Available online at www.eccomasproceedia.org
Eccomas Proceedia COMPDYN (2025) 229-242

ECCOMAS
Proceedia

COMPDYN 2025
10th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis (eds.)
Rhodes Island, Greece, 15-18 June 2025

A MACHINE LEARNING APPROACH FOR GROUND RESPONSE ANALYSIS

Leonidas Alexandros S. Kouris[1], Kyriaki Gkoktsi[1], Konstantinos I. Gkatzogias[1], Daniel A. Pohoryles[1], Georgios Tsionis[1] and Elisabeth Krausmann[1]

[1] European Commission, Joint Research Centre
Via Enrico Fermi 2749 I-21027 Ispra (VA) Italy
e-mail: {Leonidas.KOURIS, Kyriaki.GKOKTSI,Konstantinos.GKATZOGIAS,Daniel.POHORYLES, Georgios.TSIONIS, Elisabeth.KRAUSMANN}@ec.europa.eu

Abstract

Ground response is an integral part of the dynamic response of the overlying infrastructure. When a large number of ground response analyses is needed, for instance in the case of seismic hazard analysis on a large geographical scale or, when assessing uncertainty in seismic structural response due to site effects, computational efficiency is critical. Machine-learning (ML) models can predict response at a very low cost and require fewer parameters. An ML procedure is proposed here to model the ground response using artificial neural networks (ANNs) and deep-learning methods. Ground response is affected by various parameters, including the mechanical properties of the underlying soil strata and their geometry. In the current study, soil response is assumed elastic while seismic excitation is modelled through harmonic waves with unit amplitude and various frequencies in a given range of interest. The physical soil model is approximated with a recurrent neural network (RNN), which is combined with long short-term memory (LSTM) units to handle time-series data. The RNN model receives as input the harmonic waves and gives as output the estimated ground acceleration responses. The latter are compared against the target output signals derived from conventional ground response analyses to evaluate the accuracy of the model. To train the RNN, the ADAptive Moment estimation (ADAM) optimisation algorithm is applied, given its suitability for treating large datasets. To increase the computation efficiency of the RNN algorithm, the full-length input signal is split in several batches of fixed length. To tune the ML parameters and assess their influence, analyses are conducted for two loss metrics, three activation functions, and three batch lengths. It is shown that the computational demands of traditional analytical methods could be overcome by ML techniques that can efficiently process complex and large datasets of ground responses.

Keywords: Ground response, Machine Learning, Recurrent Neural Networks, ADAM algorithm.

ISSN:2623-3347
Peer-review under responsibility of the organizing committee of COMPDYN 2025.
doi: 10.7712/120125.12404.26203

1 INTRODUCTION

The application of Artificial Neural Networks (ANNs) has become a powerful tool in the field of civil engineering as a reliable approach that efficiently treats the trade-off between high accuracy and fast numerical procedures e.g. [1], [2]. These computational models compiled from neurons are inspired by the human brain's neural networks and are capable of identifying complex patterns and relationships within data.

The key step to develop a reliable model is the compilation of large datasets either experimentally e.g. [3], [4] or analytically e.g. [5], [6]. Then, ANNs can learn from the data and once validated, can replace traditional analytical methods, which would require substantial computational time for modeling nonlinear and complex behaviour [7]. While training a reliable model (e.g. the dynamic response of a complex structure) requires extensive computations, once the model trained, its application to predict structural response would be very fast. Therefore, the use of ANNs in civil engineering has been gradually extended to cover several aspects from the prediction of failure modes of structural members [3] to heating energy demand of buildings [8] and reliability analysis [9].

In the context of dynamics, ANNs have been used to predict how structures respond to various dynamic loads such as earthquakes [10], wind [11], [12], and traffic-induced vibrations [13]. A detailed reference of ANN application in dynamics can be found in [14] updated here with some recent and relevant to the context of this study papers. The use of ANNs in this domain offers several advantages: ANNs can provide rapid predictions [15], [16], which is crucial for real-time monitoring [10] and control [17], [18] applications, minimising data requirements [19]. Additionally, ANNs can be employed to identify damage [20] or changes in structural behavior over time [21], enhancing the capability for structural health monitoring. Data-driven [22] and physics-informed [23], [24] training of the ANN have been carried out with merits and drawbacks.

Dynamic response of soil includes several complexities and uncertainties, while there is often lack of sufficiently available data on soil properties, such as the shear wave velocity [25], which is critical for estimating seismic wave amplification. Machine learning (ML) models offer the possibility to estimate critical soil parameters [25] and run multiple analyses for various soil property combinations with minimal computational cost [26]. ANNs have been applied to estimate soil amplification considering time-dependent and time-independent multiple layer perceptrons to train the ANN model with response spectra, real and synthetic ground motions [26].

In terms of applied methods, the recurrent neural networks (RNNs) are commonly used, e.g. [11], [26], [27], as they offer the ability to handle better sequential data. RNNs have been implemented in several applications, either in the frequency domain e.g. [10], [26] or in the time domain [26], [28], [29]. Other options can be convolutional neural networks (CNNs) [30], [31] or graphical neural networks (GNNs) [32].

The advantage of RNNs is that the input for every neural unit (cell) includes time-delayed parts (history) of the input and output signals (recursive process), together with a hidden layer that is combined with a dense layer form the structure of RNNs. RNNs fused with long short-term memory (LSTM) [33] can be used effectively in long time series data. The LSTM layer is an improved version of RNN recursive process for dealing with long time-series data through a selective memory process. The LSTM cell is designed with an architecture that allows to maintain and update information over long durations. This is done through interacting layers and memory gates without vanishing or exploding the gradients of the loss function during training of the network. In every cell, apart from the input and output gates, two additional gates exist, which retain recent memory (‘*update gate*’) and discard the old one (‘*forget gate*’). These

gates control the gradients of the loss function by not allowing them to vanish or explode and to keep the relevant long-term information.

