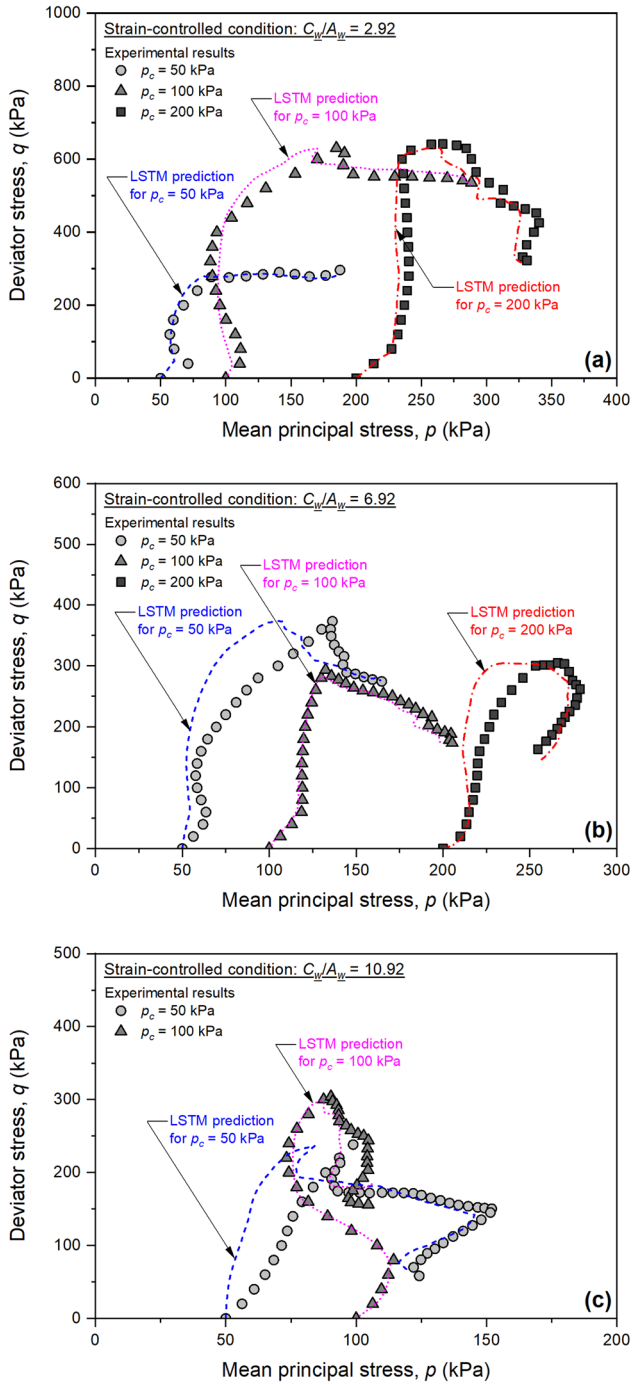

Fig. 6 LSTM performance on prediction of $q - p$ relationships under strain-controlled triaxial tests with different mixing proportions

