Predicting soil stress–strain behaviour with bidirectional long short-term memory networks

Kacper Cerek
Institute of Geotechnical Engineering and Construction Management, Hamburg University of Technology, Hamburg, Germany
Arjun Gupta
Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, Canada, and
Duy Anh Dao, Elnaz Hadjiloo and Jürgen Grabe
Institute of Geotechnical Engineering and Construction Management, Hamburg University of Technology, Hamburg, Germany

Abstract

Purpose – Artificial intelligence, particularly deep learning (DL), has increasingly influenced various scientific fields, including soil mechanics. This paper aims to present a novel DL application of long short-term memory (LSTM) networks for predicting soil behaviour during constant rate of strain (CRS) tests.

Design/methodology/approach – LSTMs are adept at capturing long-term dependencies in sequential data, making them suitable for predicting the complex, nonlinear stress–strain behaviour of soil. This paper evaluates various LSTM configurations, optimising parameters such as step size, batch size, data sampling rate and training subset size to balance prediction accuracy and computational efficiency. The study uses a comprehensive data set from numerical finite element method simulations conducted with PLAXIS 2D and laboratory CRS tests.

Findings – The proposed LSTM model, trained on data at lower stress levels, accurately forecasts soil behaviour at higher stress levels. The optimal LSTM setup achieved a median error of 3.59% and 5.10% for numerical data and 3.86% for laboratory data, presenting the setup's effectiveness.

Originality/value – This approach reduces the required time to complete extensive laboratory testing, aligning with sustainable industrial practices. The findings suggest that LSTM networks can enhance geotechnical engineering applications by efficiently predicting soil behaviour.

Keywords Neural networks, Artificial intelligence, Laboratory tests, Computational geotechnics, Sustainable development

Paper type Research paper

List of notations

u_1 = axial displacement in CRS element test;
u_3 = radial displacement in CRS element test;
ε_1 = axial strain in CRS test;
ε_3 = radial strain in CRS test;
σ_1 = axial stress in CRS test;
H_{sample} = initial soil sample height;
h_s = granulate hardness in the hypoplastic constitutive model;
n = exponent n in the hypoplastic constitutive model;
e_0 = initial void ratio in CRS test;
e_{end} = final void ratio at the end of CRS test;
$\Delta\varepsilon_i$ = uniform strain step;
Ω_{train} = training subset;
Ω_{predict} = predicted subset;
Ω_{test} = testing subset;
N_{step} = step size;
B_k = batch size;
f_s = data sampling rate;
$x_{\text{train},i}$ = input subsequence for supervised learning derived from Ω_{train};
$y_{\text{train},i}$ = output label for supervised learning derived from Ω_{train};
t_{run} = runtime of the training process;
$\widehat{y}_i$ = predicted value of data point in Ω_{predict};
y_i = actual value of data point in Ω_{test}; and
$\widetilde{e}$ = median value of percentage errors.

1. Introduction

The application of artificial intelligence in all fields of life has been rapidly growing in recent years (Mian, 2022). In particular, the application of deep learning (DL) (LeCun *et al.*, 2015) has significantly contributed to the development of complex fields of science, including material science (Choudhary *et al.*, 2022), mathematics (Davies *et al.*, 2021),

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/3029-0414.htm**

Machine Learning and Data Science in Geotechnics
1/1 (2025) 59–76
[ISSN 3029-0414]
[DOI 10.1108/MLAG-08-2024-0007]

Received 22 August 2024
Revised 3 January 2025
4 April 2025
Accepted 21 April 2025

image segmentation in medicine (Pepe *et al.*, 2020) and robot navigation using neural radiance fields (NeRF) (Adamkiewicz *et al.*, 2022). One of the biggest advantages of DL is its ability to capture complex, often nonlinear dependencies between input and output data (Chen and Billings, 1992). Considering the highly nonlinear stress–strain behaviour of soil (Atkinson, 2000), the field of soil mechanics could significantly benefit from the application of DL techniques.

Efforts have been made to approximate the nonlinear behaviour of soil using optimisation methods to calibrate soil parameters for constitutive models (Machaček *et al.*, 2022; Mendez *et al.*, 2021). Similarly, evolutionary algorithms have proven to be valuable tools for structural optimisation (Pucker and Grabe, 2011) and for enhancing the design process, as demonstrated by Cerek and Grabe (2023) and Cerek *et al.* (2024d, 2024e). Additionally, these algorithms have potential applications in inverse problems (Grabe *et al.*, 2010). However, the effectiveness of those approaches is constrained by the accuracy of the equations governing the soil model. Evolutionary polynomial regression (EPR) has been beneficial in various geotechnical applications (Rezania, 2008; Rezania *et al.*, 2008), especially when the data is expressed as a polynomial. Although EPR does not depend on constitutive models, identifying a polynomial equation that accurately fits the data can be computationally intensive.

Advancements in machine learning (ML) and DL have significantly impacted geotechnical engineering and soil science. Feedforward neural networks (FNNs) have been used for soil classification (Levine *et al.*, 1996), generating stiffness degradation curve from basic soil classification data (Charles and Gourvenec, 2024), interpretation of cone penetration test (CPT) data (Rauter and Tschuchnigg, 2021; Erzin and Ecemis, 2017), predicting pavement surface condition (Deneko *et al.*, 2024) and finding correlations between numerical simulations and soil properties (Irani *et al.*, 2024). Convolutional neural networks (CNNs) have advanced soil classification techniques (Kiran Pandiri *et al.*, 2024; Hamzah and Ronaldo, 2021; Tatar *et al.*, 2024), monitoring soil cracks (Xu *et al.*, 2022) and performing spatial analysis of geotechnical properties (Chen *et al.*, 2024). Recurrent neural networks (RNNs) and FNNs have been applied to model soil behaviour (Ellis *et al.*, 1995; Zhu *et al.*, 1998a, 1998b), and physics-informed data-driven modelling has been used for consolidation analysis (Zhang *et al.*, 2024).

FNNs are designed for static input–output relationships, while CNNs excel in tasks involving spatial hierarchies, patterns and image recognition. However, both FNNs and CNNs are limited in their ability to capture dependencies in sequential or time-dependent data, e.g. soil settlement and consolidation, which are frequently encountered in geotechnical engineering. Traditional RNNs struggle to maintain information over long sequences, reducing their effectiveness in modelling processes requiring long-term interactions. By contrast, long short-term memory (LSTM) neural networks, a specific type of RNNs, were designed to capture and maintain long-term patterns (Hochreiter, 1991).

Although the application of LSTMs in geotechnical engineering has been increasing over the past few years, FNNs remain the network of primary choice (Liu *et al.*, 2024). Gao *et al.* (2021) applied LSTMs to predict tunnel boring machine performance, while Qi and Fourie (2018) used them in rock engineering for estimation of rheological parameters. Xu and Niu (2018) used LSTMs to forecast landslides. Guan *et al.* (2023) conducted a finite element geotechnical analysis incorporating a DL-based soil model using LSTM. Building on these advancements, Cerek *et al.* (2024a) proposed an approach for predicting data points related to the stress–strain behaviour of soil. They incorporated numerically generated compression curves from constant rate of strain (CRS) tests using a hardening soil model. Zhang *et al.* (2021) investigated the performance of LSTMs for modelling the stress–strain behaviour of soil through a neural network-based constitutive model, leveraging a relatively small test group, consisting of 127 samples of numerical and 29 samples of experimental data under triaxial conditions. Their study highlighted the superiority of LSTMs over FNNs in capturing the nonlinear behaviour of soil. The method of Zhang *et al.* (2021) requires prior knowledge of at least one soil parameter, axial strain levels and the applied confining pressure while predicting the corresponding deviatoric stress as the output. Their neural network architecture combines fully connected dense layers with an LSTM layer, with a fixed number of iterations applied during the training process.

In this paper, we propose a novel approach for predicting soil behaviour during CRS test using LSTM. Our goal is to develop a more generalised method that does not rely on specific preconditions, unlike the approach presented in Zhang *et al.* (2021). Additionally, we implemented early stopping criteria to optimise the training process and used a substantially larger and more diverse numerical data set to evaluate the robustness of the proposed LSTM network. Various configurations are examined to identify the best-performing architecture of LSTM based on trade-offs between runtime and prediction error metrics. Throughout the study, key parameters such as the number of steps N_{step}, training subset Ω_{train}, data sampling rate f_s and batch size B_k are systematically varied. Leveraging the LSTM's ability to capture nonlinear patterns, we train the neural network to predict future soil responses during CRS tests. Initially, the approach is tested on numerical data generated through finite element method (FEM) element tests, using a highly nonlinear hypoplastic soil model. Subsequently, it is validated using results derived from laboratory tests. By applying LSTM networks, which are effective in capturing temporal dependencies in sequential data, we demonstrate the potential to accurately predict unseen stress–strain behaviour of unsaturated coarse-grained soil using only data at lower stress levels derived from CRS test.

The main aim of this study is to investigate the capabilities and limitations of LSTM in accurately predicting the stress–strain behaviour of soil at higher stress levels based only on CRS laboratory test results at lower stress levels to reduce the time and costs associated with conducting these tests. By implementing LSTM in geotechnical engineering, the study aims to contribute to the optimisation of industrial processes and resource utilisation, aligning with the United Nations Sustainable Development Goal 9 – Industry, Innovation and Infrastructure (United Nations, 2015). This novel approach has the potential to minimise the resources required for extensive laboratory testing, thereby supporting sustainable industrial practices. For instance, compression tests for soft, fine-grained soils, which can take several days, might be optimised by using the presented approach.

In the following, Sections 2 and 3 present the theoretical background and methodology of the approach. Section 4 elaborates on the results obtained from the conducted studies. Section 5 concludes the paper by summarising the findings, addressing limitations and offering suggestions for future research and applications.