The LSTM ability to control memory through this gate structure makes it highly effective for dynamic response and time-series prediction [34], where the effect of long-term dependencies is highly evident. The RNN weights and biases are obtained through a training process, which is based on an optimisation process that minimises a selected loss function, such as the mean square error (MSE). ADAM [35] has been found to be an adequate optimisation algorithm for RNN training [12], [30].

In this study, RNNs are used to estimate soil response in the time domain. The performance of this ML algorithm is evaluated in terms of its capacity to capture the response of an elastic soil profile subjected to harmonic wave excitations at the bedrock, considering a range of excitation frequencies. The presented research work forms a first step towards considering nonlinear soil dynamics under real seismic records at bedrock.

2 PHYSICAL SOIL MODEL AND GROUND RESPONSE ANALYSIS

2.1 Soil profile

For the numerical evaluation of the proposed method, a soil profile is assumed with a depth of 66 m above bedrock. The considered soil profile comprises three soil layers with the properties reported in Table 1. As shown in Fig. 3, the shear wave velocity, V_s, is assumed to increase with depth, taking values in the range of [150, 550] m/s (see also Table 1). A constant mass density at ρ=2200 kg/m^3 and a constant damping ratio at 2% are assumed across the entire soil profile. The assumed properties of the bedrock are V_s=1100 m/s and ρ=2396 kg/m^3, and the interface between the bedrock and Layer 3 is considered the reference elevation level.

Layer	Depth	Thickness	Shear wave velocity (V_s)	Shear Modulus (G)
	[m]	[m]	[m/s]	[MPa]
1	0.0	51	150	49.55
2	-51.0	6	450	445.91
3	-57.0	9	550	666.11
Bedrock	-66.0	-	1100	2899.16

Table 1: Soil properties.

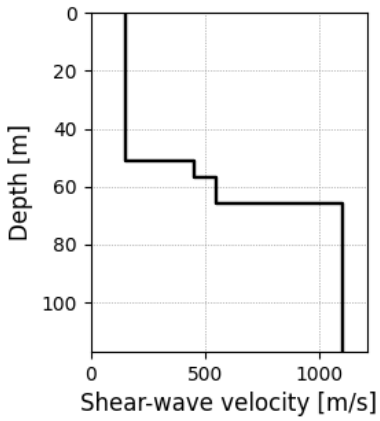


Figure 1: The soil profile.

The last column of Table 1 gives the shear modulus, *G,* of the soil layers and bedrock, estimated from the shear velocity, V_s, and the mass density, ρ, according to Eq. (1).

$$G = \rho \cdot V_s^2 \tag{1}$$

2.2 Soil response

To simulate the soil, an elastic 1D model is developed using the SeismoPy code [36], [37] in Python considering the elastic properties of the soil presented in Table 1. Linear ground response analyses are performed for a series of 100 harmonic waves, $x(t)$, applied to bedrock, according to the expression

$$x(t) = A \cdot \sin(2\pi f t) \tag{2}$$

In Eq. (2), *A* is the acceleration amplitude equal to 1g, with g=9.81 m/s^2 being the acceleration of gravity, *f* is the excitation frequency ranging from 0.05 to 5 Hz, and *t* is the time-vector with a duration of 20s, sampled at dt = 0.01s, and pertaining to 2000 sample points. The acceleration response signals at the ground surface are obtained from the 1D ground response analyses results (i.e., the "target" signals for the ML algorithm). Fig. 2 plots together the input (excitation) and output (ground response) signals for six excitation frequencies and a duration of 0.3s.. From this figure, the input/output differences in terms of amplitude and phase are readily observed. Fig. 2 also indicates the resonance phenomenon, which occurs at excitation frequencies closer to the natural frequencies of the soil profile. This phenomenon is manifested with the increase of the ground response amplitude relatively to the amplitude of the excitation signal at bedrock.

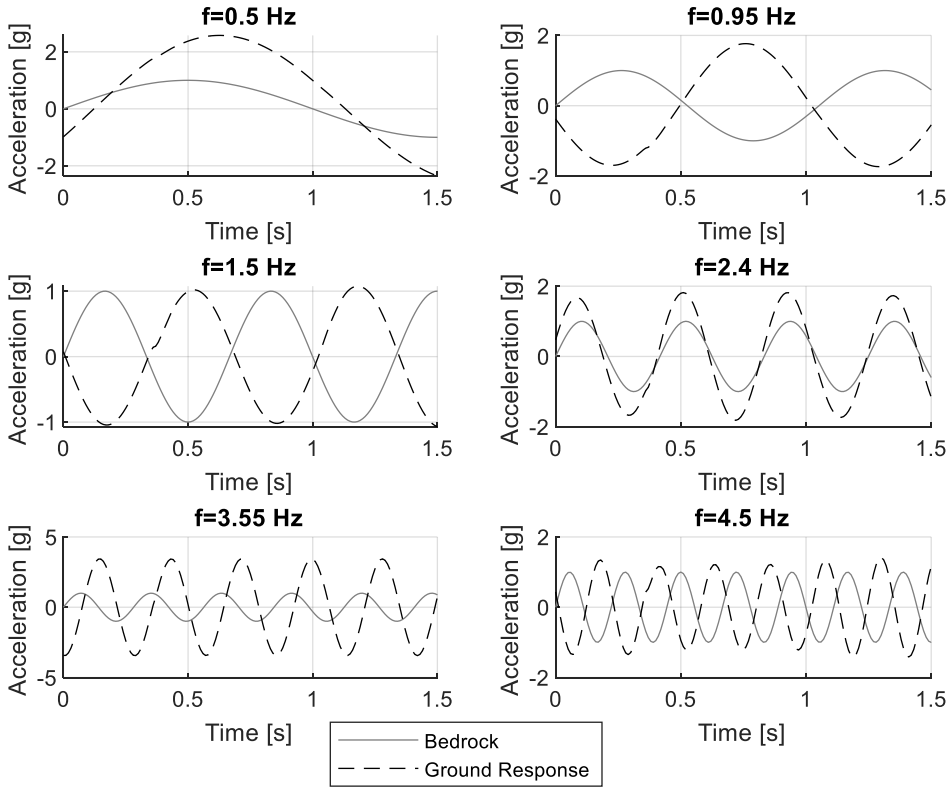


Figure 2: Bedrock excitation overlaid with ground acceleration response for six excitation frequencies (for the first 1.5s).