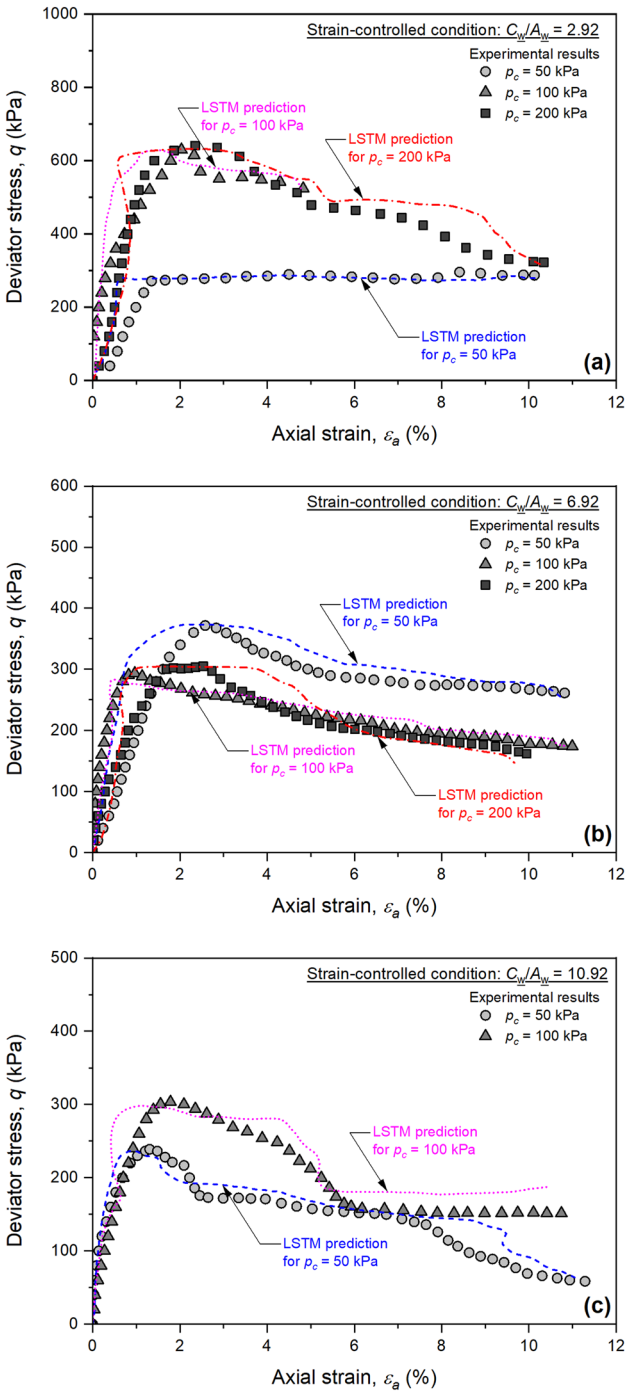

Fig. 7 LSTM performance on prediction of $q - \varepsilon_a$ relationships under strain-controlled triaxial tests with different mixing proportions

which restricts its generalizability to other CACs. A larger and more diverse dataset is needed to improve the robustness and accuracy of the DL model. However, the proposed DL model architecture with LSTM shows a high predictive capability on unseen data (validation data), achieving a low MAPE of 4% through a k-fold cross-validation procedure. The model framework and structure can be applied to other cemented soils with different soil types or bonding materials, using a similar RNN model structure. The model can also be extended to a full generative model based on the sequence to sequence transformer concept [73]. This study provides a basis for developing a constitutive model for cemented soil, which can facilitate realistic simulations in finite element analysis.

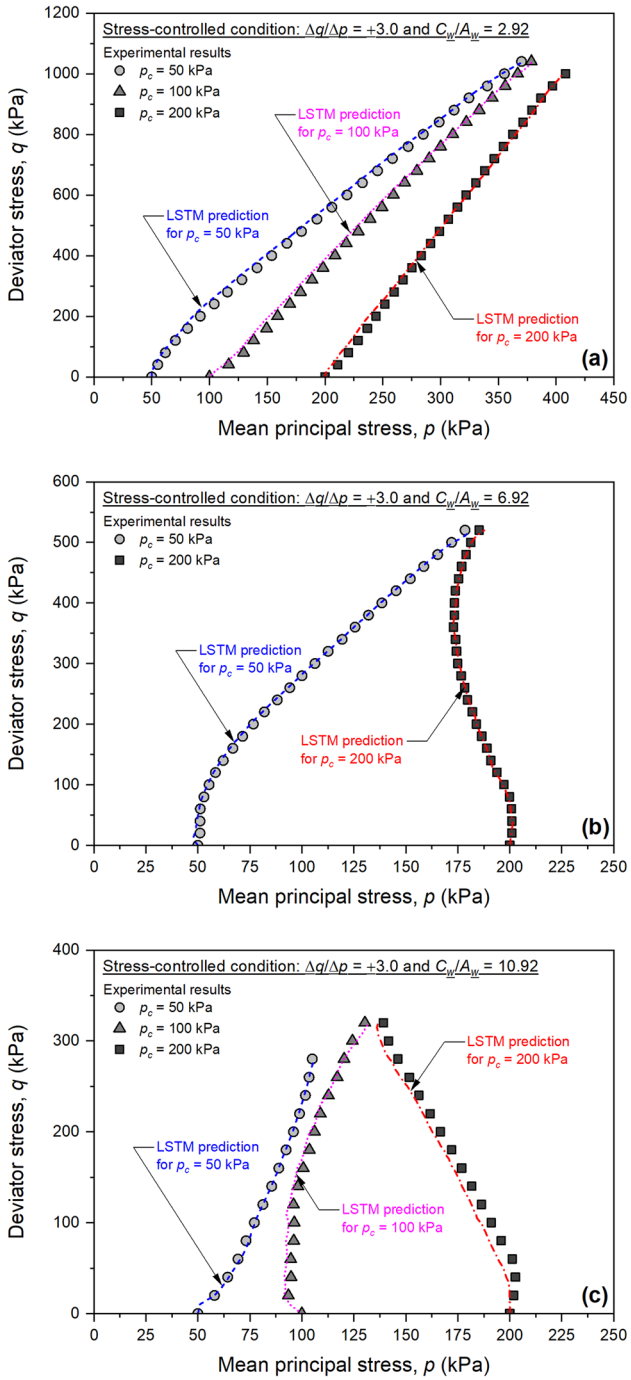

Fig. 8 LSTM performance on prediction of $q - p$ relationships under stress-controlled triaxial tests with stress path of $\Delta q/\Delta p = +3.0$ by varying mixing proportions

