Proceedings of the 21st ICSMGE, Vienna, Austria, 14 – 19 June 2026. Pistrol, Adam & Schweiger (eds.)
Published by: ÖGG, Austrian Society for Geomechanics, Salzburg, Austria, ISBN 978-3-9503898-4-5
https://doi.org/10.53243/ICSMGE2026-1117

Data-driven prediction of soil compression behaviour using LSTM neural networks

Kacper Cerek, Elnaz Hadjiloo, Jürgen Grabe
Institute of Geotechnical Engineering and Construction Management, Hamburg University of Technology, Hamburg, Germany, kacper.cerek@tuhh.de

ABSTRACT: This study investigates the potential of long short-term memory (LSTM) neural networks for predicting the stress-strain behaviour of soil in constant rate of strain (CRS) tests. Unlike traditional methods reliant on predefined constitutive models, data-driven approaches offer the ability to capture complex, nonlinear patterns directly from test data. In this work, we evaluate four deep learning architectures, unidirectional LSTM, bidirectional LSTM (BiLSTM), and two sequence-to-sequence (seq2seq) variants, for sequential prediction of axial stress at higher strain levels based on stress values at lower strains. The models are trained and tested on numerically generated CRS data for unsaturated coarse-grained soil. Different training set sizes are systematically explored to assess the influence of data availability on prediction accuracy. Early stopping is employed to prevent overfitting, and model performance is evaluated using the mean absolute percentage error (MAPE). The results demonstrate that all architectures perform similarly well when at least 60% of the data is used for training, with minor differences in accuracy and training times. Although the BiLSTM seq2seq model showed slightly better performance for smaller training sets, the advantage diminished with larger datasets. The study highlights the potential of LSTM-based models to predict soil behaviour without prior knowledge of material parameters, paving the way for reducing the duration and cost of laboratory testing in geotechnical engineering. This work presents an initial comparison based on a single numerically generated dataset, serving as a proof of concept for future research. Further studies incorporating experimental data, model regularisation, and broader statistical evaluation are recommended to validate the robustness of these models in practical applications.

KEYWORDS:, Neural networks, Artificial intelligence, Computational geotechnics, Long short-term memory networks.

1 INTRODUCTION

Recent advances in machine learning, particularly deep learning, have enabled significant progress across various scientific disciplines, including material science, mathematics, and engineering (LeCun et al., 2015; Choudhary et al., 2022). In geotechnical engineering, the nonlinear and time-dependent behaviour of soil presents challenges that traditional constitutive models cannot always capture. While methods like evolutionary algorithms and regression models have been applied for parameter calibration (Machaček et al., 2022; Rezania et al., 2008) and design optimisation (Cerek & Grabe, 2023; Cerek et al., 2024d, 2024e), their effectiveness remains limited by the underlying model assumptions.

Machine learning models — especially feedforward (FNN) and convolutional neural networks (CNN) — have found applications in soil classification, property prediction, and test interpretation (Rauter & Tschuchnigg, 2021; Levine et al., 1996). However, these models are typically suited for static or spatial problems and fall short when applied to sequential, time-dependent data such as stress-strain behaviour. Recurrent neural networks (RNNs), particularly long short-term memory (LSTM) networks (Hochreiter, 1991), offer a solution by capturing long-term dependencies in sequential data, making them a promising tool for modelling soil behaviour under load.

Although LSTM applications in geotechnics are still emerging, studies have shown their potential in predicting soil behaviour, tunnel performance and prediction of soil parameters (Qi & Fourie, 2018; Zhang et al., 2021; Guan et al., 2023; Cerek et al., 2024a, 2024b, 2025a, 2025b). Building on these works, this study explores LSTM-based architectures, including unidirectional, bidirectional, and sequence-to-sequence models, for predicting soil response during constant rate of strain (CRS) tests.

The primary goal is to examine the potential of various LSTM networks for predicting the stress-strain behaviour of coarse-grained unsaturated soil at higher stress levels based on initial stress data from CRS tests. This approach aims to reduce the need for full laboratory testing, thereby optimising both cost and time in geotechnical investigations and contributing to more sustainable engineering practices.