To better capture the resonance phenomenon, the transfer function of the soil profile is computed from the ratio of the output acceleration response signals on the ground surface over the input signals at the bedrock in the frequency domain and plotted in Fig. 3. The transfer function carries the information of the natural frequencies of the soil profile, which are retrieved from the location of the peaks in the left panel of Fig. 3. Table 2 presents the estimated natural frequencies, f_n, at the first five modes of vibration of the considered soil profile, which are between 0.70 Hz and 6.30 Hz. The latter justifies the selected frequency range, f= [0.5, 5] Hz, for the excitation signals at bedrock. For the input/output signals shown in Fig. 2, the highest signal amplification is observed at f=3.55 Hz, which is close to the natural frequency of the soil profile, f_n=3.50 Hz, at its 3rd mode of vibration.

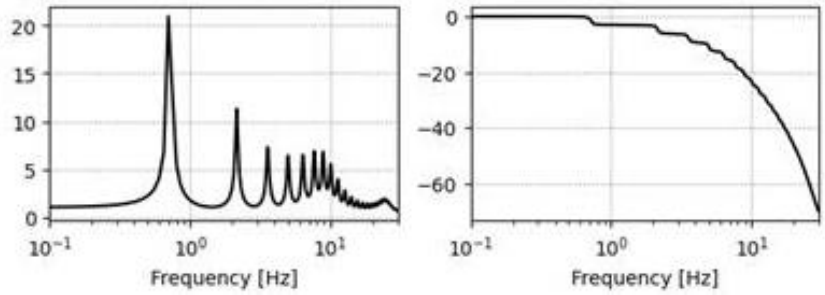


Figure 3: Transfer function of the soil profile; amplitude (left) and phase (right).

	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
f_n (Hz)	0.70	2.10	3.50	4.90	6.30

Table 2: Natural frequencies of the considered soil profile at the first five modes of vibration.

3 RECURRENT NEURAL NETWORK (RNN) WITH LONG SHORT-TERM MEMORY (LSTM)

3.1 Architecture

An ANN typically comprises an input layer, one or more hidden layers and an output layer. RNN offers the ability of a feedback mechanism to convey memory in sequence data (Fig. 4a) between time steps. To achieve this, the hidden layer consists of a recursive procedure where memory loops after each time step connected with a dense layer. Therefore, the key characteristic of RNN is that it feeds an input layer to the next moment neuron to establish its output layer.

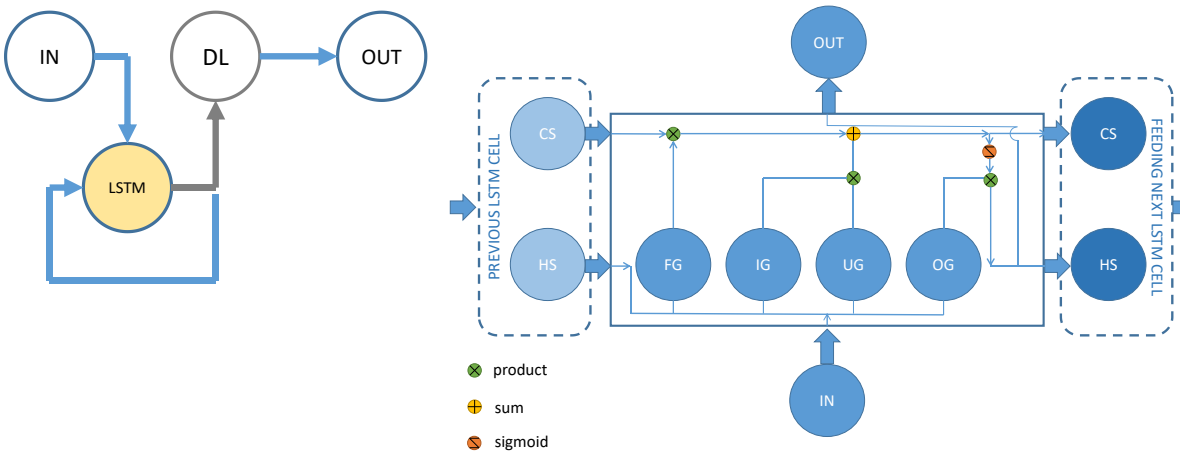
Training of an ANN involves the update of the weights of the neurons. A single forward pass through the network, followed by backpropagation through time (BPTT) are performed to train the RNN network [38].

Fig. 4b presents the LSTM architecture, emphasising the flow of information within the LSTM cell, i.e., the incoming information from the previous state and the outgoing information that feeds the next step. The cell state memory retains the long-term memory of the network, allowing information to persist over time.

LSTMs use gates to control the flow of information and determine the cell state (Fig. 4b). The forget gate determines what information to discard from the previous hidden state using an activation function. The hidden state is the output of the LSTM cell at each time-step and serves as input to the next cell in the time-sequence. It encapsulates the current output information of the cell. The input gate decides which new information to add to the cell state. The update gate

adjusts the information in the LSTM unit, with the new cell state consisting of two components: one selectively forgets information based on previous cell state, and the other is derived by multiplying it with the candidate cell state from input gate.