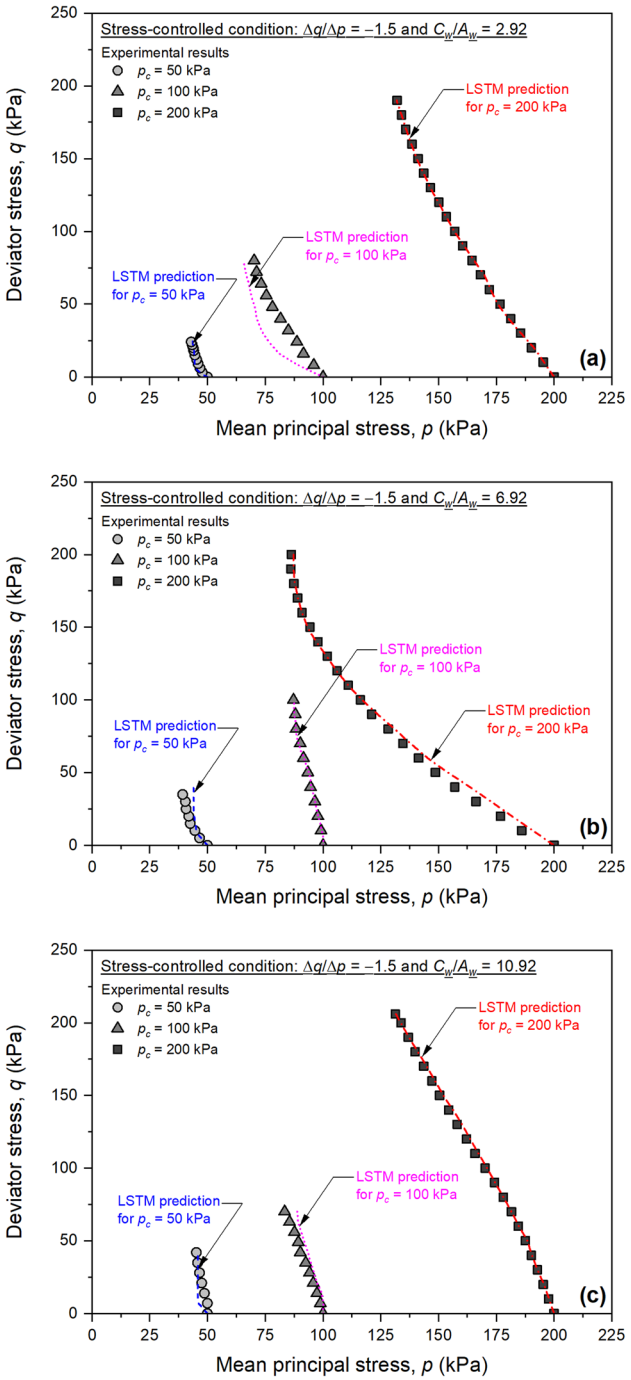

Fig. 9 LSTM performance on prediction of $q - p$ relationships under stress-controlled triaxial tests with stress path of $\Delta q/\Delta p = -1.5$ by varying mixing proportions

Conclusions

In conclusion, this paper presented a novel application of recurrent neural networks (RNNs) to predict the stress–strain behavior of cement-admixed clays (CACs) under various mixing proportions, confining pressures, stress paths, and shearing conditions. Three types of RNNs, namely, traditional RNN, long short-term memory (LSTM) neural network, and gated recurrent unit (GRU) neural network, were compared in this study. The conclusion can be made as follows:

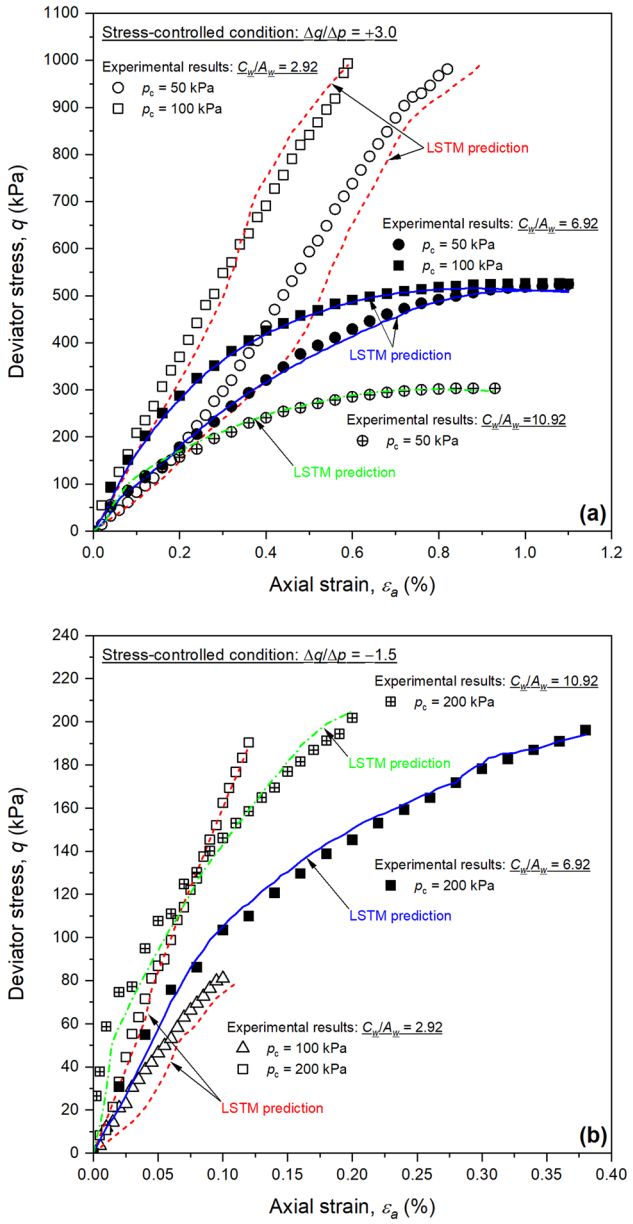

Fig. 10 Examples of predicting $q - \varepsilon_a$ relationships of LSTM model under stress-controlled condition with different stress paths

1. The proposed structure for DL, combining RNNs and feed-forward neural network perceptron, incorporated the features of the previous time step in the analysis. The results demonstrated that LSTM and GRU neural networks outperformed the traditional RNN in terms of prediction accuracy. These trained models were able to effectively capture the complex and nonlinear characteristics of the stress–strain relationship of CACs under different stress paths using a unified set of parameters.
2. Based on the findings, the LSTM neural network with a 2-time step backward approach was identified as the best model for predicting the behaviors of CACs in both strain- and stress-controlled triaxial tests. The cross-validation results showed a very low prediction error of approximately 4% according to the MAPE. These outcomes highlight the potential of utilizing DL techniques, specifically the LSTM and GRU neural networks, for accurate and reliable predictions of CAC behavior, contributing to improved understanding and design of CAC materials.
3. This study employed RNN models to simulate the stress–strain behaviors of CAC across various stress paths and mixing ratios, utilizing a consistent model weight and architecture. The algorithm is capable of extension for use as a constitutive model to determine the deformation characteristics of CAC. Additionally, it can be integrated into a finite element method to facilitate the analysis and prediction of ground improvement efforts involving CAC.

Appendix

The fundamental architecture of traditional RNN is comprised of recurrent neurons that process inputs and integrate information from the previous inputs by considering their previous states During the prediction process of traditional RNN, at each time step, a recurrent neuron takes the input value (x_t), combines the current hidden state (h_t) with the hidden state derived from the previous time step (h_{t-1}), and generates the output value (o_t) with the help of activation function. The current hidden state h_t can be calculated as follows.

$$h_t = \tanh\left(w_h \times h_{t-1} + w_x \times x_t + b_h\right) \tag{A1}$$

The output o_t can be generated by the current hidden state h_t as given below.

$$o_t = \tanh\left(w_v \times h_{t-1} + b_o\right) \tag{A2}$$

In these above equations, w_x, w_h, and w_v coefficients represent respectively the input, recurrent, and output weight matrices shared across the sequence; b_h and b_o are the bias weight vectors; tanh refers to the applied hyperbolic tangent activation function, which can be expressed as follows.

$$\tanh(x) = e^x - e^{-x} / e^x + e^{-x} \tag{A3}$$

According to the valuable research of Hochreiter and Schmidhuber [41], LSTM neural networks are composed of memory cells that allow them to selectively retain and update information over time. The LSTM unit incorporates three main components: the forget gate, the input gate, and the output gate, which regulate the flow of information within the network through a sigmoid activation function.

The forget gate (f_t) is computed using the current input x_t and the previous hidden state h_{t-1} as presented follows.

$$f_t = \text{sigmoid}\left(w_f \times x_t + u_f \times h_{t-1} + b_f\right) \tag{A4}$$

The input gate (i_t) justified the entered values for updating the cell state (c_t) by utilizing two inputs (h_{t-1}, x_t) and the sigmoid function as expressed below.

$$i_t = \text{sigmoid}\left(w_i \times x_t + u_i \times h_{t-1} + b_i\right) \tag{A5}$$

The output gate (o_t) determines which information in the LSTM unit should be exported according to the block memory and inputs (h_{t-1}, x_t) as given below.

$$o_t = \text{sigmoid}\left(w_o \times x_t + u_o \times h_{t-1} + b_o\right) \tag{A6}$$

The candidate cell state (g_t) is determined to update the previous cell state (c_{t-1}) using two inputs (h_{t-1}, x_t), and the tanh activation function to squash the values between − 1 and 1, as represented below.

$$g_t = \tanh\left(w_g \times x_t + u_g \times h_{t-1} + b_g\right) \tag{A7}$$

The updated cell state (c_t) is consequently calculated by combining the contributions from the forget gate f_t, previous cell state c_{t-1}, input gate i_t and candidate cell state g_t, which is formulated as follows.

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{A8}$$

The hidden state is computed by applying the output gate o_t to the hyperbolic tangent activation function of the updated cell state c_t, which is defined as below.