2. Theorical background

2.1 Constant rate of strain test

In a CRS test, a cylindrical sample of soil is compressed uniaxially at a constant strain rate, calculated relative to the initial height of soil sample H_{sample}, down to 0.01%/min for very soft clays until a predefined axial strain ε_1 is reached. Radial strain ε_3 is restricted by a steel ring, while the resulting axial stress σ_1 is constantly monitored (Smith and Wahls, 1969). CRS tests primarily provide information about compression properties of tested soil by delivering a stress–strain σ_1-ε_1 curve. Parameters, such as the granulate hardness h_s or the exponent n (Niemunis and Herle, 1997), derived from CRS tests and together with other hypoplastic parameters are used to predict large deformations in soil–structure interaction problems (Bienen *et al.*, 2021; Stapelfeldt *et al.*, 2020; Dao *et al.*, 2023).

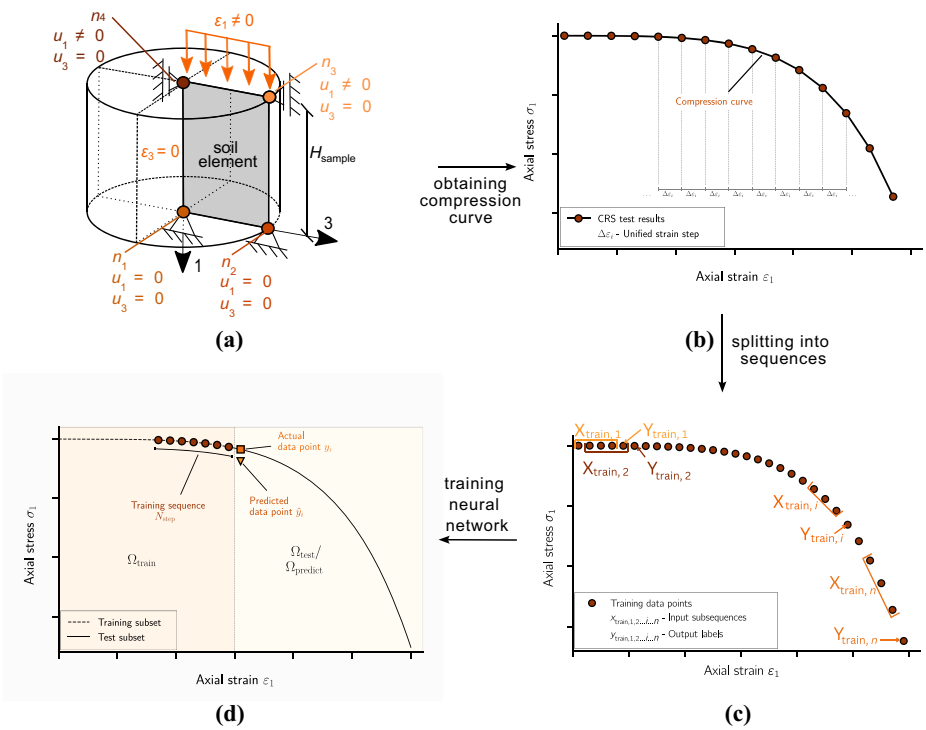
This paper incorporates an extensive open-source database compiled by Cerek *et al.* (2024b), which includes results of 2D axisymmetric, displacement-controlled element tests conducted with FEM. Dirichlet boundary conditions were applied to allow only axial displacement u_1 of the top edge nodes n_3 and n_4, where the axial strain ε_1 is imposed at a constant rate. No lateral displacement u_3 is allowed. Figure 1 schematically illustrates the methodology of this paper, including the setup of a CRS element test, as detailed by Cerek *et al.* (2024a). The element test simulations were conducted using a hypoplastic constitutive model, which can reproduce the pressure- and density-dependent stiffness of the soil. This model was initially developed by von Wolffersdorff (1996) and later extended through the intergranular strain concept by Niemunis and Herle (1997).

2.2 Data collection

This study uses stress–strain σ_1-ε_1 compression curves derived from either numerically performed CRS element tests (TS1) and physically conducted CRS laboratory tests (TS2) on coarse-grained, unsaturated soil. These compression curves are used for the training and testing an LSTM neural network.

The databank TS1 comprises of 10,000 entries, where various combinations of hypoplastic parameters were simulated (Cerek *et al.*, 2024b). The ranges for the parameter combinations were selected based on real values found in the literature (Herle and Gudehus, 1999). Each data set consists of 3,000 data points, depicting the axial stress–strain behaviour (σ_1-ε_1) until the 30% of the axial strain ε_1 is reached. The strain rate was kept constant during the conducted numerical CRS test. For this study, a test group of 100 data sets, denoted as

Figure 1 Methodology for generating and preprocessing datasets, followed by training an LSTM network. (a) Setup of FEM element test with boundary conditions, (b) stress–strain (σ_1-ε_1) compression curve with unified strain step $\Delta\varepsilon_i$, (c) splitting of data points into sequences for the training process and (d) training and using the LSTM network to predict unseen data points $\hat{y}_i$ and compare them with actual data points y_i



Source: Figure by authors

TS1.1, was randomly selected from the databank TS1 to test different configurations of the LSTM neural network. Following this, 1,983 data sets, denoted as TS1.2, were randomly selected from TS1 and used to evaluate the performance of the best-performing LSTM setup on a larger test group. The number of data sets in TS1.2 was constrained by the available computational resources. The FEM numerical simulations were conducted with PLAXIS (2024) 2D.

The best-performing LSTM configuration is subsequently applied to various laboratory results of CRS tests to validate the proposed approach on real-life data. A set of eight compression curves (TS2) for various coarse-grained soil was selected from Dao and Banduch (2023) and further tests from the geotechnical laboratory at Hamburg University of Technology. The maximum axial strain ε_1 applied to the soil samples is below 8%, though the exact maximum ε_1 and the applied rate of strain varies among the tests. Table A2 in Appendix 2 presents the complete values of the maximum axial strain ε_1, maximum axial stress σ_1, initial void ratio e_0 and final void ratio e_{end} at the end of the CRS test, for all soil samples S1 to S8 in test group TS2.

2.3 Long short-term memory neural network

LSTM is a specialised type of RNN designed for efficient handling of sequential data (Hochreiter and Schmidhuber, 1997). Its ability to capture long-term patterns makes it suitable for various tasks such as time-series forecasting (Sagheer and Kotb, 2019), language modelling (Merity *et al.*, 2017) and speech recognition (Sak *et al.*, 2014). Siami-Namini *et al.* (2018) demonstrated the superiority of LSTMs in time-series forecasting compared to other approaches like ARIMA.

RNN models commonly experience two issues during backpropagation known as exponential gradient and vanishing gradient. A gradient is the partial derivative of the loss function with respect to the model parameters, which guides updates during training. The gradient applied to each parameter is scaled by a value held in the model's weight matrices. These problems arise when the backpropagated gradients either grow exponentially large (exploding gradient) or diminish to near zero (vanishing gradient), making it difficult for the model to learn effectively (Staudemeyer and Morris, 2019). The vanishing gradient problem, in particular, leads to the RNN's failure to capture long-term dependencies in the data. LSTMs were developed to overcome RNN's tendency to scale error signals exponentially, by using constant error carousel (CEC), which eliminates scaling in the backpropagation step of the learning sequence by enforcing the weight of the feedback signal to be equal to 1. This allows for the ability to learn from over 1,000 discrete steps, making LSTMs significantly more effective than traditional RNN models, which can incorporate approximately ten previous time steps (Hochreiter, 1991). Therefore, LSTMs were selected for this study because of their greater flexibility and robustness compared to classic RNNs, regardless of the number of data points incorporated for prediction.

The core of an LSTM network lies in its capacity to leverage historical data to forecast future data points through an iterative training process known as an epoch. A neural network is structured into three primary sequential layers: the input layer, the hidden layer and the output layer (Sak *et al.*, 2014). The input layer is responsible for receiving the sequential data and can be defined explicitly as standalone layer or implicitly within the first hidden layer. Typically, the input data is organised as a two-dimensional array, where the dimensions represent the step size N_{step}, and number of features, which indicates the quantity of variables at each step included in the LSTM model. The recurrent hidden layer, consisting of LSTM units, processes the sequential data and captures long-term dependencies and temporal patterns within it. Each LSTM unit is comprised of three gates (forget, input and output), which regulate the flow of information, and LSTM cells, which stores long-term information required to complete predictive calculations. The gates consist of neurons that use an activation function to control the flow of information (Sak *et al.*, 2014). Figure 2 provides a schematic representation of an LSTM unit integrated into the hidden layer. During the iterative learning process, the hidden layer computes equations (1)–(6). The calculated input, forget, and output gate values are represented by i_t, f_t and o_t, respectively, at data step t. $\overline{C}_t$ is the hidden candidate state, C_t is the internal memory of the unit and h is the output of the unit that acts as the input to nodes of the next hidden layer. U represents the weight matrix that connects the inputs to the hidden layer, while W represents the weighted connection between different layers, b is a calculated bias vector and x represents the input from the provided data set (Greff *et al.*, 2017). The logistic sigmoid is used as gate activation function, and the rectified linear unit (ReLU) is used as unit input and output activation function (Dubey *et al.*, 2022). The output layer of the neural network typically comprises of a fully connected dense layer, which converts the last hidden state of the LSTM units into the final prediction o_x by using a linear activation function, a bias b and the weight matrix W, as given in equation (7).