The new cell state is updated by combining the previous cell state that is scaled by the forget gate, and the update gate that is scaled by the input gate. The output gate determines the next hidden state by combining data from current and previous states. It controls the information to be outputted from the cell state, using a combination of a sigmoid function of the updated cell state. At any time, the inputs to the LSTM cell are the previous hidden states and the current input vector.



(a) (b)

Figure 4: (a) Schematic representation of the RNN structure that includes the input (IN), the LSTM, the dense (DL), and the output (OUT) layers. (b) The LSTM unit structure to handle information between time steps and determine current state (CS) and hidden state (HS) with four gates included: the forget gate (FG), the input gate (IG), the update gate (UG) and the output gate (OG).

3.2 Recurrent neural network (RNN) with long short-term memory (LSTM) and dense layer

The model is simulated in Python using the Keras library. The input is 'sliced' in a given number of batches, B, of fixed length, L, to reduce the computational cost of processing an entire harmonic wave of N=2000 sample points in one step (i.e., B analysis steps are performed, pertaining to B time-sequences of length L, with N=B×L). This approach improves the ability of the model to capture patterns and relationships among the batches [39]. To this end, the input layer receives the excitation input at bedrock as a 3D array whose first dimension is equal to the time-domain support of the full-length signal (i.e. N), the second dimension is equal to the length of the batches (i.e. L), and the third one is equal to the number of features (e.g. soil profiles). In the numerical example herein, the third dimension is equal to one given that a single soil profile is considered.

Three LSTM and two dense layers are applied:

- **LSTM layer 1**with 128 units. This layer processes the input sequence and outputs a sequence of values with matrix dimensions ($N \times L \times 128$).
- **LSTM layer 2** with 96 units. This layer processes the output of the first LSTM layer and outputs a sequence of values with matrix dimensions ($N \times L \times 96$).

- **LSTM layer 3** with 64 units. This layer outputs only a single value for each sample in the batch. It processes the output of the LSTM layer 2 and outputs a value with shape ($L \times 64$).
- **Dense layer 1** with 32 units and an activation function. This layer processes the output of the third LSTM layer and outputs a value with shape ($L \times 32$).
- **Dense layer 2** with one unit and no activation function. This layer processes the output of the first dense layer and outputs a single value, i.e. the response of the ground at the surface. Its shape has dimensions ($N \times 1$).

The two dense layers in the model are used to transform the output of the LSTM layer into the final predicted sequence. The dense layer 1 is used to introduce non-linearity into the model and to reduce the dimensionality of the output from the LSTM layer. The dense layer 2 is used to produce the final predicted sequence. The number of units in this layer is equal to the number of features in the input data, which means that the model is predicting one sequence of output.

Using two dense layers instead of one is beneficial for several reasons, i.e.:

- Improved feature extraction: The first dense layer can help to extract more complex features from the output of the LSTM layer, which can improve the accuracy of the model.
- Better handling of non-linear relationships: The ReLU activation function in the first dense layer can help handle non-linear relationships between the input data and the output data.
- More flexible output: The second dense layer can produce a more flexible output, which can be beneficial for sequence-to-sequence problems where the output time-sequence can have a different length or structure than the input time-sequence.

However, using two dense layers also increases the complexity of the model and can lead to overfitting if the model is not regularised properly. In general, the choice of the number and type of dense layers depends on the specific problem and the characteristics of the data.

4 TUNING THE RECURRENT NEURAL NETWORK (RNN) WITH LONG SHORT-TERM MEMORY (LSTM) PARAMETERS

4.1 Loss metric

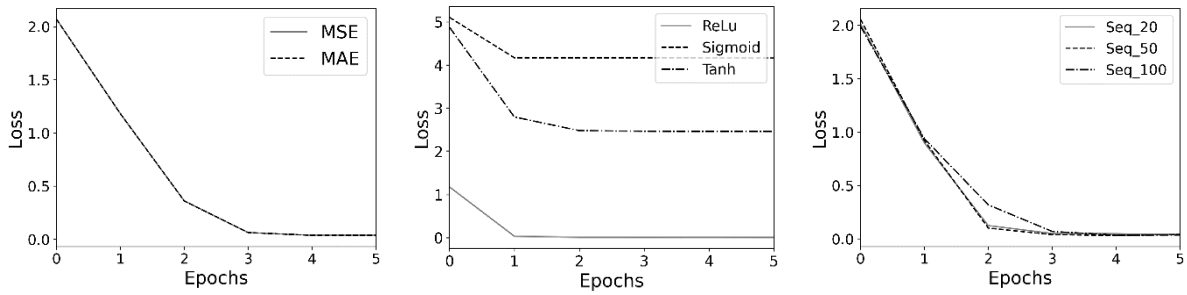
The choice of the loss metric influences the performance of the optimisation algorithm that minimises the loss function. Two loss metrics have been assessed in this study, i.e.: (i) the mean square error (MSE), and (ii) the mean absolute error (MAE), which are computed for the predicted/estimated ground response signals with respect to the pertinent target signals derived from the 1D ground response analyses.

For the two considered loss metrics, the performance of the RNN network and the ADAM optimisation algorithm is compared in Fig. 5a in terms of validation loss versus epochs. Fig. 5a shows that the loss curves coincide for the two loss metrics, confirming that the RNN network using the ADAM algorithm has the same performance for both loss metrics.