2 THEORY

2.1 *CRS test*

The CRS test involves the one-dimensional compression of a cylindrical soil sample, wherein axial strain is applied at a uniform rate. Lateral deformation is constrained, and vertical stresses are continuously recorded during the procedure. The present work utilises a comprehensive data set developed by Cerek et al. (2024c), which comprises numerical simulations designed to emulate two-dimensional compression element tests. These simulations were performed using the finite element method (FEM), incorporating rigorously defined boundary conditions. As depicted in Figure 1, axial strain was incrementally imposed on the specimen's upper boundary until a specified strain level was attained. The element test simulations utilised a hypoplastic constitutive model, initially developed by von Wolffersdorff (1996) and later extended through the intergranular strain concept by Niemunis and Herle (1997).

2.2 *Database*

The database employed in this study consists of axial stress-strain ($\sigma_1 - \varepsilon_1$) compression curves generated via FEM by Cerek et al. (2024c). The synthetic dataset contains 10,000 instances of CRS element test results for coarse-grained, dry soil, representing compression behaviour up to 30% axial strain. Each entry consists of 3,000 data points that capture the relationship between axial strain ε_1 and the corresponding axial stress σ_1. A randomly selected sample of this dataset, defined by specific soil parameters, is utilised in this study to compare various LSTM-based approaches for predicting soil compression behaviour based on the initial stress conditions in the CRS test. The following material model parameters were used for conducting the CRS test:

- φ_c (Critical state friction angle): 30.4 (°)
- h_s (granular stiffness): 39,950.0 (kN/m²)
- n (exponent): 0.42 (-)
- e_{d0} (minimum void ratio): 0.33 (-)
- e_{c0} (maximum void ratio): 0.5 (-)

- e_{i0} (critical void ratio): 1.62 (-)
- α (exponent): 0.02 (-)
- β (exponent): 6.2 (-)
- R (constant): 0.000131 (-)
- B_R (exponent): 0.7 (-)
- χ (exponent): 6.1 (-)
- m_R (factor): 7.9 (-)
- m_T (factor): 3.5 (-)
- e_0 (initial void ratio): 0.865 (-)

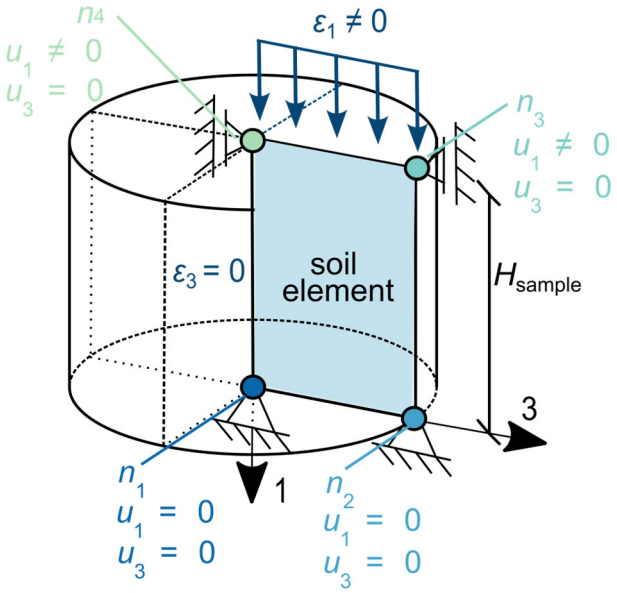

Figure 1. Setup of the numerical CRS element test.

2.3 *Long short-term memory network (LSTM)*

LSTM networks, a specialised form of RNNs, are well-suited for processing and learning from sequential data (Hochreiter & Schmidhuber, 1997). Due to their capacity to retain information across more than 1,000 data steps (Hochreiter, 1991), LSTMs have proven effective in applications such as time series prediction (Sagheer & Kotb, 2019). In contrast to conventional RNNs, LSTMs overcome limitations such as vanishing and exploding gradients by incorporating a constant error carousel, which ensures stable backpropagation over long sequences (Staudemeyer & Morris, 2019).

LSTMs operate by iteratively refining internal weights during training, allowing them to learn meaningful dependencies within the data. A typical neural network architecture comprises three principal components: an input layer, one or more hidden layers, and an output layer (Sak et al., 2014). Input data is formatted as a two-dimensional array, where one axis corresponds to the number of steps and the other to the input features. A LSTM layer consists of memory cells with distinct gate mechanisms, namely, forget, input, and output gates, that regulate the flow of information. In a LSTM network, an activation function is applied to the cell output to shape the signal passed to the next layer, while a recurrent activation function is used within the gating mechanisms to regulate information flow across data steps. The final prediction is typically produced by a dense output layer following the sequence-processing LSTM units.