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{A9}$$

Referring to the Eqs. (A4) to (A9), coefficients w_f, w_i, w_o, and w_g represent the weights of forget gate, input gate, output gate and input unit, respectively; coefficients u_f, u_i, u_o, and u_g are the recurrent weights of forget gate, input gate, output gate and input unit, respectively; coefficients b_f, b_i, b_o, and b_g represent the bias weight vectors of forget gate, input gate, output gate and input unit, respectively; $\odot$ is the Hadamard product, which refers to the element-wise product among the matrices; and sigmoid represents another activation function as given in following equation.

$$\text{sigmoid}(x) = 1/1 + e^{-x} \tag{A10}$$

For the GRU neural network, the key innovation of GRU unit lies in their integration of two gates: the update gate (z_t) and the reset gate (r_t). Based on the study of Chung et al. [46], the update gate z_t in GRU unit determines how much of the new information should be integrated into the current hidden state h_t. It also takes the concatenation of the previous hidden state h_{t-1} and the current input x_t as inputs, enabling GRU cell to adaptively reset and update its internal representation based on the input at each time step. The reset gate r_t in GRU unit decides how much of the previous hidden state h_{t-1} should be forgotten or reset. It takes as input the concatenation of the previous hidden state h_{t-1} and the current input x_t, allowing GRU cell to determine the relevance of past information in the current context. The calculation of a standard GRU cell is expressed by the following formulae:

$$z_t = \text{sigmoid}\left(w_z \times x_t + u_z \times h_{t-1} + b_z\right) \tag{A11}$$

$$r_t = \text{sigmoid}\left(w_r \times x_t + u_r \times h_{t-1} + b_r\right) \tag{A12}$$

$$k_t = \tanh\left(w_k \times x_t + u_k \times \left(r_t \odot h_{t-1}\right) + b_k\right) \tag{A13}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot k_t \tag{A14}$$

where z_t and r_t represent the output data of update gate and reset gate at instant t, respectively; coefficients w_z, w_r, and w_k are the weights of update gate, reset gate, and temporary output, respectively; coefficients u_z, u_r, and u_k are the recurrent weights of update gate, reset gate, and temporary output, respectively; coefficients b_z, b_r, and b_k denote the biases of update gate, reset gate, and temporary output, respectively; and k_t is the candidate hidden state.

Acknowledgements This research was funded by the National Research Council of Thailand (NRCT) through Grant No. NRCT5-RSA63006. The authors also appreciate the financial support provided by King Mongkut's University of Technology Thonburi (KMUTT), Thailand Science Research and Innovation (TSRI), and National Science, Research and Innovation Fund (NSRF) Basic Research Fund: Fiscal year 2024 under project Advanced and Sustainable Construction Towards Thailand 4.0.

Author Contributions Sompote Youwai: conceptualization, methodology, software, validation, formal analysis, investigation, data curation. Praiya Ratanakjkul: conceptualization, methodology, software, validation, formal analysis, investigation, data curation. Chana Phutthananon: writing—original draft, visualization, funding acquisition. Warat Kongkitkul: writing—review and editing, project administration. Pornkasem Jongpradist: writing—review and editing, funding acquisition.

Data Availability Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Cheng X, Chen YH, Chen G, Li BY (2021) Characterization and prediction for the strength development of cement stabilized