4.2 Activation function

The Rectified Linear Unit (ReLU) activation function is commonly used in deep learning models because it is computationally efficient and helps to avoid the vanishing gradient problem. ReLU maps all negative values to 0 and all positive values to the same value.

$$ReLU(x) = max(0, x) \tag{3}$$



(a) (b) (c)

Figure 5: Loss vs. epochs: (a) for two loss metrics MSE and MAE, (b) for three activation functions (ReLu, sigmoid, tanh), and (c) three batches length with L_1=20 (Seq_20), L_2=50 (Seq_50) and L_3=100 (Seq_100) samples.

ReLU activation function has been applied to all hidden layers. However, a number of different activation functions could be applied for Dense Layer 1. In this study, the performance of the RNN algorithm has been assessed with two additional activation functions, i.e.: (i) the sigmoid, and (ii) the hyperbolic tangent given in Eq. (4) and Eq. (5), respectively

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{4}$$

$$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1} \tag{5}$$

For the three activation functions, the obtained results from the RNN analyses are compared in Fig. 5b in terms of validation loss versus epochs. Fig. 5b confirms the better performance of the linear activation function that yields significantly lower loss values compared to the sigmoid and the hyperbolic tangent activation function. This better performance is consistently observed for all examined epochs.

4.3 Batch length

To increase the computation efficiency of the ML algorithm, the input signal to the RNN model is split in *B* batches of length *L* (see also sub-section 3.2). The length of the batch plays an important role in the model as the longer the batch is (i.e. higher *L* value), the more complex patterns in data can be captured. This comes at the cost of increased computational time per batch, although fewer analysis steps are required) to process the entire input signal of length *N*. The number of batches, *B*, governs the number of analysis steps. However, there is the possibility of data overfitting as the batch length increases. On the other hand, a shorter-length batch can lead to a faster convergence despite that the number of analysis steps are increasing, but complex patterns in data may not be adequately captured. To evaluate the trade-off between computational efficiency of the RNN algorithm versus accuracy in data pattern identification, a parametric analysis was carried out using three different pairs of number of batches and associated length: (i) B_1=*100* batches, L_1=20 samples, (ii) B_2=*40* batches, L_2=50 samples, and (iii) B_3=*20* batches, L_3=100 samples. Fig. 5c compares the derived error plots in terms of loss versus epoch. This figure proves that the most accurate analysis for the longer batch with L_3=100 samples yields the same loss value with the other cases after three epochs at a reasonable analysis duration.

5 ML-BASED SOIL MODEL AND APPROXIMATION OF GROUND RESPONSE ANALYSIS

The dataset, consisting of 100 input and output signals (i.e. bedrock harmonic waves and ground acceleration responses), each with 2000 samples, was divided into training andvalidation sets. Each signal was split with a ratio of 80% for training and 20% for validation. Initially, the model is trained on the first input/output dataset at one excitation frequency and subsequently updated for the remaining input/output datasets at the other frequencies.

The analysis resulted in very good estimates as can be observed in the predicted signals presented in Fig. 6 versus the target ground response signals derived from the 1D ground response analyses. To assess the accuracy of the ML-based soil response, the MSE and the MAE are computed between the predicted and the target output signal for each excitation frequency. The obtained MSE and MAE results are presented in Fig. 7 with respect to the vector of excitation frequencies. As the MSE yields lower values compared to the MAE, thus it is selected as the preferred loss metric thereafter. The MSE is well below 0.1 g^2 in almost all excitation frequencies except those around the natural frequencies of the soil profile due to the resonance phenomenon and the amplification of the soil responses (see also section 2.2 and Table 2). For the first natural frequency at f_n=0.70 Hz, the MSE reaches 0.5 g^2, while for the second one at f_n=2.10 Hz the MSE is below 0.1 g^2. As the ground responses were not scaled, small differences are expected between the actual and estimated response in highly amplified signals, resulting in higher MSE values. Similar observations hold for the MAE versus the excitation frequencies in Fig. 7, attaining its maximum value around the resonant frequencies of the soil profile.