Equations completed by the hidden layer:

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i + b_i\right) \tag{1}$$

$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f + b_f\right) \tag{2}$$

$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o + b_o\right) \tag{3}$$

$$\overline{C}_t = ReLU\left(x_t U^c + h_{t-1} W^c + b_c\right) \tag{4}$$

$$C_t = \sigma\left(f_t C_{t-1} + i_{t\overline{C}_t}\right) \tag{5}$$

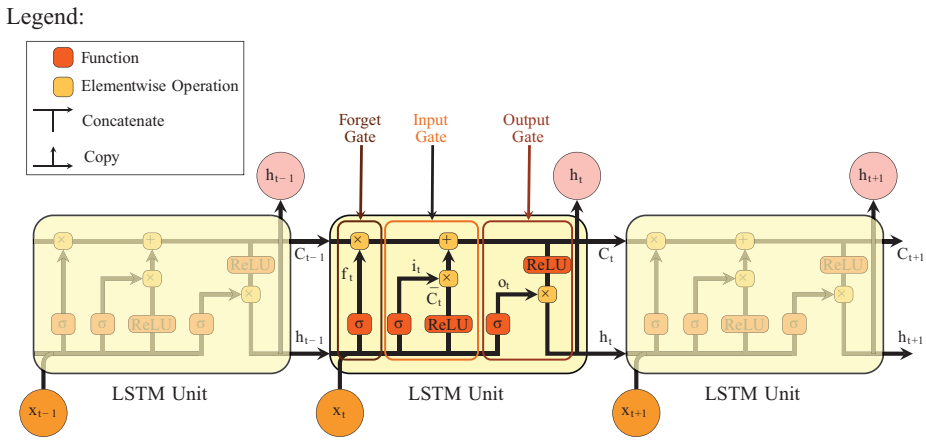
$$h_t = o_t \cdot ReLU(C_t) \tag{6}$$

Equation completed by the output layer:

$$o_x = Wx + b \tag{7}$$

The neural network used in this study uses a variant of LSTM known as bidirectional, introduced by Schuster and Paliwal (1997). The bidirectional LSTM is known to be more effective than unidirectional LSTMs and well suited to tasks, where context is vitally important (Graves and Schmidhuber, 2005). This approach involves analysing the sequence in both forward

Figure 2 Schematic representation of a LSTM unit integrated into the hidden layer



Source: Figure by authors

and backward directions, with separate layers dedicated to processing data in each direction. The output of these layers is subsequently combined through concatenation.

3. Methodology

This research uses stress–strain curves obtained from both numerical and physical CRS experiments across three test groups TS1.1, TS1.2 and TS2. The sequential data sets, consisting of axial stress values σ_1, are preprocessed to align with the format expected by the LSTM model and then input into the neural network for the supervised learning process. The iterative training process involves a subset of the data set at lower stress levels Ω_{train} and continues until a specified convergence criterion, in this case, a minimal change in loss, is achieved. The trained LSTM network is then used to predict future values at higher stress levels. Finally, the predicted values $\Omega_{predict}$ are compared with corresponding unseen testing data points Ω_{test}, and the results are evaluated. The code used in this study has been published as open source (Cerek *et al.*, 2024c). Figure 3 presents the general computational process of the code.

This paper aims to identify the best-performing setup of LSTM network by investigating various combinations of parameters relevant to the training process. The step size N_{step} represents the number of data points considered when predicting future values. The batch size B_k is a hyperparameter that determines the number of sequences processed in parallel. The data sampling rate f_s specifies that every n^{th} data point from the training subset Ω_{train} is selected for input into the LSTM. The study explores 108 different parameter combinations for the test group TS1.1 by altering the following parameters within defined sets:

- the step size $N_{step} = \{10, 20, 50\}$;
- the batch size $B_k = \{5, 10, 20, 50\}$;
- the training subset $\Omega_{train} = \{60\%, 70\%, 80\%\}$; and
- the data sampling rate $f_s = \{2, 3, 4\}$.

Subsequently, the best-performing setup is identified based on trade-offs between runtime and prediction error metrics. This LSTM architecture is tested on a larger data set TS1.2 and validated against laboratory results TS2.

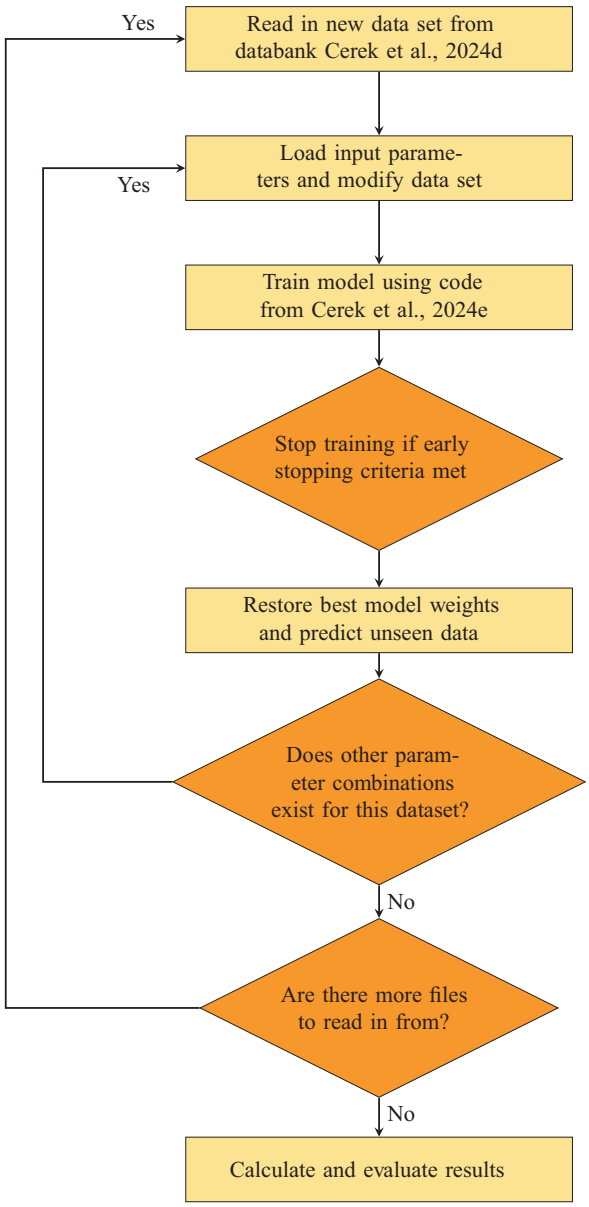
3.1 Preprocessing

The LSTM neural network used in this study uses only a single feature for training: the axial stress σ_1, with the axial strain ε_1 treated as a data step. Thus, the LSTM model is provided solely with axial stress levels σ_1. Therefore, the data points must be uniformly spaced in terms of strain steps $\Delta\varepsilon_i$, as seen in the numerical test groups TS1.1 and TS1.2. The data sets in TS2 were derived from the laboratory tests interpolated to ensure 1,000 uniform strain steps $\Delta\varepsilon_i$ to leverage the LSTM's ability to capture long-term patterns. Since each training process was conducted for a separate data set, truncation and padding techniques were not used. It is important to note that neither normalisation nor standardisation of data sets was used. Although data scaling is known to enhance the efficiency of the training process (Sinsomboonthong, 2022), its implementation would have unnecessarily complicated the neural network's application for predicting future data points due to the need for extrapolation of scaled points into the future. This extrapolation introduces potential inaccuracies and operational complexities, especially when future data distributions differ from the training data. The provided subsets Ω_{train} were segmented into suitable subsequences by creating input–output pairs for supervised learning. The length of the input subsequence $x_{train,i}$ reflects the number of steps N_{step}, whereas the output label $y_{train,i}$ corresponds to the value immediately following each $x_{train,i}$, as illustrated in Figure 1.

3.2 Architecture of long short-term memory network

The neural network architecture used in this study is a sequential model. It features three bidirectional LSTM layers, each consisting of 25 neurons. These hidden layers process the sequences from the implicit input layer in both forward and backward directions, leveraging the ReLU activation function. The network concludes with a dense output layer with a single neuron, using a linear activation function to produce the final prediction. Figure 4 illustrates the architecture of used LSTM

Figure 3 Flowchart of the Python code for the training of LSTM neural network



Source: Figure by authors

neural network. The number of layers and neurons was determined based on preliminary studies exploring various architectures. These studies investigated and evaluated combinations of one to five layers and 5–105 neurons (in increments of 20) based on prediction error metrics and computational cost. Similarly, the activation functions were selected based on their performance in these initial evaluations, which included functions such as softmax, sigmoid, tanh and the selected ReLU activation function. The model is compiled with the Adam optimiser because of its efficiency with large data sets (Kingma and Ba, 2014), using mean squared error (MSE) as the loss function and root mean squared error (RMSE) (Liano, 1996) as an additional performance metric. To avoid overfitting, early stopping is implemented, monitoring the training loss with a patience of 500 epochs, which halts training if there is no improvement in loss over this period and restores the best weights found (Prechelt, 2002). Training is configured to run for a maximum of 5,000 epochs, although it may terminate earlier if early stopping criteria are met.

The neural network undergoes supervised learning, where it compares its predicted axial stress values σ_1 to the actual values and calculates its error using the loss function. The weights in the hidden layers are updated between epochs based on this loss and the learning rate. The learning rate, a critical hyperparameter, controls the extent of weight adjustments, with smaller rates typically preferred to ensure system stability. In this study, the default learning rate of 0.001 (–) was used (Chollet *et al.*, 2015).

The trained LSTM network is deployed for forecasting unseen values of the axial stress σ_1, which were not passed to the neural network during training, based on the patterns it learned during training. The last input subsequence $x_{\text{train},i}$ from Ω_{train} is used for predicting a future label $\widehat{y}_i$, which is added to the subset with predictions Ω_{predict} and rolled into the input subsequence $x_{\text{train},i+1}$, effectively creating a sliding window of inputs for generating the next prediction $\widehat{y}_{i+1}$.

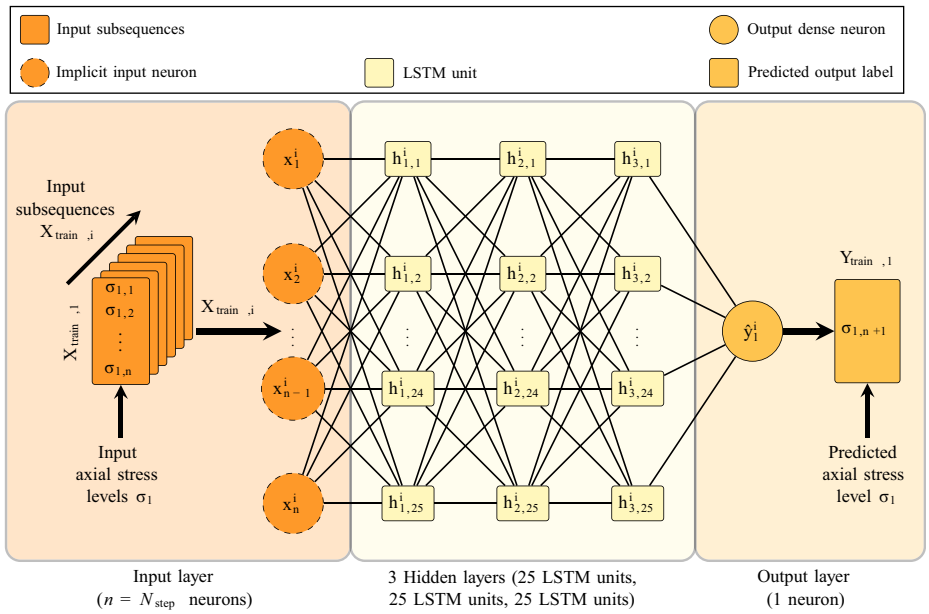
The LSTM neural network was developed and executed using Python (Van Rossum and Drake, 2009), leveraging the TensorFlow framework (Abadi *et al.*, 2015) and the Keras API (Chollet *et al.*, 2015). Computational tasks were performed on a high-performance computing (HPC) cluster at TUHH. The used HPC node is equipped with two Intel Xeon Gold 5318Y CPU, each running at 2.10 GHz, 512 GB of RAM and running on AlmaLinux OS 8. A total of 32 CPU cores were used for running the training processes.