dredged sediment. Mar Georesources Geotechnol 39(9):1015–1024. https://doi.org/10.1080/1064119X.2020.1795014
2. Yin KS, Xiao T, Luo HY, Zou HF, Zhang LM (2023) Probabilistic modeling of offshore deep cement mixing improved ground. Comput Geotech 156:105266. https://doi.org/10.1016/j.compgeo.2023.105266
3. Nguyen B, Takeyama T, Kitazume M (2016) Internal failure of deep mixing columns reinforced by a shallow stabilized soil beneath an embankment. Int J Geosynth Gr Eng 2:30. https://doi.org/10.1007/s40891-016-0072-4
4. Chai JC, Khan S, Shrestha S (2021) Behavior of an embankment with a reinforced slope on column improved clay deposits. Int J Geosynth Gr Eng 7:88. https://doi.org/10.1007/s40891-021-00331-5
5. Nguyen PLT, Tran MH, Tran TD, Nguyen BP (2023) Numerical analysis of arching behavior of geosynthetic-reinforced and DCM column-supported embankment with geosynthetic characteristics. Int J Geosynth Gr Eng 9:56. https://doi.org/10.1007/s40891-023-00474-7
6. Phutthananon C, Jongpradist P, Kandavorawong K, Dias D, Guo X, Jamsawang P (2023) Reliability assessment for serviceability limit states of stiffened deep cement mixing column-supported embankments. J Rock Mech Geotech Eng 15(9):2402–2422. https://doi.org/10.1016/j.jrmge.2023.05.008
7. Phutthananon C, Jongpradist P, Wonglert A, Kandavorawong K, Sanboonsiri S, Jamsawang P (2023) Field and 3D numerical investigations of the performances of stiffened deep cement mixing column-supported embankments built on soft soil. Arab J Sci Eng 48(4):5139–5169. https://doi.org/10.1007/s13369-022-07322-2
8. Waichita S, Jongpradist P, Jamsawang P (2019) Characterization of deep cement mixing wall behavior using wall-to-excavation shape factor. Tunn Undergr Sp Technol 83:243–253. https://doi.org/10.1016/j.tust.2018.09.033
9. Chen S, Guan Y, Dai J (2023) Behaviour of anchored sheet pile quay stabilized with deep cement mixing columns in soft soil: centrifuge and numerical modelling. Comput Geotech 160:105504. https://doi.org/10.1016/j.compgeo.2023.105504
10. Makararotrit W, Youwai S (2023) Cement admixed clay columns as a solution for river wall stability problems: a case study of Pasak river. Int J Geosynth Gr Eng 9:71. https://doi.org/10.1007/s40891-023-00490-7
11. Phutthananon C, Jongpradist P, Dias D, Jamsawang P (2021) Numerical study of the deformation performance and failure mechanisms of TDM pile-supported embankments. Transp Geotech 30:100623. https://doi.org/10.1016/j.trgeo.2021.100623
12. Phutthananon C, Jongpradist P, Jongpradist P, Dias D, Jamsawang P, Bergado DT (2021) Performance-based design optimization of embankments resting on soft soil improved with T-shaped and conventional DCM columns. Acta Geotech 16(10):3301–3326. https://doi.org/10.1007/s11440-021-01258-x
13. Phutthananon C, Jongpradist P, Dias D, Guo X, Jamsawang P, Baroth J (2022) Reliability-based settlement analysis of embankments over soft soils reinforced with T-shaped deep cement mixing piles. Front Struct Civ Eng 16(5):638–656. https://doi.org/10.1007/s11709-022-0825-1
14. Jamsawang P, Yoobanpot N, Thanasisathit N, Voottipruex P, Jongpradist P (2016) Three-dimensional numerical analysis of a DCM column-supported highway embankment. Comput Geotech 72:42–56. https://doi.org/10.1016/j.compgeo.2015.11.006
15. Fulambarkar S, Bangari G, Manna B, Shahu JT (2021) Numerical analysis of embankment built on Indian marine clay improved with DM column. Int J Geosynth Gr Eng 7:95. https://doi.org/10.1007/s40891-021-00342-2
16. Jamsawang P, Voottipruex P, Tanseng P, Jongpradist P (2019) Effectiveness of deep cement mixing walls with top-down construction for deep excavations in soft clay: case study and 3D simulation. Acta Geotech 14(1):225–246. https://doi.org/10.1007/s11440-018-0660-7
