

Original Article

# Monte Carlo and Neural Network-based Prediction of Base Shear Force in Airport Tunnels using the Pushover Method Considering Both Soil-Structure Interaction and Non-Interaction across Different Soil Types

Awatif EL BRIGUI<sup>1</sup>, Mouna EL MKHALET<sup>2</sup>, Nouzha LAMDOUAR<sup>3</sup>

<sup>1,2,3</sup>Civil Engineering and Construction Laboratory, Mohammadia School of Engineers, Mohammed V University in Rabat.

<sup>1</sup>Corresponding Author : [adnanecelia391@gmail.com](mailto:adnanecelia391@gmail.com)

Received: 17 January 2026

Revised: 20 February 2026

Accepted: 05 March 2026

Published: 29 September 2026

**Abstract** - Artificial Neural Networks (ANNs) are powerful nonlinear function approximators that can learn complicated relationships from data. They offer increasing value in civil, seismic, and geotechnical engineering applications. Recent findings indicate that ANNs can also be applied to structural health monitoring, material behaviour prediction and the evaluation of deep buried structures, whereas their application to airport tunnels remains largely unexplored. This study expands the Monte Carlo framework developed for the seismic assessment of an airport tunnel. It integrates the principles of a neural network. 200 samples are generated for each scenario by incorporating uncertainties of peak ground acceleration and material strength into Monte Carlo simulations. The findings indicate a significant variation in the structure's roof displacement and base shear and larger dispersion of these parameters is observed as the ground acceleration increases. In the case without SSI, the roof displacement varies between 1.40 and 3.44 cm while the base shear varies from 690.0 to 1,845.1 kN; these are the minimum and maximum values obtained over the 200 Monte Carlo samples and correspond exactly to the figures reported in Table 4. The performance of tunnels is sensitive to the interdependence between seismic input and material. As a whole, the results obtained by combining ANN concepts, nonlinear static procedures, and probabilistic Monte Carlo simulation yield a sound framework for evaluation of performance under uncertainties. A combination of data analysis and modelling would enhance accuracy and strengthen decision-making for airport tunnel design and safety management.

**Keywords** - Artificial Neural Networks, Monte Carlo simulations, Nonlinear static analyses, Pushover methods, Seismic responses.

## 1. Introduction

### 1.1. Presentation

The use of Artificial Intelligence (AI) in civil and seismic engineering integrates data-driven learning with the physics of the field to identify patterns from images, vibration signatures, point cloud, and text that are too complex for manual analysis [1-3]. The conventional paradigms include supervised learning for prediction/classification, unsupervised learning for representation/discovery, and reinforcement learning for sequential decision making.

Deep neural networks that are flexible function approximators are a common choice in each of these paradigms [1, 4]. Recent applications increasingly adopt hybrid approaches which leverage physics-informed or physics-guided networks together with surrogate models. These approaches allow for better generalization, improved interpretability, and lower computational cost relative to full finite-element or nonlinear time-history simulations [5, 6].

AI plays an important role across the civil engineering lifecycle, from structural health monitoring, defect and crack detection, concrete mix and strength prediction, soil classification, construction progress monitoring, traffic forecasting, to BIM and point-cloud automation. Deployment is moving from lab scale to field scale, on-device, and real-time [3, 7-9].

The main shortcomings - data scarcity and imbalance, distribution shift, labelling noise, and limited explainability related to machine learning deployment - lead to the use of transfer learning, uncertainty quantification, and SHAP-style attribution to support engineering decisions and codes [1,5]. Artificial intelligence is also being used in seismic engineering for the computation of structural demands, generation of fragility functions, acceleration of pushover or dynamic analyses through surrogate models, and automation of post-event damage mapping from images or remote sensing [10-12].



The current research focuses on transfer learning with few labels, multi-task learning that involves simultaneous classification and regression, and hybrid pipelines combining simulated and observed data to achieve code-level reliability [13-15].