3.3 Postprocessing of results

The architecture of the LSTM network was complemented with postprocessing scripts designed to save and analyse the obtained results, including predicted values Ω_{predict} and the runtime of the training process t_{run}. The identification of the best-performing LSTM network setup for the test group TS1.1 involves analysing the trade-offs between the runtime t_{run} and the median percentage error value $\widetilde{e}$ derived from all error values for each predicted compression curve from a test group, as outlined in equation (8). For each forecasted data point $\widehat{y}_i$ from Ω_{predict}, the corresponding actual value y_i from the test subset Ω_{test} is subtracted, normalised and converted to a percentage. The median of errors was chosen over the average value for evaluation to minimise the influence of outliers and the effects of non-symmetric distribution of results. Given the high magnitude of axial stress σ_1 and the absence of data normalisation in this study, the percentage-based error metrics leads to easier handling and final evaluation of the results.

It is essential to acknowledge that some of the training processes produce not-a-number (NaN) results, which cannot be considered in the study and therefore excluded from evaluation. There may be several root causes, including code-relevant bugs and code-irrelevant bugs (Wang *et al.*, 2023). This matter is not further discussed in this paper. If any of the percentage error values for a compression curve exceed 200%, the training process is classified as an outlier and excluded from

Figure 4 Architecture of the LSTM neural network used in this study



Source: Figure by authors

the evaluation to avoid overly biasing the median error value. The number of skipped files due to NaN results or classification as outliers is tracked and included in the results.

Median value of error:

$$\tilde{e} = med\left(\left\{\frac{\hat{y}_i - y_i}{y_i} \cdot 100\right\}_{i=1}^{n}\right) \quad (8)$$

4. Results

4.1 Numerical data test group TS1.1

The goal of the tests conducted with the TS1.1 test group, which consists of numerically generated data, was to identify the best-performing parameter combination for the LSTM network. The code was designed to iterate through four nested loops, allowing for all combinations of the percentage of the data set used for training Ω_{train}, the sampling rate f_s, the step size N_{step} and the batch size B_k to be tested. These parameters were then individually reviewed to determine their discrete effects on error and runtime.

Figure 5 illustrates the performance metrics, specifically the median error value $\tilde{e}$ and runtime t_{run}, for different percentages of the data set used for training Ω_{train}. The data reveals a strong correlation between the percentage of data set and the median error value. When 80% of the data set is used for training, the median error value hovers around 5%, indicating a significant improvement over the 70% training subset, where the majority of errors exceed 10%. Using 60% of the data set for training does not yield any useful predictions, with errors reaching up to 30%. Additionally, the average runtime for the 80% training subset was faster, averaging 4,789 s, compared to 5,526 s for the 70% subset. This suggests that using a higher percentage of the data set for training not only improves prediction accuracy but also reduces computation time. Therefore, the 80% of data set is used for test group TS1.2 an TS2. It was observed that the accuracy of predictions based on previously predicted data points tends to diminish. As the prediction horizon extends, the deviation between predicted and actual values increases. Consequently, balancing the desired accuracy with the amount of available data for predictions is essential for obtaining reliable results.

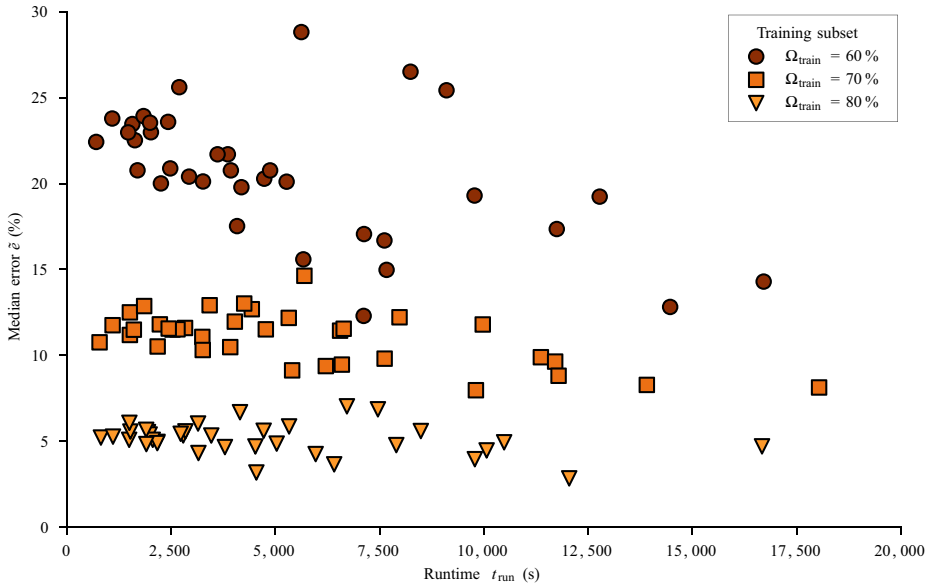
The results presented in Figure 6 show that there is no distinct pattern correlating the sampling rate f_s with the median error value. The training data sets used contain over 1,500 data points, which is adequate for leveraging the capabilities of the LSTM network effectively. Analysis indicates that reducing the number of data points does not significantly affect prediction accuracy. Specifically, the results of the average run time and the average median error were:

- using every second data point – 12.1% and 4,390 s;
- using every third data point – 12.4% and 6,364 s; and
- using every fourth data point – 12.2% and 4,806 s.

Using every fourth data point still provides sufficient data to capture long-term patterns while significantly reducing computational load per epoch. Therefore, for the larger-scale test, the decision was made to use every fourth data point to balance performance and efficiency.

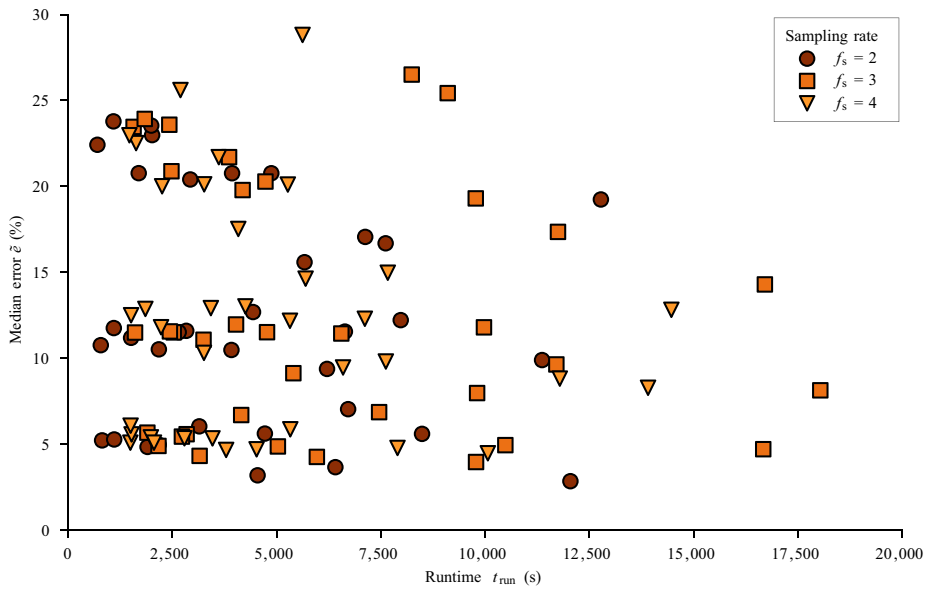
Figure 7 presents the results for the step size N_{step}. It is important to note that over 40% of computations with a step size of 50 were skipped due to excessively large errors, rendering this configuration unreliable for the best-performing setup. Although a step size of 10 results in a lower average runtime of 3,278 s, it is associated with a higher average median error of 13.5%. In comparison, a step size of 20 yields an average runtime of 4,393 s and an average median error of 12.2%. Therefore, the step size of 20 was chosen as the optimal configuration for the best-performing setup.

Figure 5 Results for test group TS1.1 representing the influence of the subset used for training Ω_{train}



Source: Figure by authors

Figure 6 Results for test group TS1.1 representing the influence of the data sampling rate f_s



Source: Figure by authors

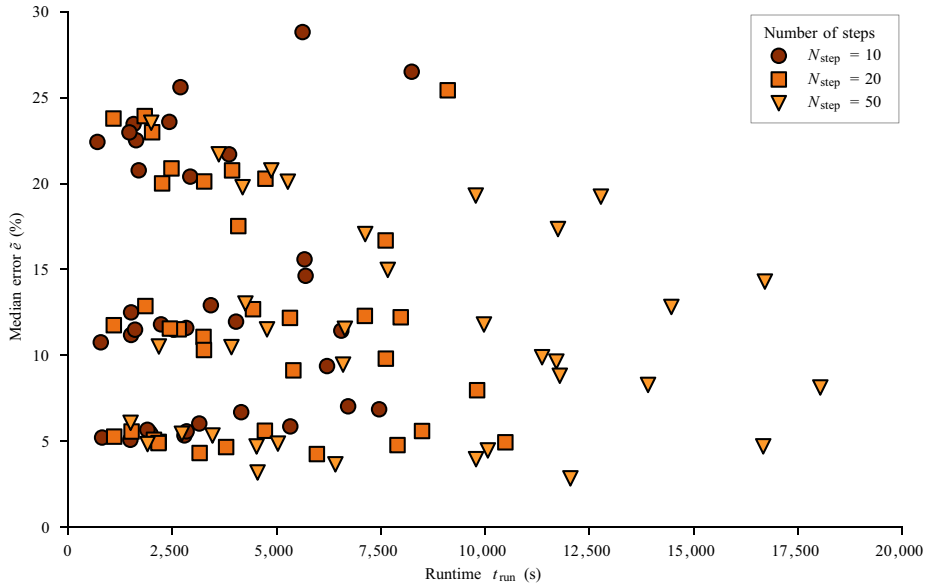
Figure 8 illustrates the impact of batch size B_k on performance. The analysis revealed a positive correlation between batch size and error: a smaller batch size generally resulted in a slightly lower error. However, batch size had a significant effect on runtime. Specifically, the results of the average runtime and the average median error were:

- 9,616 s and 11.5% for B_k of 5;
- 5,541 s and 11.9% for B_k of 10;
- 3,576 s and 12.4% for B_k of 20; and
- 2,013 s and 13.2% for B_k of 50.