17. Choosrithong K, Schweiger HF (2020) Numerical investigation of sequential strut failure on performance of deep excavations in soft soil. Int J Geomech 20(6):04020063. https://doi.org/10.1061/(ASCE)GM.1943-5622.0001695
18. Choosrithong K, Schweiger HF, Marte R (2020) Finite element analysis of mixed-in-place columns (MIP) supporting excavations in slopes considering tension softening. Can Geotech J 57(6):785–800. https://doi.org/10.1139/cgj-2019-0093
19. Waichita S, Jongpradist P, Schweiger HF (2020) Numerical and experimental investigation of failure of a DCM-wall considering softening behaviour. Comput Geotech 119:103380. https://doi.org/10.1016/j.compgeo.2019.103380
20. Zhao L, Chen Y, Chen W, Wang J, Ren C (2023) The performance of T-shaped deep mixed soil cement column-supported embankments on soft ground. Constr Build Mater 369:130578. https://doi.org/10.1016/j.conbuildmat.2023.130578
21. Kasama K, Ochiai H, Yasufuku N (2000) On the stress–strain behaviour of lightly cemented clay based on an extended critical state concept. Soils Found 40(5):37–47
22. Liu MD, Carter JP (2002) A structured Cam Clay model. Can Geotech J 39(6):1313–1332. https://doi.org/10.1139/t02-069
23. Lee K, Chan D, Lam K (2004) Constitutive model for cement treated clay in a critical state frame work. Soils Found 44(3):69–77. https://doi.org/10.3208/sandf.44.3_69
24. Horpibulsuk S, Liu MD, Liyanapathirana DS, Suebsuk J (2010) Behaviour of cemented clay simulated via the theoretical framework of the Structured Cam Clay model. Comput Geotech 37(1–2):1–9. https://doi.org/10.1016/j.compgeo.2009.06.007
25. Vatsala A, Nova R, Sirinivasa Murthy BR (2001) Elastoplastic model for cemented soils. J Geotech Geoenviron Eng 127:679–687. https://doi.org/10.1061/(ASCE)1090-0241(2001)127:8(679)
26. Suebsuk J, Horpibulsuk S, Liu MD (2010) Modified Structured Cam Clay: a generalised critical state model for destructured, naturally structured and artificially structured clays. Comput Geotech 37(7–8):956–968. https://doi.org/10.1016/j.compgeo.2010.08.002
27. Nguyen L, Fatahi B, Khabbaz H (2014) A constitutive model for cemented clays capturing cementation degradation. Int J Plast 56:1–18. https://doi.org/10.1016/j.ijplas.2014.01.007
28. Nguyen L, Fatahi B, Khabbaz H (2017) Development of a constitutive model to predict the behavior of cement-treated clay during cementation degradation: C3 model. Int J Geomech 17(7):04017010. https://doi.org/10.1061/(ASCE)GM.1943-5622.0000863
29. Xiao H, Lee FH, Liu Y (2017) Bounding surface Cam-Clay model with cohesion for cement-admixed clay. Int J Geomech 17(1):04016026. https://doi.org/10.1061/(ASCE)GM.1943-5622.0000671
30. Zhang N, Zhou A, Zhen-yu J, Yin YF, Shen SL (2023) An enhanced deep learning method for accurate and robust modelling of soil stress–strain response. Acta Geotech. https://doi.org/10.1007/s11440-023-01813-8
31. Zhang N, Shen SL, Zhou A, Jin YF (2021) Application of LSTM approach for modelling stress–strain behaviour of soil. Appl Soft Comput 100:106959. https://doi.org/10.1016/j.asoc.2020.106959
32. Raja MNA, Shukla SK (2021) Predicting the settlement of geosynthetic-reinforced soil foundations using evolutionary artificial intelligence technique. Geotext Geomembranes 49(5):1280–1293. https://doi.org/10.1016/j.geotexmem.2021.04.007
33. Raja MNA, Abbas Jaffar ST, Bardhan A, Shukla SK (2023) Predicting and validating the load-settlement behavior of large-scale geosynthetic-reinforced soil abutments using hybrid intelligent modeling. J Rock Mech Geotech Eng 15(3):773–788. https://doi.org/10.1016/j.jrmge.2022.04.012