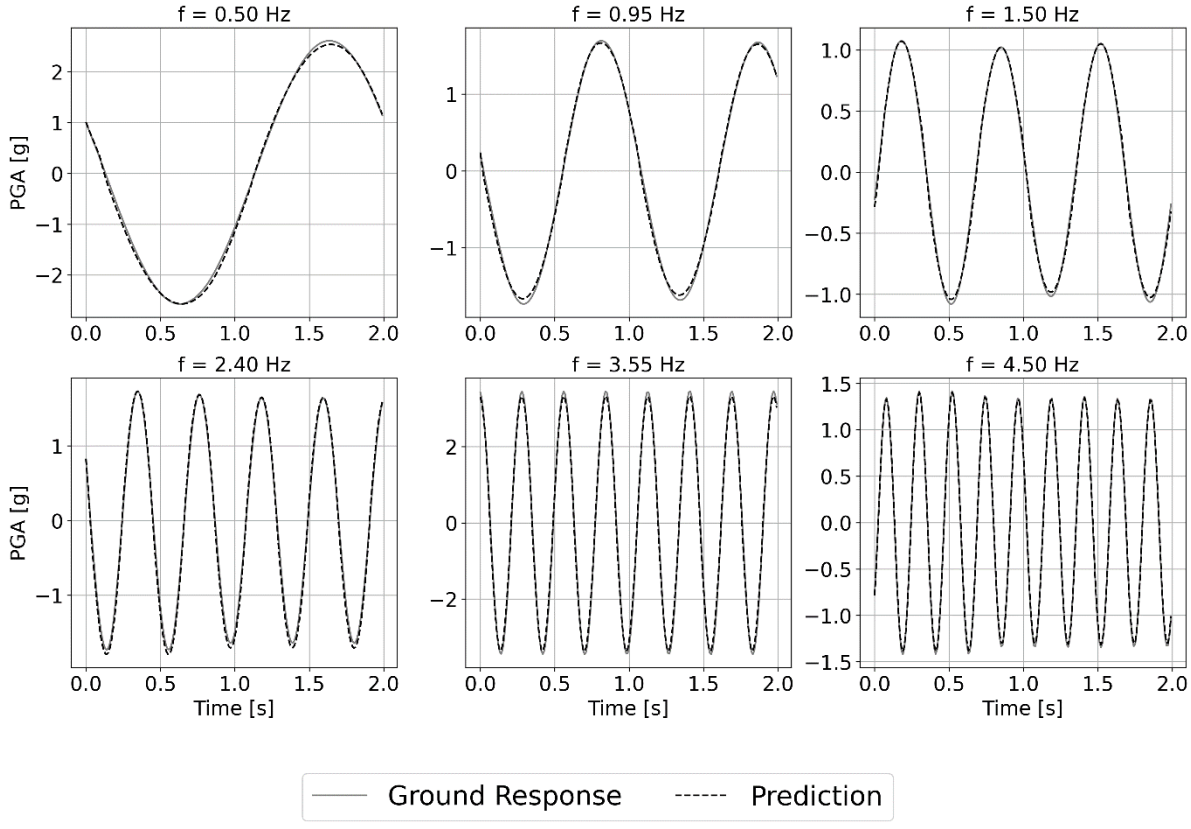


Figure 6: Comparison of the numerical and RNN prediction responses.

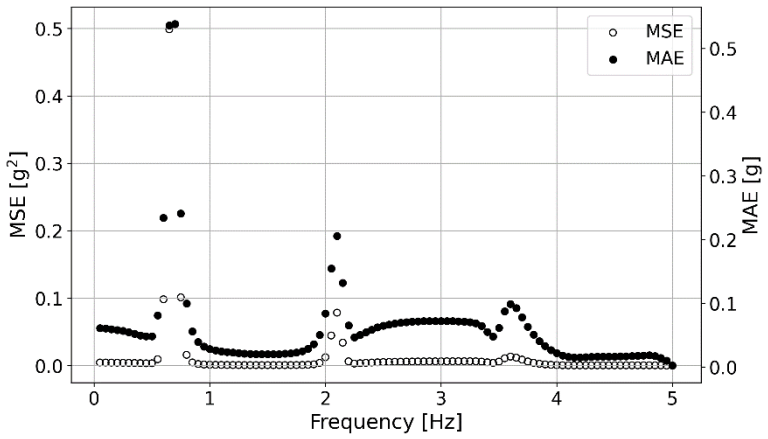


Figure 7: Error in terms of MSE and MAE versus the excitation frequencies of the input signals.

6 VERIFICATION TEST

A verification test is carried out using unknown harmonic signals to the RNN model, pertaining to five different frequencies (see Eq. 2) from those used to train and validate the model. The frequencies of the verification signals are listed in Table 3.

Excitation	1	2	3	4	5
f (Hz)	0.062	0.169	0.275	0.381	0.487

Table 3: Frequencies of the verification signals.

Fig. 8 plots together the ground response predictions and the target output, while the associated MSE is given at the bottom right side of the figure. The computed MSE is of the same order as in the training/validation datasets, not exceeding the value of 0.005 g^2.

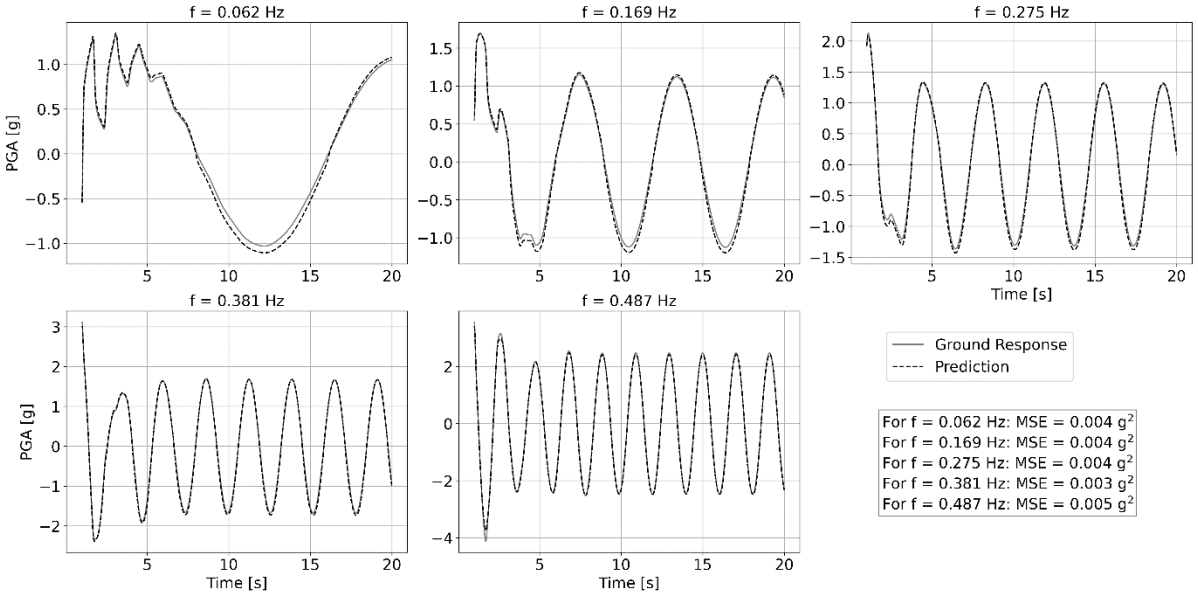


Figure 8: Verification test of the model.