This trend is expected, as the batch size has an inverse relationship with the number of computations performed in parallel. Consequently, a batch size of 20 was selected to achieve a balance between reducing runtime and maintaining a minimal impact on error.

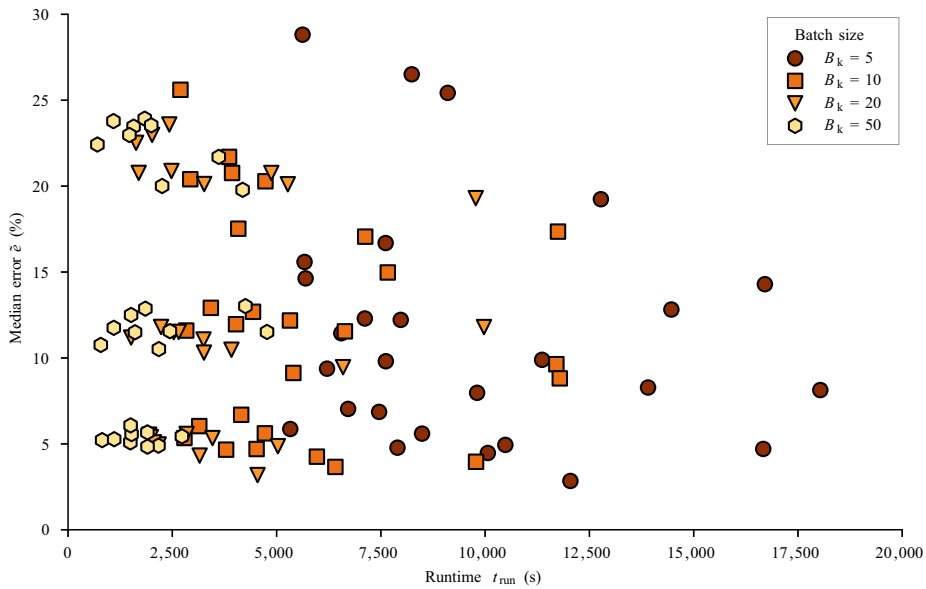
For the tests conducted for TS1.2 in Section 4.2 and TS2 in Section 4.3, the best-performing LSTM setup consisted of 80% of the data set for training, a step and batch size of 20 and a sampling rate of 4 was selected. This configuration achieved an average median error of 3.59% in the TS1.1 test group, with an average

Figure 7 Results for test group TS1.1 representing the influence of the step size N_{step}



Source: Figure by authors

Figure 8 Results for test group TS1.1 representing the influence of the batch size B_k



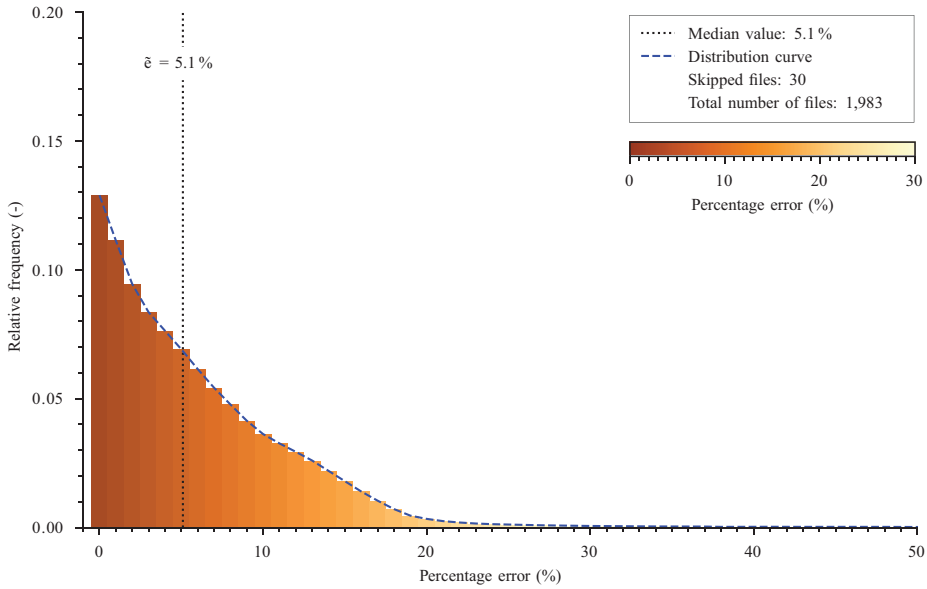
Source: Figure by authors

runtime of 2,063 s. The complete result table for all parameter combinations tested is provided in Table A1 in Appendix 1.

4.2 Numerical data test group TS1.2

The previously identified best-performing LSTM setup was evaluated on the TS1.2 test group, which consists of 1,983 files, to verify whether the selected parameter combination maintains its performance on a significantly larger data set. Figure 9 displays a histogram showing the relative frequency of percentage errors across all files, based on the comparison between predicted data points $\hat{y}_i$ and corresponding actual values y_i. The median error $\tilde{e}$ for the TS1.2 test group was 5.1%, which is consistent with the results observed for the TS1.1 test group. The results follow a power law distribution, suggesting that the trained LSTM models tend to overestimate the predicted axial stress levels σ_1. The results indicate a higher frequency of lower percentage errors. Furthermore, 30 files, representing approximately 1.5% of the test group, were excluded from the analysis due to the presence of NaN values or

Figure 9 Histogram showing the relative frequency of percentage errors across all investigated files from the training of test group TS1.2



Source: Figure by authors

excessively high error rates. Figure 10 presents the training results for selected samples from the TS1.2, conducted using the identified best-performing LSTM configuration. Figure 11 shows the RMSE values across training epochs for selected samples from TS1.2. The high magnitudes result from the absence of data scaling, with overall trends indicating clear convergence.

4.3 Laboratory data test group TS2

Table 1 presents the results obtained from applying the previously determined best-performance LSTM setup to the test group TS2, which comprises eight data sets S1 to S8 from CRS laboratory tests. The median error values $\tilde{e}$ range from 0.41% to 7.50%. The average median error value of 3.86% demonstrates that the predicted axial stress σ_1 values at higher stress levels closely align with the actual laboratory measurements. This indicates that the LSTM model effectively captures the stress–strain behaviour of the soil at higher stress levels, based on the data from lower stress levels.

5. Conclusions

This study demonstrated the effectiveness of bidirectional LSTM networks in predicting soil stress–strain behaviour during CRS tests. Extensive parameter studies identified the optimal LSTM architecture for capturing nonlinear sequential patterns and complex dependencies in soil compression. The trained model effectively predicted new data based on provided values at lower stress levels. Additionally, the study revealed a direct correlation between the proportion of the data set used for training and prediction accuracy, highlighting the need to balance accuracy with data availability. The identified LSTM setup was successfully applied to both a larger numerically generated test group and real-life laboratory data sets, accurately predicting soil behaviour at higher stress levels. The presented approach offers a valuable tool for geotechnical engineering, potentially reducing the time, costs and resources associated with conducting extensive laboratory tests.

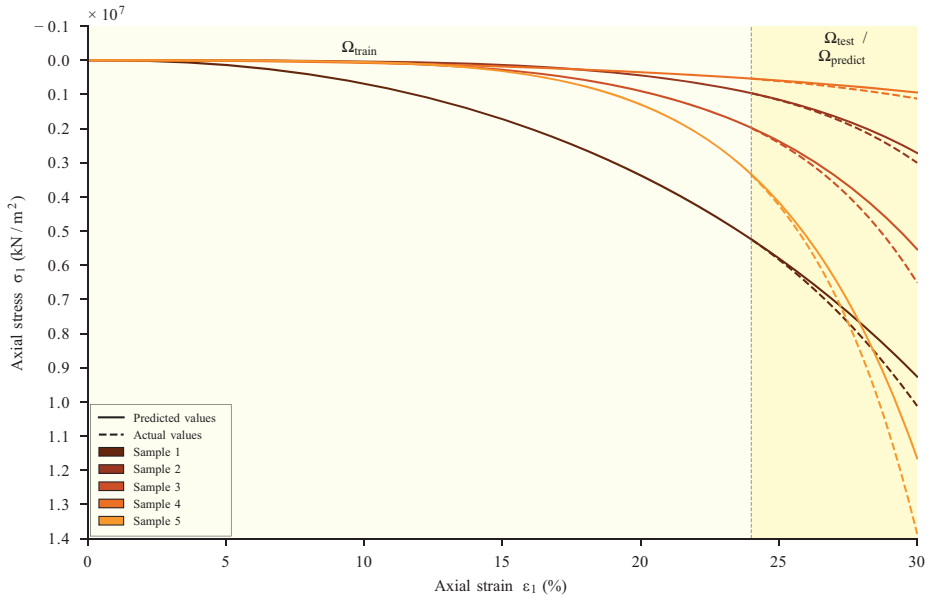
5.1 Limitations

When applying LSTM networks for predicting nonlinear soil behaviour, several limitations should be considered. LSTMs, like all neural networks, are often considered "black boxes" due to their complex internal structures, which consist of interconnected layers of neurons. This complexity makes it difficult to interpret the prediction mechanisms and gain insights into how decisions are made. Additionally, there is no universal LSTM architecture that is suitable for every task. As a result, preliminary studies and extensive experimentation are necessary to identify the optimal setup for each specific application. LSTM models also struggle with predicting rare or extreme changes in data sets that are underrepresented in the training data. Patterns that significantly deviate from those seen during training, such as abrupt changes or edges in a curve, are particularly challenging for LSTMs to predict accurately.