How can artificial intelligence techniques, including Monte Carlo simulation and neural networks, be systematically applied to predict the seismic behaviour of critical underground structures such as airport tunnels, while explicitly accounting for uncertainty in ground motion, material properties, and soil–structure interaction? This question is posed in the present study.

The main goal of the current research is to address this question. The objective is to determine whether a more trustworthy, interpretable, and computationally efficient framework can be developed for evaluating key seismic performance indicators—such as roof displacement and base shear at the base of the structure—by combining AI-based reasoning with probabilistic sampling approaches and nonlinear structural procedures (pushover analysis). This framework is developed for different conditions in which soil characteristics and uncertainty levels vary.

In doing so, the research effectively bridges data-driven intelligence and engineering analysis to support the design of safer and more resilient infrastructure. Artificial intelligence has been applied extensively to civil engineering structures since 2018 and, more recently, to underground works [1-15, 17]; to the authors' knowledge, no study has yet addressed airport tunnels specifically. Some works operating in this direction have been done for various applications such as traffic management, structure health monitoring, vulnerability assessment of structures during earthquake events, predictive

maintenance and intelligent safety control [1-15]. Various AI-based techniques have shown potential to enhance the performance of tunnels and tunnel infrastructures in normal [1, 5, 6] and extreme operating modes [9, 14, 15], while facilitating dynamic adaptability through real-time predictive framework applied to correct actions based on operational learning leveraging deep learning, reinforcement learning or combination of Algorithms (Hybrid), rendering those methodologies advantageous for optimal decision-making.

There are several combined architectural, geotechnical and functional elements that form an airport tunnel and characterize its operational performance as well as structural safety [18, 27, 28, 30]. These include:

- Tunnel lining: the concrete structural system around the inside of tunnels [27] that provides support and stability.
- Waterproofing membrane: the layer of sealing material which is set up between the tunnel lining and surrounding ground to avoid water invasion [18].
- Ventilation system: air circulation ducts and equipment for renewing air and removing harmful gases [18].
- Tunnel lighting refers to the lighting devices that provide visibility and operational safety within tunnel [18].
- Roadway: A pavement surface upon which a vehicle moves [18]
- Utility conduits : the ducts used for cables, pipelines and other service networks required by the operation of the tunnel [18].
- Vertical/horizontal soil and rock deposits, groundwater conditions: the geotechnical environment surrounding tunnels which have a great impact on the behavior of tunnel structure during different load phases but especially seismic loading [18, 27, 28, 30].

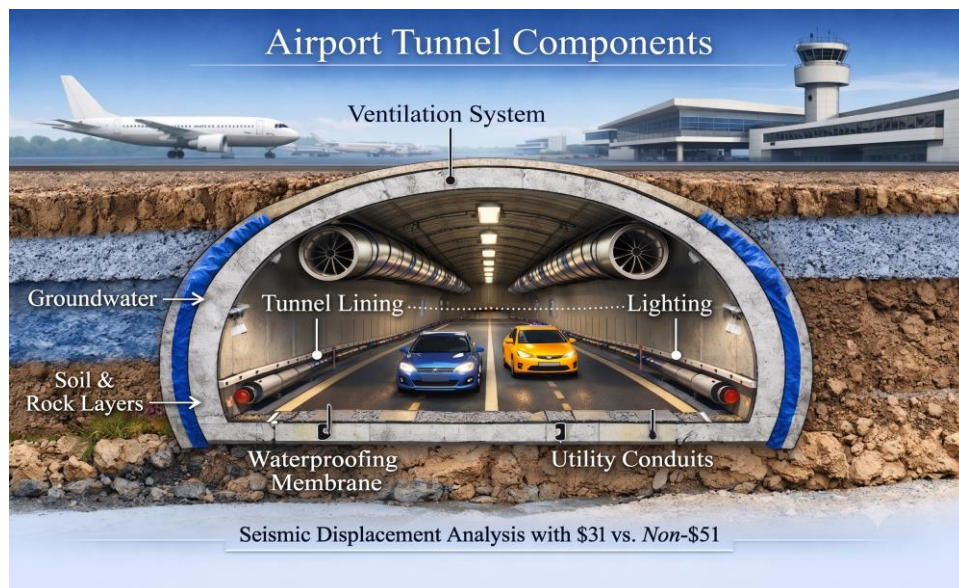


Fig. 1 Schematic illustration of airport tunnel components and seismic displacement analysis under SSI and non-SSI conditions