2.3.1 *Sequence-to-sequence LSTM*

Sequence-to-sequence (seq2seq) LSTM models differ from standard LSTMs by employing an encoder-decoder architecture designed to map input sequences directly to output sequences of potentially different lengths. The encoder processes the entire input sequence into a fixed-length context vector, which the decoder then uses to generate the predicted output sequence step-by-step (Sutskever et al., 2014). This approach is particularly beneficial for modelling complex temporal dependencies and enables the network to predict entire sequences rather than single points. A key advantage of seq2seq LSTMs is their ability to mitigate error accumulation, which often occurs when predictions are fed back as inputs in traditional stepwise forecasting. By learning to generate sequences holistically, seq2seq models can improve stability and accuracy in long-term predictions, making them well-suited for applications such as soil compression behaviour forecasting where cumulative errors can significantly affect outcomes.

2.3.2 *Bidirectional LSTMs*

Bidirectional LSTM (BiLSTM) networks enhance the predictive capability of standard LSTMs by processing input sequences in both forward and backward directions. BiLSTMs were developed by Schuster and Paliwal (1997). This dual-pass approach allows the model to incorporate both past and future context when making predictions at each data step. As a result, BiLSTMs are particularly advantageous in tasks where understanding the full temporal context improves accuracy, such as sequence labelling and regression problems involving temporally interdependent data (Graves & Schmidhuber, 2005). Implementing a BiLSTM effectively doubles the number of neurons compared to a unidirectional LSTM, as separate set of units process the forward and backward passes.

3 METHODOLOGY

This study utilises stress-strain curves obtained from numerical CRS experiments. The selected compression curve is formatted to suit the LSTM model and employed in the supervised training process. Training uses a subset of axial stress values at lower stress levels Ω_{train} and continues until early stopping criteria are met. The trained LSTM model then predicts values at higher stress levels Ω_{predict} which are compared against the actual test values Ω_{test} excluded from training.

The paper aims to identify the best-performing LSTM architecture for predicting laboratory test results, such as those from CRS tests. The following network types are investigated:

- LSTM;
- BiLSTM;
- LSTM seq2seq;
- BiLSTM seq2seq.

Each neural network is trained on four different sizes of training subsets Ω_{train}, corresponding to train/test splits of 20/80, 40/60, 60/40, and 80/20. This setup allows investigation of the model's performance in relation to the amount of training data provided. The results are compared based on accuracy, measured by mean absolute percentage error (MAPE).

3.1 *Preprocessing*

The LSTM model in this study uses only axial stress σ_1 as input, with axial strain ε_1 defining uniform strain steps. Data from numerical tests already have uniformly spaced strain increments, therefore no interpolation is needed No padding or truncation was applied. No normalisation or standardisation was performed to avoid complications and inaccuracies in future predictions caused by scaling and extrapolation. Training data subsets were divided into supervised input-output pairs, where input sequences of length 20 steps are used for predictions. The sequence length was chosen based on Cerek et al. (2025a). The input sequences are fed into the neural network, which iteratively adjusts its weights to predict the output sequence with sufficient accuracy.

3.2 *Architecture tuning*

Preliminary studies explored various architectures of unidirectional LSTM and unidirectional LSTM seq2seq models to identify the most accurate configuration for the task. Architectural tuning was performed using Keras Tuner (O'Malley et al., 2019), specifically employing the Hyperband tuner (Li et al., 2018) due to its efficiency in combining random search with early stopping criteria. The bidirectional architectures were derived from the unidirectional LSTMs to evaluate the added value of processing sequences in both directions, accounting for the doubled number of neurons compared to forward-only models.

An early stopping strategy monitored the validation loss, measured by mean squared error (MSE) (Chicco et al., 2021), halting training if no improvement of more than 1.0 was observed over 20 consecutive epochs. The maximum number of epochs during tuning was set to 100. The validation split was set to 20%.

The following hyperparameters were varied within specified ranges throughout the tuning process:

- Number of hidden LSTM layers: 1 to 3;
- Number of neurons per hidden layer: 32 to 128 (step 32);
- Learning rate: 1e-4 to 1e-2;
- Activation functions for output: ReLU, tanh, sigmoid, ELU, softplus;
- Recurrent activation functions: sigmoid, tanh.