This study focused exclusively on unsaturated coarse-grained soils. To improve the model's generalisability, future research will incorporate data sets from a wider range of soil types and address the following experimental constraints of this study in the representation of soil behaviour. The CRS tests did not include unloading and reloading cycles, which are critical for capturing the full stress–strain response under complex loading paths. Furthermore, the soil samples were not subjected to preloading, neglecting the effects of overconsolidation, which are particularly significant for soft, fine-grained soils.

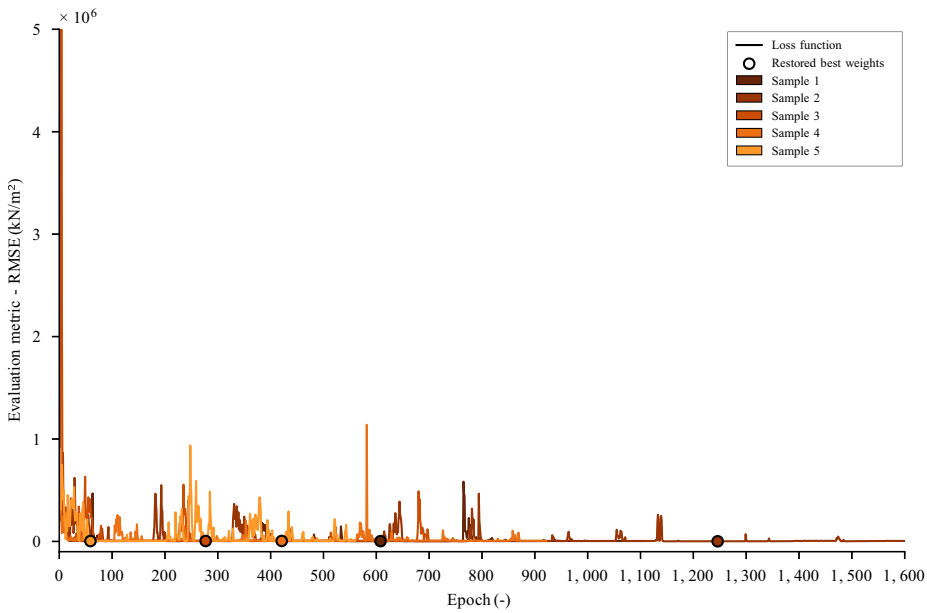
To reduce computational effort, the use of a simpler constitutive model, such as the power law-based models by Ohde (1939) or Janbu (1970), capable of capturing nonlinear soil behaviour under compression (Massarsch and Fellenius, 2019), is possible. Furthermore, when large data

Figure 10 Training results of the best-performing model configuration for selected samples from test group TS1.2



Source: Figure by authors

Figure 11 RMSE values across training epochs for selected samples from test group TS1.2



Source: Figure by authors

sets are not available, as was the case in this study, modelling a single integration point instead of a full finite element to simulate a CRS test offers a faster method for generating a new data set.

Overall, the objective of this work is to establish a foundation for future studies that will incorporate more complex soil behaviours.

5.2 Outlook

Although the presented approach offers a promising solution for predicting the nonlinear behaviour of coarse-grained soil during CRS tests, further investigation is needed to assess its adaptability to different soil types, such as fine-grained soil. Due to the lack of laboratory data, this study primarily relies on numerically generated results. Therefore, the method should be extensively

Table 1 Results for the test group TS2

Soil sample	Median error $\widetilde{e}$ (%)
S1	0.96
S2	6.50
S3	6.77
S4	1.85
S5	0.92
S6	7.50
S7	5.94
S8	0.41

Source(s): Table by authors

validated using a large test group consisting of laboratory results. More complex soil behaviour, including unloading and reloading cycles, can be incorporated into the input data sets to assess the robustness of the developed LSTM framework. Modifying this approach to predict triaxial test results and applying it to forecast stress levels beneath engineering structures, offers promising directions for future research. Furthermore, integrating easily measurable soil features into the framework, such as the void ratio in the densest and loosest states, further refines the approach.

Implementing data scaling techniques is known to enhance computational efficiency and can improve the robustness of the framework, despite the challenges associated with predicting future data points, similarly to using regularisation and dropout methods. Exploring other advanced techniques, such as physically informed neural networks (PINN) and EPR, offers complementary solutions to LSTM for future enhancing model performance.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X. (2015), "TensorFlow: large-scale machine learning on heterogeneous systems", doi: 10.5281/zenodo.4724125.

Adamkiewicz, M., Chen, T., Caccavale, A., Gardner, R., Culbertson, P., Bohg, J. and Schwager, M. (2022), "Vision-only robot navigation in a neural radiance world", *IEEE Robotics and Automation Letters*, Vol. 7 No. 2, pp. 4606-4613, doi: 10.1109/LRA.2022.3150497.

Atkinson, J.H. (2000), "Non-linear soil stiffness in routine design", *Géotechnique*, Vol. 50 No. 5, pp. 487-508, doi: 10.1680/geot.2000.50.5.487.

Bienen, B., Fan, S., Schröder, M. and Randolph, M.F. (2021), "Effect of the installation process on monopile lateral response", *Proceedings of the Institution of Civil Engineers – Geotechnical Engineering*, Vol. 174 No. 5, pp. 530-548, doi: 10.1680/jgeen.20.00219.

Cerek, K. and Grabe, J. (2023), "Numerical simulation and optimization of dike geometry using multi-objective evolutionary algorithm NSGA-II", in Zdravkovic, L., Kontoe, S., Tsiampousi, A. and Taborda, D.M.G. (Eds), *10th European Conference on Numerical Methods in Geotechnical Engineering, London, England*, doi: 10.53243/NUMGE2023-80.

Cerek, K., Dao, D.A., Hadjiloo, E. and Grabe, J. (2024a), "Application of LSTM time series forecasting method for predicting compression curves of soil", *Proceedings of the 17th Pan-American Conference on Soil Mechanics and Geotechnical Engineering (XVII PCSMGE), and 2nd Latin-American Regional Conference of the International Association for Engineering Geology and the Environment (IAEG), La Serena, Chile*, 2024.

Cerek, K., Dao, D.A., Hadjiloo, E. and Grabe, J. (2024b), "Dataset of simulated CRS tests for advanced soil parameter identification", TUHH Universitätsbibliothek, doi: 10.15480/882.9435.

Cerek, K., Gupta, A., Dao, D.A., Hadjiloo, E. and Grabe, J. (2024c), "Python implementation of bidirectional LSTM for sequential data processing", Source Code, TUHH Universitätsbibliothek, doi: 10.15480/882.13190.

Cerek, K., Hadjiloo, E. and Grabe, J. (2024d), "Sustainable dike adaptation measures using finite element method and optimization algorithm NSGA-II", in Duc Long, P. and Dung, N.T. (Eds), *Proceedings of the 5th International Conference on Geotechnics for Sustainable Infrastructure Development*, Vol. 395, Springer Nature Singapore, Singapore, pp. 2077-2091.

Cerek, K., Hadjiloo, E. and Grabe, J. (2024e), "Application of optimization algorithms in geotechnical engineering as decision-making support tool", *Proceedings of the XVIII European Conference on Soil Mechanics and Geotechnical Engineering (ECSMGE) 2024, Lisbon, Portugal*, ISBN: 978-1-032-54816-6 (manuscript accepted).

Charles, J. and Gourvenec, S. (2024), "GγSANDnet: a neural network tool for prediction of shear stiffness (G)–shear strain (γ) relationship for sands", *SoftwareX*, Vol. 27, p. 101823, doi: 10.1016/j.softx.2024.101823.

Chen, S. and Billings, S.A. (1992), "Neural networks for nonlinear dynamic system modelling and identification", *International Journal of Control*, Vol. 56 No. 2, pp. 319-346, doi: 10.1080/00207179208934317.

Chen, W., Ding, J., Shi, C., Wang, T. and Connolly, D.P. (2024), "Geotechnical correlation field-informed and data-driven prediction of spatially varying geotechnical properties", *Computers and Geotechnics*, Vol. 171, p. 106407, doi: 10.1016/j.compgeo.2024.106407.

Chollet, F. and others (2015), "Keras: the python deep learning library, version 2", available at: https://keras.io

Choudhary, K., DeCost, B., Chen, C., Jain, A., Tavazza, F., Cohn, R., Woo Park, C., Choudhary, A., Agrawal, A., Billinge, S.J.L., Holm, E., Ping Ong, S. and Wolverton, C. (2022), "Recent advances and applications of deep learning methods in materials science", *Npj Computational Materials*, Vol. 8 No. 1, p. 59, doi: 10.1038/s41524-022-00734-6.

Dao, D.A. and Banduch, M. (2023), *Geotechnical Laboratory Investigation on the Baltic Sea Sand Sample Z_VC_011*, TUHH Universitätsbibliothek, Hamburg, doi: 10.15480/882.8440.

Dao, D.A., Alkateeb, D. and Schröder, M. (2023), "Discrepancies between element tests and large-scale LDFE simulations: a case study on anchor kinematics during

installation in clay", *Computers and Geotechnics*, Vol. 163, p. 105698, doi: 10.1016/j.compgeo.2023.105698.

Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A., Lackenby, M., Williamson, G., Hassabis, D. and Kohli, P. (2021), "Advancing mathematics by guiding human intuition with AI", *Nature*, Vol. 600 No. 7887, pp. 70-74, doi: 10.1038/s41586-021-04086-x.

Deneko, E., Filaj, E., Gheibi, M. and Moezzi, R. (2024), "Predicting pavement surface conditions through artificial neural networks", *Proceedings of the Institution of Civil Engineers - Municipal Engineer*, Vol. 177 No. 3, pp. 99-110, doi: 10.1680/jmuen.23.00054.