7 CONCLUSIONS

In this paper, RNNs with LSTM units have been used to simulate the dynamic soil amplification and predict the ground acceleration responses due to excitations at bedrock. The architecture of the employed RNN network comprises three LSTM units and two dense layers. For the numerical evaluation of the ML-based approach, a 66-meter-thick soil profile with three soil layers above bedrock isassumed. Using the SeismoPy code in Python, 1D linear ground response analyses are conducted to derive the target ground acceleration responses. For the execution of those analyses, harmonic excitation signals are applied at the bedrock, considering 100 excitation frequencies around the first five modes of vibration of selected soil profile. A verification test showed the efficiency of the model in terms of both accuracy and time. From the training and validation of the RNN algorithm, the following concluding remarks are drawn:

- The RNN with the LSTM units can effectively capture the response of the soil with a relatively small number of epochs.
- The RNN algorithm is efficient when the model is trained with batches of 100 samples of the excitations, given that the required computation time is acceptable.
- The linear activation function is proved more efficient compared to the sigmoid and the hyperbolic tangent functions.
- The MSE and MAE metrics to estimate loss function presented no difference.

ACKNOWLEDGMENTS

This research work is part of the exploratory research project FROST-QUAKE: “Climate change impact on critical infrastructure in the northern hemisphere: thawing permaFROST in earthQUAKE-prone areas”, of the *European Commission, Joint Research Centre*, 2024 – 2026.The help of Nicolò Tambone in Python analyses is gratefully acknowledged.

REFERENCES

[1] R. Yi, X. Li, S. Zhu, Y. Li, and X. Xu, “A deep leaning method for dynamic vibration analysis of bridges subjected to uniform seismic excitation,” *Soil Dynamics and Earthquake Engineering*, vol. 168, 2023, doi: 10.1016/j.soildyn.2023.107830.

[2] Z. Xu, J. Chen, J. Shen, and M. Xiang, “Recursive long short-term memory network for predicting nonlinear structural seismic response,” *Eng Struct*, vol. 250, 2022, doi: 10.1016/j.engstruct.2021.113406.

[3] K. G. Megalooikonomou and G. N. Beligiannis, “Random Forests Machine Learning Applied to PEER Structural Performance Experimental Columns Database,” *Applied Sciences (Switzerland)*, vol. 13, no. 23, 2023, doi: 10.3390/app132312821.

[4] D. Zhang *et al.*, “Prediction of seismic acceleration response of precast segmental self-centering concrete filled steel tube single-span bridges based on machine learning method,” *Eng Struct*, vol. 279, 2023, doi: 10.1016/j.engstruct.2022.115574.

[5] Y. Liao, R. Lin, R. Zhang, and G. Wu, “Attention-based LSTM (AttLSTM) neural network for Seismic Response Modeling of Bridges,” *Comput Struct*, vol. 275, 2023, doi: 10.1016/j.compstruc.2022.106915.

[6] R. Zhang, Z. Chen, S. Chen, J. Zheng, O. Büyüköztürk, and H. Sun, "Deep long short-term memory networks for nonlinear structural seismic response prediction," *Comput Struct*, vol. 220, pp. 55–68, 2019, doi: 10.1016/j.compstruc.2019.05.006.

[7] B. Ahmed, S. Mangalathu, and J. S. Jeon, "Generalized stacked LSTM for the seismic damage evaluation of ductile reinforced concrete buildings," *Earthq Eng Struct Dyn*, vol. 52, no. 11, 2023, doi: 10.1002/eqe.3869.

[8] A. Veljkovic, D. A. Pohoryles, and D. A. Bournas, "Heating energy demand estimation of the EU building stock: Combining building physics and artificial neural networks," *Energy Build*, vol. 298, 2023, doi: 10.1016/j.enbuild.2023.113474.

[9] V. Papadopoulos, D. G. Giovanis, N. D. Lagaros, and M. Papadrakakis, "Accelerated subset simulation with neural networks for reliability analysis," *Comput Methods Appl Mech Eng*, vol. 223–224, pp. 70–80, 2012, doi: 10.1016/j.cma.2012.02.013.

[10] A. A. Torky and S. Ohno, "Deep learning techniques for predicting nonlinear multi-component seismic responses of structural buildings," *Comput Struct*, vol. 252, 2021, doi: 10.1016/j.compstruc.2021.106570.

[11] P. Modé, C. Demartino, C. T. Georgakis, and N. D. Lagaros, "Short-term extreme wind speed forecasting using dual-output LSTM-based regression and classification model," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 259, 2025, doi: 10.1016/j.jweia.2025.106035.

[12] S. P. Hareendran and A. Alipour, "Prediction of nonlinear structural response under wind loads using deep learning techniques," *Appl Soft Comput*, vol. 129, 2022, doi: 10.1016/j.asoc.2022.109424.

[13] F. Yang, X. Zhang, Y. Han, L. Yang, and L. Wang, "Random Vibration Analysis of Train–Bridge System Based on Parallel Adaptive Enhanced-Surrogate Model," *International Journal of Structural Stability and Dynamics*, 2025, doi: 10.1142/S0219455426501646.

[14] Y. Xie, M. Ebad Sichani, J. E. Padgett, and R. DesRoches, "The promise of implementing machine learning in earthquake engineering: A state-of-the-art review," *Earthquake Spectra*, vol. 36, no. 4, pp. 1769–1801, 2020, doi: 10.1177/8755293020919419.

[15] Y. Liao, H. Tang, R. Li, L. Ran, and L. Xie, "Seismic response prediction and parameters estimation of the frame structure equipped with the base isolation-fluid inerter system (FS-BIFI) based on the PI-LSTM model," *Eng Struct*, vol. 309, 2024, doi: 10.1016/j.engstruct.2024.118077.