3.3 *Final architectures*

The LSTM architectures used in this study feature a two-dimensional input layer that defines both the length of the input sequence and the number of features. Each model is trained using input sequences of 20 steps and a single feature—axial stress levels. The output layer in all four models consists of one dense neuron with linear activation function. All models employ the Adam optimiser (Kingma & Ba, 2014) and use mean squared error (MSE) as the loss function. No regularisation techniques were applied to the network architectures. A summary of the architectures is presented in Table 1.

The unidirectional LSTM and BiLSTM models use the exponential linear unit (ELU) activation function (Clevert et al., 2015), while the seq2seq models employ the softplus activation function (Dugas et al., 2000). Given the similarity in the shape of these two functions, their use in the respective architectures appears justified. None of the models included more than one hidden LSTM layer. As expected, the number of trainable parameters increases with the model's complexity.

The seq2seq architecture follows an encoder–decoder structure. First, the encoder LSTM processes the entire input sequence and compresses its information into a fixed-size vector representation. This vector serves as a summary of the input sequence. Then, the decoder LSTM uses this compressed representation to generate an output sequence, predicting each data step one by one.

Training is limited to a maximum of 5,000 epochs. An early stopping strategy is implemented to prevent overfitting, terminating the training if the validation loss does not improve by more than 1.0 over 500 consecutive epochs (Prechelt, 1998).

The neural network was implemented and executed using Python (Van Rossum and Drake 2009), with the TensorFlow framework (Abadi et al. 2015) and the Keras API (Chollet et al. 2015). Computational tasks were carried out on a workstation equipped with an AMD Ryzen Threadripper PRO 7975WX 32-core CPU at 4.00 GHz, 512 GB of RAM, and SSD storage. The system featured an NVIDIA RTX A6000 GPU with 48 GB of VRAM, however no GPU was applied for training in this study.

Table 1. Final architectures used for predicting results of CRS test

Model type	Activation function for output	Recurrent activation function	Number of hidden LSTM layers	Total number of LSTM units per hidden layer	Number of trainable parameters	Learning rate
LSTM	ELU	sigmoid	1	96	37,729	0.00210
BiLSTM	ELU	sigmoid	1	2 x 96 = 192 (forward/backward direction)	75,457	0.00210
LSTM seq2seq	softplus	sigmoid	2 (1 encoder + 1 decoder)	64	49,985	0.00863
BiLSTM seq2seq	softplus	sigmoid	2 (1 encoder + 1 decoder)	2 x 64 = 128 (forward/backward direction)	132,737	0.00863

Table 2. Training metrics and performance evaluation of the models

Train/test split	20/80		40/60		60/40		80/20	
Model type	Training time (s)]	MAPE (%)	Training time (s)]	MAPE (%)	Training time (s)]	MAPE (%)	Training time (s)]	MAPE (%)
LSTM	146.4	168.6	164.7	19.9	492.1	1.9	230.1	2.7
BiLSTM	213.4	136.6	140.2	20.1	285.0	1.0	384.5	1.2
LSTM seq2seq	214.4	252.6	362.5	40.6	448.2	2.2	648.7	0.5
BiLSTM seq2seq	213.1	68.5	164.4	15.8	255.2	0.9	655.4	2.5

3.3.1 *Postprocessing*

The postprocessing phase involves predicting the missing part of the compression curve. The model used the last input sequence from the training subset Ω_{train} to predict the first data point or data sequence of the predicted subset Ω_{predict} and continued until the predicted subset is filled with data points. Subsequently, MAPE was calculated, as outlined in Equation (1). The predicted output label $\hat{y}_i$ from Ω_{predict} is subtracted from the actual value y_i belonging to the test subset Ω_{test}.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \cdot 100\,\% \quad (1)$$

Additionally, the time required to complete the supervised learning process of the LSTM network is considered during the evaluation phase.

4 RESULTS

Figure 2 presents a comparison of the achieved MAPEs for each model type across different train/test split ratios. It becomes apparent that all models perform similarly well when trained with 60% or 80% of the data, as indicated by their comparable accuracy levels. In contrast, no clear pattern emerges regarding training times, as shown in Table 2, which summarises both the required training times and the corresponding MAPEs.