Dubey, S.R., Singh, S.K. and Chaudhuri, B.B. (2022), "Activation functions in deep learning: a comprehensive survey and benchmark", *Neurocomputing*, Vol. 503, pp. 92-108, doi: 10.1016/j.neucom.2022.06.111.

Ellis, G.W., Yao, C., Zhao, R. and Penumadu, D. (1995), "Stress-strain modeling of sands using artificial neural networks", *Journal of Geotechnical Engineering*, Vol. 121 No. 5, pp. 429-435, doi: 10.1061/(asce)0733-9410(1995)121:5(429).

Erzin, Y. and Ecemis, N. (2017), "The use of neural networks for the prediction of cone penetration resistance of silty sands", *Neural Computing and Applications*, Vol. 28 No. S1, pp. 727-736, doi: 10.1007/s00521-016-2371-z.

Gao, B., Wang, R.R., Lin, C., Guo, X., Liu, B. and Zhang, W. (2021), "TBM penetration rate prediction based on the long short-term memory neural network", *Underground Space*, Vol. 6 No. 6, pp. 718-731, doi: 10.1016/j.undsp.2020.01.003.

Grabe, J., Henke, S., Kinzler, S. and Pucker, T. (2010), "Inverse determination of soil density and stress state using dispersion wave measurements and cone penetration tests in a non-layered soil", *Soil Dynamics and Earthquake Engineering*, Vol. 30 No. 6, pp. 481-489, doi: 10.1016/j.soildyn.2010.01.004.

Graves, A. and Schmidhuber, J. (2005), "Framewise phoneme classification with bidirectional LSTM and other neural network architectures", *Neural Networks: The Official Journal of the International Neural Network Society*, Vol. 18 Nos 5/6, pp. 602-610, doi: 10.1016/j.neunet.2005.06.042.

Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R. and Schmidhuber, J. (2017), "LSTM: a search space odyssey", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28 No. 10, pp. 2222-2232, doi: 10.1109/TNNLS.2016.2582924.

Guan, Q.Z., Yang, Z.X., Guo, N. and Hu, Z. (2023), "Finite element geotechnical analysis incorporating deep learning-based soil model", *Computers and Geotechnics*, Vol. 154, p. 105120, doi: 10.1016/j.compgeo.2022.105120.

Hamzah, D.M. and Ronaldo, A.D. (2021), "Effective soil type classification using convolutional neural network", *International Journal of Informatics and Computation*, Vol. 3 No. 1, p. 20, doi: 10.35842/ijicom.v3i1.33.

Herle, I. and Gudehus, G. (1999), "Determination of parameters of a hypoplastic constitutive model from properties of grain assemblies", *Mechanics of Cohesive-Frictional Materials*, Vol. 4 No. 5, pp. 461-486, doi: 10.1002/(SICI)1099-1484(199909)4:5<461::AID-CFM71>3.0.CO;2-P.

Hochreiter, S. (1991), "Untersuchungen zu dynamischen neuronalen netzen", Diploma Thesis, Technical University of Munich, Munich, Germany.

Hochreiter, S. and Schmidhuber, J. (1997), "Long short-term memory", *Neural Computation*, Vol. 9 No. 8, pp. 1735-1780, doi: 10.1162/neco.1997.9.8.1735.

Irani, N., Salimi, M., Golestaneh, P., Tafili, M., Wichtmann, T. and Lederer, J. (2024), "Deep learning-based analysis of true triaxial DEM simulations: role of fabric and particle aspect ratio", *Computers and Geotechnics*, Vol. 173, p. 106529, doi: 10.1016/j.compgeo.2024.106529.

Janbu, N. (1970), "Grunnlag i geoteknikk", Tapir Forlag.

Kingma, D.P. and Ba, J. (2014), "Adam: a method for stochastic optimization", *3rd International Conference on Learning Representations, San Diego, USA*, doi: 10.48550/arXiv.1412.6980.

Kiran Pandiri, D.N., Murugan, R. and Goel, T. (2024), "Smart soil image classification system using lightweight convolutional neural network", *Expert Systems with Applications*, Vol. 238 No. 1, p. 122185, doi: 10.1016/j.eswa.2023.122185.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, Vol. 521 No. 7553, pp. 436-444, doi: 10.1038/nature14539.

Levine, E.R., Kimes, D.S. and Sigillito, V.G. (1996), "Classifying soil structure using neural networks", *Ecological Modelling*, Vol. 92 No. 1, pp. 101-108, doi: 10.1016/0304-3800(95)00199-9.

Liano, K. (1996), "Robust error measure for supervised neural network learning with outliers", *IEEE Transactions on Neural Networks*, Vol. 7 No. 1, pp. 246-250, doi: 10.1007/978-3-7091-6230-9_106.

Liu, H., Su, H., Sun, L. and Dias-da-Costa, D. (2024), "State-of-the-art review on the use of AI-enhanced computational mechanics in geotechnical engineering", *Artificial Intelligence Review*, Vol. 57 No. 8, p. 196, doi: 10.1007/s10462-024-10836-w.

Machaček, J., Staubach, P., Tavera, C.E.G., Wichtmann, T. and Zachert, H. (2022), "On the automatic parameter calibration of a hypoplastic soil model", *Acta Geotechnica*, Vol. 17 No. 11, pp. 5253-5273, doi: 10.1007/s11440-022-01669-4.

Massarsch, K.R. and Fellenius, B.H. (2019), "In situ tests for settlement design of compacted sand", *Proceedings of the Institution of Civil Engineers – Geotechnical Engineering*, Vol. 172 No. 3, pp. 207-217, doi: 10.1680/jgeen.18.00046.

Mendez, F.J., Pasculli, A., Mendez, M.A. and Sciarra, N. (2021), "Calibration of a hypoplastic model using genetic algorithms", *Acta Geotechnica*, Vol. 16 No. 7, pp. 2031-2047, doi: https://dx.doi.org/10.1007/s11440-020-01135-z.

Merity, S., Keskar, N.S. and Socher, R. (2017), *Regularizing and Optimizing LSTM Language Models*, Salesforce Research, Palo Alto, CA, doi: 10.48550/arXiv.1708.02182.

Mian, S. (2022), "Foundations of artificial intelligence and applications", *Journal of Artificial Intelligence and Technology*, Vol. 2 No. 1, pp. 1-2, doi: 10.37965/jait.2022.01.

Niemunis, A. and Herle, I. (1997), "Hypoplastic model for cohesionless soils with elastic strain range: mechanics of cohesive-frictional materials", *An International Journal on Experiments,*

Modelling and Computation of Materials and Structures, Vol. 2 No. 4, pp. 279-299, doi: 10.1002/(SICI)1099-1484(199710)2:4%3C279::AID-CFM29%3E3.0.CO;2-8.

Ohde, J. (1939), "Zur theorie der druckverteilung im baugrund", *Der Bauingenieur*, Vol. 20, pp. 451-459.

Pepe, A., Li, J., Rolf-Pissarczyk, M., Gsaxner, C., Chen, X., Holzapfel, G.A. and Egger, J. (2020), "Detection, segmentation, simulation and visualization of aortic dissections: a review", *Medical Image Analysis*, Vol. 65, p. 101773, doi: 10.1016/j.media.2020.101773.

PLAXIS (2024), "PLAXIS 2D manual, version 2024.1", PLAXIS Bentley.

Prechelt, L. (2002), "Early stopping – but when?", in Orr, G.B. and Müller, K.R. (Eds), *Neural Networks: Tricks of the Trade*, Springer, Heidelberg, pp. 55-69.

Pucker, T. and Grabe, J. (2011), "Structural optimization in geotechnical engineering: basics and application", *Acta Geotechnica*, Vol. 6 No. 1, pp. 41-49, doi: 10.1007/s11440-011-0134-7.

Qi, C. and Fourie, A. (2018), "A real-time back-analysis technique to infer rheological parameters from field monitoring", *Rock Mechanics and Rock Engineering*, Vol. 51 No. 10, pp. 3029-3043, doi: 10.1007/s00603-018-1513-2.

Rauter, S. and Tschuchnigg, F. (2021), "CPT data interpretation employing different machine learning techniques", *Geosciences*, Vol. 11 No. 7, p. 265, doi: 10.3390/geosciences11070265.

Rezania, M. (2008), "Evolutionary polynomial regression based constitutive modelling and incorporation in finite element analysis", PhD Thesis, Exeter University, Exeter, England.

Rezania, M., Javadi, A.A. and Giustolisi, O. (2008), "An evolutionary-based data mining technique for assessment of civil engineering systems", *Engineering Computations*, Vol. 25 No. 6, pp. 500-517, doi: 10.1108/02644400810891526.

Sagheer, A. and Kotb, M. (2019), "Time series forecasting of petroleum production using deep LSTM recurrent networks", *Neurocomputing*, Vol. 323 No. 3, pp. 203-213, doi: 10.1016/j.neucom.2018.09.082.

Sak, H., Senior, A. and Beaufays, F. (2014), "Long short-term memory recurrent neural network architectures for large scale acoustic modeling", *Proceedings of the 15th Annual Conference of the International Speech Communication Association, INTERSPEECH, Singapore*, pp. 338-342, doi: 10.21437/Interspeech.2014-80.

Schuster, M. and Paliwal, K.K. (1997), "Bidirectional recurrent neural networks", *IEEE Transactions on Signal Processing*, Vol. 45 No. 11, pp. 2673-2681, doi: 10.1109/78.650093.

Siami-Namini, S., Tavakoli, N. and Siami Namin, A. (2018), "A comparison of ARIMA and LSTM in forecasting time series", *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA*, pp. 1394-1401, doi: 10.1109/ICMLA.2018.00227.

Sinsomboonthong, S. (2022), "Performance comparison of new adjusted min-max with decimal scaling and statistical column normalization methods for artificial neural network classification", *International Journal of Mathematics and Mathematical Sciences*, Vol. 2022 No. 6, pp. 1-9, doi: 10.1155/2022/3584406.

Smith, R. and Wahls, H. (1969), "Consolidation under constant rates of strain", *Journal of the Soil Mechanics and Foundations Division*, Vol. 95 No. 2, pp. 519-539, doi: 10.1061/JSFEAQ.0001263.