34. Raja MNA, Abdoun T, El-Sekelly W (2023) Smart prediction of liquefaction-induced lateral spreading. J Rock Mech Geotech Eng. https://doi.org/10.1016/j.jrmge.2023.05.017
35. Mojtahedi FF, Ahmadihosseini A, Eidgahee DR, Rezaee M, Spagnoli G (2024) Bio-inspired predictive models development for strength characterization of cement deep-mixed plastic soils. Int J Geosynth Gr Eng 10(1):9. https://doi.org/10.1007/s40891-023-00508-0
36. Ghaboussi J, Garrett JH Jr, Wu X (1991) Knowledge-based modeling of material behavior with neural networks. J Eng Mech 117(1):132–153. https://doi.org/10.1061/(ASCE)0733-9399(1991)117:1(132)
37. Ellis GW, Yao C, Zhao R, Penumadu D (1995) Stress–strain modeling of sands using artificial neural networks. J Geotech Eng 121(5):429–435. https://doi.org/10.1061/(ASCE)0733-9410(1995)121:5(429)
38. Penumadu D, Zhao R (1999) Triaxial compression behavior of sand and gravel using artificial neural networks (ANN). Comput Geotech 24(3):207–230. https://doi.org/10.1016/S0266-352X(99)00002-6
39. Zhang W, Li H, Li Y, Liu H, Chen Y, Ding X (2021) Application of deep learning algorithms in geotechnical engineering: a short critical review. Artif Intell Rev 54(8):5633–5673. https://doi.org/10.1007/s10462-021-09967-1
40. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536. https://doi.org/10.1038/323533a0
41. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
42. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18(5–6):602–610. https://doi.org/10.1016/j.neunet.2005.06.042
43. Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schmidhuber J (2017) LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst 28(10):2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924
44. Cho K, van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder–decoder approaches. arXiv preprint. arXiv:1409.1259. https://doi.org/10.48550/arXiv.1409.1259
45. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint :arXiv:1406.1078. https://doi.org/10.48550/arXiv.1406.1078
46. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint :arXiv:1412.3555. https://doi.org/10.48550/arXiv.1412.3555
47. Zhang P, Yin ZY, Jin YF, Ye GL (2020) An AI-based model for describing cyclic characteristics of granular materials. Int J Numer Anal Methods Geomech 44(9):1315–1335. https://doi.org/10.1002/nag.3063
48. Shi LL, Zhang J, Zhu QZ, Sun HH (2022) Prediction of mechanical behavior of rocks with strong strain-softening effects by a deep-learning approach. Comput Geotech 152:105040. https://doi.org/10.1016/j.compgeo.2022.105040
49. Werbos PJ (1988) Generalization of backpropagation with application to a recurrent gas market model. Neural Netw 1(4):339–356. https://doi.org/10.1016/0893-6080(88)90007-X
50. Williams RJ, Zipser D (1989) A learning algorithm for continually running fully recurrent neural networks. Neural Comput 1(2):270–280. https://doi.org/10.1109/ijcnn.1989.118645
51. 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(5):3661–3686. https://doi.org/10.1007/s11831-020-09524-z
52. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166. https://doi.org/10.1109/72.279181
53. Lin SS, Shen SL, Zhang N, Zhou A (2021) Modelling the performance of EPB shield tunnelling using machine and deep learning algorithms. Geosci Front 12(5):101177. https://doi.org/10.1016/j.gsf.2021.101177
54. Chen X, Yang J, He G, Huang L (2023) Development of an LSTM-based model for predicting the long-term settlement of land reclamation and a GUI-based tool. Acta Geotech. https://doi.org/10.1007/s11440-022-01749-5
55. Wang L, Wu C, Yang Z, Wang L (2023) Deep learning methods for time-dependent reliability analysis of reservoir slopes in spatially variable soils. Comput Geotech 159:105413. https://doi.org/10.1016/j.compgeo.2023.105413
56. Xu Q, Huang X, Zhang B, Zhang Z, Wang J, Wang S (2023) TBM performance prediction using LSTM-based hybrid neural network model: case study of Baimang River Tunnel Project in Shenzhen, China. Undergr Sp. https://doi.org/10.1016/j.undsp.2022.11.002
57. Zhang N, Zhang N, Zheng Q, Xu YS (2022) Real-time prediction of shield moving trajectory during tunnelling using GRU deep neural network. Acta Geotech 17(4):1167–1182. https://doi.org/10.1007/s11440-021-01319-1
58. Yang J, Liu Y, Yagiz S, Laouafa F (2021) An intelligent procedure for updating deformation prediction of braced excavation in clay using gated recurrent unit neural networks. J Rock Mech Geotech Eng 13(6):1485–1499. https://doi.org/10.1016/j.jrmge.2021.07.011
59. Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. Neural Comput 12(10):2451–2471. https://doi.org/10.1385/ABAB:94:2:127
60. Jongpradist P, Youwai S, Jaturapitakkul C (2011) Effective void ratio for assessing the mechanical properties of cement-clay admixtures at high water content. J Geotech Geoenviron Eng 137(6):621–627. https://doi.org/10.1061/(ASCE)GT.1943-5606.0000462
61. Phutthananon C, Jongpradist P, Nakin S, Youwai S, Hajiazizi M, Jamsawang P (2023) State parameter governing the mechanical properties of cement-treated clays. Mar Georesources Geotechnol 41(4):388–399. https://doi.org/10.1080/1064119X.2022.2049935
62. Phutthananon C, Tippracha N, Jongpradist P, Tunsakul J, Tangchirapat W, Jamsawang P (2023) Investigation of strength and microstructural characteristics of blended cement-admixed clay with bottom ash. Sustainability 15(4):3795. https://doi.org/10.3390/su15043795
63. ASTM (2020) Standard test method for consolidated undrained triaxial compression test for cohesive soils. ASTM D4767–11. West Conshohocken, PA: ASTM.
64. TensorFlow Developers (2023) TensorFlow. In: Zenodo. https://doi.org/10.5281/ZENODO.4724125
65. Chollet F (2015) Keras. In: GitHub. https://github.com/keras-team/keras
66. Miura N, Horpibulsuk S, Nagaraj TS (2001) Engineering behavior of cement stabilized clay at high water content. Soils Found 41(5):33–45. https://doi.org/10.3208/sandf.41.5_33
67. Lorenzo GA, Bergado DT (2004) Fundamental parameters of cement-admixed clay—new approach. J Geotech Geoenviron Eng 130(10):1042–1050. https://doi.org/10.1061/(ASCE)1090-0241(2004)130
68. Zhang RJ, Zheng JJ, Bian XY (2017) Experimental investigation on effect of curing stress on the strength of cement-stabilized clay

at high water content. Acta Geotech 12(4):921–936. https://doi.org/10.1007/s11440-016-0511-3
69. Lin SS, Zhang N, Zhou A, Shen SL (2022) Time-series prediction of shield movement performance during tunneling based on hybrid model. Tunn Undergr Sp Technol 119:104245. https://doi.org/10.1016/j.tust.2021.104245
70. Elbaz K, Yan T, Zhou A, Shen SL (2022) Deep learning analysis for energy consumption of shield tunneling machine drive system. Tunn Undergr Sp Technol 123:104405. https://doi.org/10.1016/j.tust.2022.104405
71. Jung Y, Hu J (2015) A K-fold averaging cross-validation procedure. J Nonparametr Stat 27(2):167–179. https://doi.org/10.1080/10485252.2015.1010532
72. Cahuantzi R, Chen X, Güttel S (2021) A comparison of LSTM and GRU networks for learning symbolic sequences. arXiv preprint. arXiv:2107.02248. https://doi.org/10.48550/arXiv.2107.02248
73. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. arXiv preprint. arXiv:1706.03762. https://doi.org/10.48550/arXiv.1706.03762

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.