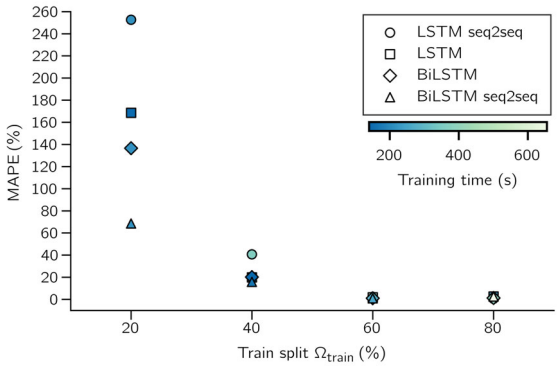

Figure 2. Achieved MAPEs for investigated train splits with required training time.

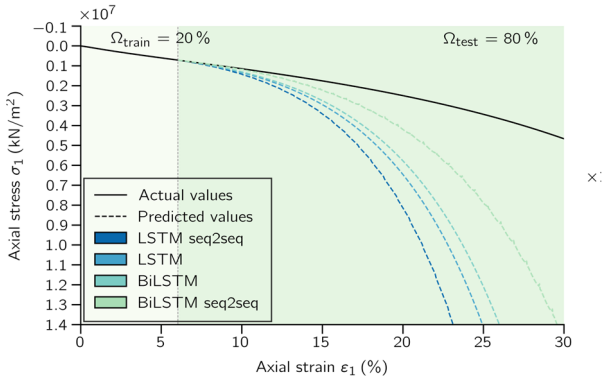

Figure 3. Actual and predicted compression curve $(\sigma_1 - \varepsilon_1)$ using train/test split of 20/80.

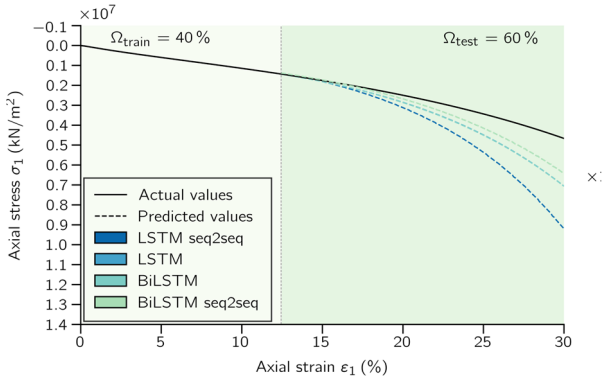

Figure 4. Actual and predicted compression curve $(\sigma_1 - \varepsilon_1)$ using train/test split of 40/60.

The models trained with 20/80 and 40/60 train/test splits fail to deliver satisfactory accuracy. The compression curves predicted by all models for the 20/80 split are shown in Figure 3. Although the BiLSTM seq2seq model outperforms the others in this case, its MAPE of 68.5% remains far too high for practical application.

Performance improves considerably with the 40/60 split, as illustrated in Figure 4. The BiLSTM seq2seq model again leads with a MAPE of 15.8%, suggesting that the sequence-to-sequence approach may help reduce error accumulation over longer sequences.

Figure 5 and Figure 6 present the results for the 60/40 and 80/20 splits, respectively. Both experiments yield consistently good performance, with only minor variations in MAPE across different architectures. These findings suggest that using at least 60% of the data for training is generally sufficient, regardless of the selected model type.

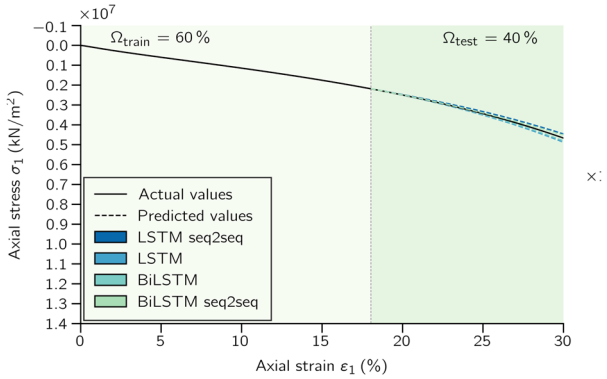

Figure 5. Actual and predicted compression curve $(\sigma_1 - \varepsilon_1)$ using train/test split of 60/40.