Stapelfeldt, M., Bienen, B. and Grabe, J. (2020), "The influence of the drainage regime on the installation and the response to the vertical cyclic loading of suction caissons in dense sand", *Ocean Engineering*, Vol. 215, p. 107105, doi: 10.1016/j.oceaneng.2020.107105.

Staudemeyer, R.C. and Morris, E.R. (2019), "Understanding LSTM – a tutorial into long Short-Term memory recurrent neural networks", doi: 10.48550/arXiv.1909.09586.

Tatar, A., Haghighi, M. and Zeinijahromi, A. (2024), "Experiments on image data augmentation techniques for geological rock type classification with convolutional neural networks", *Journal of Rock Mechanics and Geotechnical Engineering*, pp. 106-125, doi: 10.1016/j.jrmge.2024.02.015.

United Nations (2015), "Transforming our world: the 2030 agenda for sustainable development", available at: https://sdgs.un.org/2030agenda (accessed 7 August 2024).

Van Rossum, G. and Drake, F.L. (2009), *Python 3 Reference Manual*, CreateSpace, Scotts Valley, CA, ISBN: ISBN:978-1-4414-1269-0.

Von Wolffersdorff, P.A. (1996), "A hypoplastic relation for granular materials with a predefined limit state surface", *Mechanics of Cohesive-Frictional Materials*, Vol. 1 No. 3, pp. 251-271, doi: 10.1002/(SICI)1099-1484(199607)1:3%3C251::AID-CFM13%3E3.0.CO;2-3.

Wang, G., Wang, Z., Chen, J., *et al.* (2023), "An empirical study on numerical bugs in deep learning programs", *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, Rochester, MI, USA*, pp. 1-5, doi: 10.1145/3551349.3559561.

Xu, S. and Niu, R. (2018), "Displacement prediction of baijiabao landslide based on empirical mode decomposition and long short-term memory neural network in three gorges area", *Computers and Geosciences*, Vol. 111 No. 2, pp. 87-96, doi: 10.1016/j.cageo.2017.10.013.

Xu, J.J., Zhang, H., Tang, C.-S., Cheng, Q., Liu, B. and Shi, B. (2022), "Automatic soil desiccation crack recognition using deep learning", *Géotechnique*, Vol. 72 No. 4, pp. 337-349, doi: 10.1680/jgeot.20.P.091.

Zhang, P., Yin, Z.-Y. and Sheil, B. (2024), "A physics-informed data-driven approach for consolidation analysis", *Géotechnique*, Vol. 74 No. 7, pp. 620-631, doi: 10.1680/jgeot.22.00046.

Zhang, N., Shen, S., Zhou, A. and Jin, Y. (2021), "Application of LSTM approach for modelling stress–strain behaviour of soil", *Applied Soft Computing*, Vol. 100 No. 10, p. 106959, doi: 10.1016/j.asoc.2020.106959.

Zhu, J.H., Zaman, M.M. and Anderson, S.A. (1998a), "Modeling of soil behavior with a recurrent neural network", *Canadian Geotechnical Journal*, Vol. 35 No. 5, pp. 858-872, doi: 10.1139/t98-042.

Zhu, J.H., Zaman, M.M. and Anderson, S.A. (1998b), "Modelling of shearing behaviour of a residual soil with recurrent neural network", *International Journal for Numerical and Analytical Methods in Geomechanics*, Vol. 22 No. 8, pp. 671-687, doi: 10.1002/(SICI)1096-9853(199808)22:8%3C671::AID-NAG939%3E3.0.CO;2-Y.

Further reading

Abbas, A.W. and Muzammil, H. (2021), "A literature review of artificial intelligence", *UMT Artificial Intelligence Review*, Vol. 1 No. 1, pp. 1-27, doi: 10.32350/UMT-AIR/0101/01.

Lu, M., AbouRizk, S.M. and Hermann, U.H. (2001), "Sensitivity analysis of neural networks in spool fabrication productivity studies", *Journal of Computing in Civil Engineering*, Vol. 15 No. 4, pp. 299-308, doi: 10.1061/(asce)0887-3801(2001)15:4(299).

Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T. and Ramamoorthi, R. (2020), "NeRF: representing scenes as neural radiance fields for view synthesis", in Vedaldi, A., Bischof, H., Brox, T. and Frahm, J. (Eds), *Proceedings of the European Conference on Computer Vision (ECCV) 2020*, Online, pp. 405-421.

Pascanu, R., Mikolov, T. and Bengio, Y. (2012), "On the difficulty of training recurrent neural networks", *Proceedings of the 30th International Conference on Machine Learning*, *Atlanta, GA*, Vol. 28, pp. 1310-1318, doi: 10.48550/arXiv.1211.5063.

Appendix 1

Table A1 Results for the test group TS1.1

Training subset Ω_{train} (%)	Sampling rate f_s	Step size N_{step}	Batch size B_k	Median error $\tilde{e}$ (%)	Runtime t_{run} (s)	Skipped files
60	2	10	5	15.59	5,669.67	3
			10	20.40	2,930.39	0
			20	20.77	1,694.03	0
			50	22.42	703.59	1
		20	5	16.65	7,637.35	2
			10	20.63	3,927.81	1
			20	22.90	2,023.20	5
			50	23.76	1,090.67	2
		50	5	18.39	12,875.11	51
			10	16.92	7,148.55	47
			20	21.00	4,870.93	45
			50	23.87	2,007.77	59
	3	10	5	26.91	8,164.33	4
			10	21.66	3,860.52	1
			20	23.48	2,427.76	2
			50	23.45	1,554.78	3
		20	5	25.44	9,159.86	4
			10	20.63	4,758.27	3
			20	20.83	2,480.77	6
			50	23.86	1,826.05	4
		50	5	14.30	16,602.88	39
			10	17.50	11,707.84	43
			20	19.20	9,860.39	52
			50	19.37	4,288.41	58
	4	10	5	28.82	5,622.31	4
			10	25.56	2,692.54	0
			20	22.49	1,638.11	8
			50	22.49	1,481.58	7
		20	5	12.24	7,150.72	7
			10	17.28	4,044.59	2
			20	20.05	3,289.50	3
			50	20.16	2,257.96	4
		50	5	12.98	14,511.77	42
			10	14.87	7,724.67	48
			20	20.22	5,219.23	52
			50	22.19	3,646.39	48
70	2	10	5	9.38	6,209.00	4
			10	11.59	2,833.58	0
			20	11.19	1,511.635	1
			50	10.76	784.58	1
		20	5	12.26	7,949.62	5
			10	12.70	4,424.02	1
			20	11.68	2,648.71	0
			50	11.78	1,102.65	2
		50	5	9.67	11,417.95	51
			10	11.55	6,637.31	51
			20	10.47	3,950.24	61
			50	10.41	2,172.27	51
	3	10	5	11.33	6,597.91	1
			10	11.94	4,049.33	0
			20	11.48	2,536.37	1
			50	11.47	1,606.94	4
		20	5	7.96	9,836.19	2
			10	9.14	5,376.99	3
			20	11.22	3,245.09	1
			50	11.60	2,441.98	1
		50	5	8.00	18,027.94	40
			10	9.50	11,761.01	59
			20	11.98	9,569.75	50
			50	11.51	4,770.46	58
	4	10	5	14.64	5,624.33	2
			10	12.92	3,433.78	0
			20	11.75	2,228.16	0
			50	12.44	1,505.3	3

(*continued*)

Table A1

Training subset Ω_{train} (%)	Sampling rate f_s	Step size N_{step}	Batch size B_k	Median error $\tilde{e}$ (%)	Runtime t_{run} (s)	Skipped files
		20	5	9.81	7,584.89	3
			10	12.10	5,340.82	2
			20	10.43	3,236.20	1
			50	12.92	1,845.67	3
		50	5	8.38	13,844.60	41
			10	8.60	11,919.38	47
			20	9.25	6,653.15	50
			50	13.52	4,264.39	55
80	2	10	5	7.01	6,699.17	1
			10	6.03	3,146.00	1
			20	5.52	1,951.06	1
			50	5.20	818.03	0
		20	5	5.68	8,517.46	3
			10	5.58	4,704.24	1
			20	4.96	2,167.01	3
			50	5.27	1,094.96	1
		50	5	3.01	11,765.43	44
			10	3.56	6,564.75	62
			20	3.22	4,561.96	61
			50	4.84	1,924.55	52
	3	10	5	6.84	7,462.34	0
			10	6.69	4,125.34	0
			20	5.56	2,848.49	0
			50	5.66	1,849.92	0
		20	5	4.94	10,527.23	1
			10	4.22	5,995.84	1
			20	4.36	3,178.40	3
			50	4.85	2,176.88	1
		50	5	4.75	16,417.65	52
			10	3.94	9,795.91	57
			20	5.14	5,004.05	56
			50	5.23	2,689.43	51
	4	10	5	5.83	5,343.74	1
			10	5.32	2,803.18	0
			20	5.39	1,993.18	0
			50	5.07	1,482.25	1
		20	5	4.71	7,966.34	3
			10	4.73	3,789.37	3
			20	5.14	2,072.83	2
			50	5.58	1,529.13	2
		50	5	4.44	10,130.59	57
			10	4.70	4,361.71	46
			20	5.17	3,451.45	56
			50	6.08	1,463.04	60

Source(s): Table by authors

Appendix 2

Table A2 Maximum values of the axial strain level ε_1 and the axial stress σ_1 level for the soil samples in test group TS2

Soil sample	Maximum level of the axial strain ε_1 (%)	Maximum level of the axial stress σ_1 (kN/m^2)	Initial void ratio e_0 (−)	Final void ratio e_{end} (−)
S1	4.24	2,927.6	0.760	0.686
S2	7.97	3,251.8	0.860	0.711
S3	4.63	2,318.2	0.828	0.744
S4	3.45	2,411.1	0.641	0.585
S5	5.31	2,384.4	0.668	0.580
S6	2.67	2,424.1	0.646	0.602
S7	1.62	2,607.2	0.651	0.625
S8	7.36	2,610.7	0.865	0.728

Source(s): Table by authors

Corresponding author
Kacper Cerek can be contacted at: kacper.cerek@tuhh.de