### 1.2. Research-Gap

While the field of Artificial Intelligence (AI) is progressing rapidly in seismic engineering, most AI-based studies have focused on above-ground structures, such as buildings and bridges. Compared to above-ground infrastructures, only a limited number of studies have been devoted to underground infrastructures, especially tunnels. Existing AI applications for tunnels mainly focus on predicting localized seismic responses, such as acceleration or displacement, by employing neural networks trained on nonlinear dynamic analyses [16, 17]. Typically, these studies rely on simplified modeling assumptions and do not explicitly address global structural demand parameters, such as base shear force, which are critical for performance-based seismic assessment.

Several studies have examined the seismic behaviour of tunnels while considering Soil-Structure Interaction (SSI); however, these studies usually investigate either SSI or fixed-base (non-SSI) conditions. There is little systematic or quantitative comparison of SSI versus non-SSI configurations within a common analytical or predictive framework [16, 18]. Consequently, the impact of SSI on the main seismic demand parameters for tunnels is not yet well quantified, particularly from a data-driven or surrogate-modeling perspective.

In addition, nonlinear pushover analysis has seen limited application to underground tunnel systems, despite its widespread use for above-ground structures in performance-based seismic assessment. Most existing tunnel studies rely on nonlinear time-history or pseudo-static methods, and the combination of pushover-based assessment with artificial intelligence has rarely been explored [16, 26]. This reveals a methodological gap for airport tunnels with respect to pushover-oriented frameworks.

Furthermore, many current AI-based tunnel seismic prediction models provide deterministic outputs, yielding single-value estimates of seismic response and failing to explicitly account for uncertainties arising from ground-motion variability, soil heterogeneity, or material properties [6, 17]. Probabilistic methods are increasingly recognized as essential for seismic risk assessment; however, uncertainty quantification remains limited in tunnel response prediction using neural networks.

Monte Carlo simulation, a commonly used probabilistic assessment tool in geotechnical and underground engineering, has not been widely combined with neural network models to predict tunnel response in terms of seismic base shear demand across different soil types. In damage-related studies, Monte Carlo methods have generally been used either for reliability assessment or damage probability evaluation, without incorporating AI-based surrogate models for seismic demand prediction [16, 18]. As a result, the current literature still lacks a unified probabilistic-intelligent framework that

simultaneously considers uncertainty propagation, nonlinear pushover analysis, soil-structure interaction effects, and neural network-based predictions for airport tunnels.

### 1.3. Novelty of this Research

The proposed study presents a pioneering and integrated framework that links probabilistic methods, artificial intelligence, and nonlinear structural analysis to predict the seismic base shear force of airport tunnels. As compared to earlier studies, the approach, which combines Monte Carlo simulation with neural networks, produces efficient probabilistic predictions that can model complex nonlinear interactions in tunnels. Besides, nonlinear pushover analysis is integrated as a major predictor in the method for predicting structural demand with performance-based characteristics rather than other surrogate models.

An additional innovation presented within this paper is the multi-soil evaluation, where effects due to varying ground conditions are evaluated within a single record's seismic response prediction, which adds to the generalisability and practical applicability of the prediction. In addition, the framework's technology enables the outputs of a neural network to be interpreted through probabilistic mapping, which can help support engineering decisions with confidence levels.