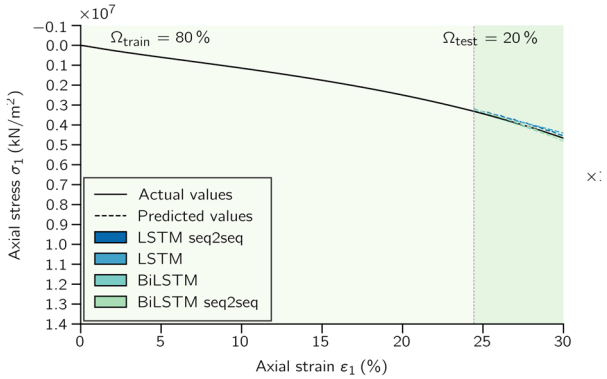

Figure 6. Actual and predicted compression curve $(\sigma_1 - \varepsilon_1)$ using train/test split of 80/20.

5 CONCLUSIONS

This study evaluated the predictive performance of various LSTM-based neural network architectures, including unidirectional LSTM, BiLSTM, and their sequence-to-sequence (seq2seq) extensions, for modelling axial stress-strain relationships in compression tests. The models were trained on different proportions of available data, and their accuracy was assessed using the mean absolute percentage error (MAPE). The results suggest that satisfactory prediction accuracy can be achieved when at least 60% of the data is used for training, regardless of the specific model architecture. Among the tested models, the BiLSTM seq2seq architecture consistently showed strong performance, particularly in cases involving longer sequences, indicating its potential to reduce error accumulation.

This study's findings should be viewed within the scope of its limitations. The neural networks were trained on a single synthetic dataset, which limits the statistical generalisability of the results. As such, the conclusions primarily reflect a comparative analysis of model architectures rather than a broad evaluation of predictive reliability across different data sets. Furthermore, no regularisation methods were applied during training, which could affect model robustness when applied to more diverse or noisy datasets.

Future research could address these limitations by testing the models on a wider range of numerical or experimental

datasets to assess their generalisability. The introduction of regularisation techniques, such as dropout, may help enhance model stability. Additionally, extending the approach to multivariate inputs or investigating advanced architectures like attention mechanisms or transformer-based models could further improve prediction accuracy and broaden the application scope of data-driven stress-strain modelling.

6 ACKNOWLEDGEMENTS

This project was supported by the Federal Ministry of Transport through funding program IHATEC under the code 19H24001.

7 REFERENCES

Abadi, M., Agarwal, A., Barham, P., et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. https://doi.org/10.5281/zenodo.4724125

Cerek, K., Grabe, J. 2023. Numerical simulation and optimization of dike geometry using multi objective evolutionary algorithm NSGA II. https://doi.org/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. In Proceedings of the 17th Pan American Conference on Soil Mechanics and Geotechnical Engineering & 2nd Latin American Regional Conference of the IAEG, La Serena, Chile.

Cerek, K., Gupta, A., Dao, D.A., Hadjiloo, E., and Grabe, J. 2024b. Python implementation of bidirectional LSTM for sequential data processing. Hamburg: TUHH Universitätsbibliothek.

Cerek, K., Dao, D.A., Hadjiloo, E., and Grabe, J. 2024c. Dataset of simulated CRS tests for advanced soil parameter identification. https://doi.org/10.15480/882.9435

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

Cerek, K., Hadjiloo, E., and Grabe, J. 2024e. Application of optimization algorithms in geotechnical engineering as decision making support tool. In N. Guerra et al. (Eds.), Geotechnical Engineering Challenges to Meet Current and Emerging Needs of Society, London: CRC Press, pp. 380–383.

Cerek, K., Gupta, A., Dao, D.A., Hadjiloo, E., and Grabe, J. 2025a. Predicting soil stress–strain behaviour with bidirectional long short-term memory networks. Machine Learning and Data Science in Geotechnics 1(1), 59–76. https://doi.org/10.1108/MLAG-08-2024-0007

Cerek, K., Hadjiloo, E., and Grabe, J. 2025b. "Prediction of Soil Parameters Using Long Short-Term Memory Neural Networks." Proceedings of the 5th International Symposium on Frontiers in Offshore Geotechnics (ISFOG2025), Nantes, France. https://doi.org/10.53243/ISFOG2025-55

Chicco, D., Warrens, M.J., and Jurman, G. 2021. The coefficient of determination R squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science 7, e623. https://doi.org/10.7717/peerj-cs.623

Chollet, F., et al. 2015. Keras: The Python deep learning library, Version 2. https://keras.io.