[16] H. Taghavi Ganji and E. Seylabi, "An LSTM RNN proposal for surrogate modeling the dynamic response of buried structures to earthquake plane waves in soil half-spaces," *Comput Geotech*, vol. 164, 2023, doi: 10.1016/j.compgeo.2023.105796.

[17] Z. Wang, F. Ubertini, and S. Laflamme, "Ensemble of long short-term memory recurrent neural network for semi-active control of tuned liquid wall damper," *Eng Struct*, vol. 270, 2022, doi: 10.1016/j.engstruct.2022.114771.

[18] H. Zhang, L. Wang, and W. Shi, "Seismic control of adaptive variable stiffness intelligent structures using fuzzy control strategy combined with LSTM," *Journal of Building Engineering*, vol. 78, 2023, doi: 10.1016/j.jobe.2023.107549.

[19] H. Pak and S. G. Paal, "A real-time structural seismic response prediction framework based on transfer learning and unsupervised learning," *Eng Struct*, vol. 323, 2025, doi: 10.1016/j.engstruct.2024.119227.

[20] A. Atia, M. Vafaei, S. C. Alih, and K. F. Tee, "Novel Deep Learning-Based Method for Seismic-Induced Damage Detection," *Arab J Sci Eng*, 2024, doi: 10.1007/s13369-024-09316-8.

[21] S. Al-Subaihawi, J. Ricles, S. Quiel, T. Marullo, and F. Malik, "Real-time hybrid simulation of structural systems with soil-foundation interaction effects using neural networks," *Earthq Eng Struct Dyn*, vol. 53, no. 15, pp. 4688–4718, 2024, doi: 10.1002/eqe.4236.

[22] A. Kundu, S. Ghosh, and S. Chakraborty, "A long short-term memory based deep learning algorithm for seismic response uncertainty quantification," *Probabilistic Engineering Mechanics*, vol. 67, 2022, doi: 10.1016/j.probengmech.2021.103189.

[23] F. Liu, J. Li, and L. Wang, "PI-LSTM: Physics-informed long short-term memory network for structural response modeling," *Eng Struct*, vol. 292, 2023, doi: 10.1016/j.engstruct.2023.116500.

[24] H. Li and C. Sun, "Nonlinear time-varying system response modeling via a real-time updated Runge-Kutta physics-informed neural network," *Eng Appl Artif Intell*, vol. 144, 2025, doi: 10.1016/j.engappai.2025.110067.

[25] J. Wang, J. Cao, S. Zhao, and Q. Qi, "S-wave velocity inversion and prediction using a deep hybrid neural network," *Sci China Earth Sci*, vol. 65, no. 4, 2022, doi: 10.1007/s11430-021-9870-8.

[26] J. E. Hurtado, J. M. Londoño, and M. A. Meza, "On the applicability of Neural Networks for soil dynamic amplification analysis," *Soil Dynamics and Earthquake Engineering*, vol. 21, no. 7, 2001, doi: 10.1016/S0267-7261(01)00037-9.

[27] H.-W. Li, J. Zhou, S. Hao, Y.-Q. Ni, and Z.-D. Xu, "Dynamic modeling and substructuring analysis leveraging long short-term memory neural network," *Structures*, vol. 70, 2024, doi: 10.1016/j.istruc.2024.107602.

[28] D. Van Nguyen, Y. Choo, and D. Kim, "Deep learning application for nonlinear seismic ground response prediction based on centrifuge test and numerical analysis," *Soil Dynamics and Earthquake Engineering*, vol. 182, 2024, doi: 10.1016/j.soildyn.2024.108733.

[29] P. Huang and Z. Chen, "Deep learning for nonlinear seismic responses prediction of subway station," *Eng Struct*, vol. 244, 2021, doi: 10.1016/j.engstruct.2021.112735.

[30] A. Djerrad, Y. Zhou, and S. Meng, "Efficient deep learning based models for rapid prediction of RC shear wall responses under monotonic and cyclic loading," *Journal of Building Engineering*, vol. 98, 2024, doi: 10.1016/j.jobe.2024.111131.

[31] W. Hao, H. Yin, J. Liu, R. Ma, D. Zhao, and S. Lu, "Seismic response prediction of soil tunnel structure based on CNN-LSTM model," *Structures*, vol. 73, 2025, doi: 10.1016/j.istruc.2025.108381.

[32] P. C. Kuo, Y. T. Chou, K. Y. Li, W. T. Chang, Y. N. Huang, and C. S. Chen, "GNN-LSTM-based fusion model for structural dynamic responses prediction," *Eng Struct*, vol. 306, 2024, doi: 10.1016/j.engstruct.2024.117733.

[33] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[34] A. Zar, S. Li, C. Li, L. Kun, and M. Akbar, "Deep learning-based ground motion inversion through recursive structural acceleration response using DRA-LSTM Net," *Eng Struct*, vol. 323, p. 119132, Jan. 2025, doi: 10.1016/J.ENGSTRUCT.2024.119132.

[35] J. Austin, "ADAM. An associative neural architecture for invariant pattern classification," in *IEE Conference Publication*, 1989, pp. 196–200.

[36] J. Shi and D. Asimaki, "A generic velocity profile for basin sediments in California conditioned on VS30," *Seismological Research Letters*, vol. 89, no. 4, pp. 1397–1409, 2018, doi: 10.1785/0220170268.

[37] D. Asimaki, J. Shi, W. Li, and P. Ayoubi, "Caltech-geoquake/PySeismoSoil." [Online]. Available: https://zenodo.org/records/13833723

[38] F. M. Salem, *Recurrent neural networks: From simple to gated architectures*. 2022. doi: 10.1007/978-3-030-89929-5.
[39] F. Lazzeri, *Machine learning for time series forecasting with python*. 2020. doi: 10.1002/9781119682394.