Choudhary, K., DeCost, B., Chen, C., et al. 2022. "Recent Advances and Applications of Deep Learning Methods in Materials Science." npj Computational Materials 8(1): 59. https://doi.org/10.1038/s41524-022-00734-6

Clevert, D. A., Unterthiner, T., and Hochreiter, S. 2015. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)." https://doi.org/10.48550/arXiv.1511.07289

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., and Garcia, R. 2000. Incorporating second order functional knowledge for better option pricing. In Advances in Neural Information Processing Systems, pp. 472–478.

Graves, A., and Schmidhuber, J. 2005. "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures." Neural Networks 18(5–6): 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

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 154: 105120. https://doi.org/10.1016/j.compgeo.2022.105120

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 9(8): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kingma, D.P., and Ba, J. 2014. Adam: A method for stochastic optimization. http://arxiv.org/pdf/1412.6980v9

LeCun, Y., Bengio, Y., and Hinton, G. 2015. "Deep Learning." Nature 521: 436–444. https://doi.org/10.1038/nature14539

Levine, E.R., Kimes, D.S., and Sigillito, V.G. 1996. "Classifying Soil Structure Using Neural Networks." Ecological Modelling 92(1): 101–108. https://doi.org/10.1016/0304-3800(95)00199-9

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. 2018. Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research 18, 1–52. http://jmlr.org/papers/v18/16-558.html.

Machaček, J., Staubach, P., Tavera, C. E. G., et al. 2022. "On the Automatic Parameter Calibration of a Hypoplastic Soil Model." Acta Geotechnica 17(11): 5253–5273. https://doi.org/10.1007/s11440-022-01669-4.

Niemunis, A., and Herle, I. 1997. "Hypoplastic Model for Cohesionless Soils with Elastic Strain Range." Mechanics of Cohesive-Frictional Materials 2(4): 279–299. https://doi.org/10.1002/(SICI)1099-1484(199710)2:4

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. 2019. KerasTuner. https://github.com/keras-team/keras-tuner.

Prechelt, L. 1998. Early stopping – but when? In Neural Networks: Tricks of the Trade, edited by G.B. Orr and K.-R. Müller, 55–69. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-49430-8_3

Qi, C., and Fourie, A. 2018. "A Real-Time Back-Analysis Technique to Infer Rheological Parameters from Field Monitoring." Rock Mechanics and Rock Engineering 51(10): 3029–3043. https://doi.org/10.1007/s00603-018-1513-2

Rauter, S., and Tschuchnigg, F. 2021. "CPT Data Interpretation Employing Different Machine Learning Techniques." Geosciences 11(7): 265. https://doi.org/10.3390/geosciences11070265

Rezania, M., Javadi, A. A., and Giustolisi, O. 2008. "An Evolutionary-Based Data Mining Technique for Assessment of Civil Engineering Systems." Engineering Computations 25(6): 500–517. https://doi.org/10.1108/02644400810891526

Sak, H., Senior, A., and Beaufays, F. 2014. "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling." In Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH), Singapore, 338–342. https://doi.org/10.21437/Interspeech.2014-80

Sagheer, A., and M. Kotb. 2019. "Time Series Forecasting of Petroleum Production Using Deep LSTM Recurrent Networks." Neurocomputing 323(3): 203–213. https://doi.org/10.1016/j.neucom.2018.09.082

Schuster, M., and Paliwal, K. K. 1997. "Bidirectional Recurrent Neural Networks." IEEE Transactions on Signal Processing 45(11): 2673–2681. https://doi.org/10.1109/78.650093

Staudemeyer, R.C., and E.R. Morris. 2019. "Understanding LSTM – A Tutorial into Long Short-Term Memory Recurrent Neural Networks.". https://doi.org/10.48550/arXiv.1909.09586

Sutskever, I., Vinyals, O., and Le, Q. V. 2014. "Sequence to Sequence Learning with Neural Networks." https://doi.org/10.48550/arXiv.1409.3215

Van Rossum, G., and Drake, F.L. 2009. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace. 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 1(3): 251–271. https://doi.org/10.1002/(SICI)1099-1484(199607)1:3

Zhang, N., Shen, S., Zhou, A., and Jin, Y. 2021. “Application of LSTM Approach for Modelling Stress–Strain Behaviour of Soil.” Applied Soft Computing 100(10): 106959. https://doi.org/10.1016/j.asoc.2020.106959