This study combines probabilistic uncertainty quantification, data-driven intelligence, and performance-based evaluation to create a holistic, scalable, and physically based tool (the parking lot model) for the seismic assessment of an airport tunnel. It offers a scientific improvement to existing methods. Furthermore, it can guide engineers in improving design and risk mitigation.

## 2. Methodology

In this research, the approach was following a hybrid method. It used an analytical and artificial-intelligence based method. The airport tunnel's input parameters were generated representative via the utilization of the Monte Carlo method.

Subsequently, to evaluate the displacement assessment of the tunnel pushover analysis and analytical study was used. The impact of soil-structure interaction was assessed in this research work. In particular, three major features of soil were considered that is the three major Moroccan soil types. In addition, one case study without considering the soil-structure interaction is also considered.

Four different cases were examined as such. In a later stage, artificial neural networks were developed and trained separately for each case to simulate the nonlinear behaviour of the tunnel and forecast its displacements. The shortcomings of the proposed methodology are discussed lastly & scientific contributions are showcased.

### 3. Results

#### 3.1. Pushover Method

##### 3.1.1. Principle of the Pushover Method

The pushover method is a nonlinear static procedure which is used to assess seismic response of a structure by applying lateral forces to a structure in a controlled manner. A capacity curve (base shear vs control-node displacement) is obtained from this monotonic loading sequence, and then judgment against seismic demand is made to infer likely performance. Currently, it has been made clear this workflow exists on performance-based assessment which specifies modeling requirement, target displacement, and acceptance criteria for components and systems. Specifically, it is a framework consisting of three tiers for evaluation and retrofitting: the code for the nonlinear static procedure, target-displacement rules, and performance objectives [19].

Recent papers underline refinements that address conventional single-pattern pushover limitations: inelastic redistribution and higher-mode participation. Feedback on the advancements from the 2023–2024 catalog include load patterns that cover multiple modes and spectrum-based analysis, upper-bound/modal combination rules, non-systematic-determinacy adaptive schemes that are defined to update the actions' distributions and properties as yielding happens in the structure, etc. The pushover predictions become more accurate by the use of these extensions especially so in the case of mid- to high-rise or torsionally irregular systems. This study is a thorough chronology and overview of current pushover strategies for structure analysis in seismic engineering [24].

The adaptive and multimode variants continue to evolve. Adaptive procedures update the lateral load vector at each step so as to track the redistribution of demand that accompanies stiffness degradation and the progressive formation of plastic mechanisms. Displacement-based adaptive pushover was formalised by Antoniou and Pinho (2004) [36], while Jalilkhani et al. (2020) proposed a multi-mode adaptive procedure that improves the estimation of seismic demands in reinforced concrete moment-resisting frames, particularly where higher-mode contributions are significant [25].

The method has been adapted for soil-structure interaction and section mechanisms characteristic of buried frames for underground/tunnel structures. An improved pushover formulation applied to shallow buried rectangular frames making use of distributed horizontal and vertical inertial forces to simulate soil confinement effects (2021) and a state-of-the-art review on tunnels (2022) place such pseudo-static/pushover approaches within broader fragility and performance assessment toolboxes. Artificial intelligence (AI) in civil and seismic engineering unites data-driven learning with domain physics to reveal patterns in images, vibration signals, point clouds, and text that are too complex for manual analysis [1-3].

AI and machine learning are popular tools in the civil engineering domain that brings benefits in various ways. The numerical simulation of construction material performance defines a frontier in which civil engineers will make further advances. Scientists model and characterize the performance of construction materials through numerical simulations on microstructural length scales, which are used to develop phenomenological theories of macroscopic behaviour. The core paradigms are supervised learning for prediction/classification, unsupervised learning for representation/discovery, as well as reinforcement learning for sequential decision-making, and with deep neural networks serving as flexible function approximators in these different modes [1, 4].

##### 3.1.2. Objectives of the Pushover Method

- Measure the limits of nonlinear capacity and deformation. Pushover reflects the degradation of strength and stiffness, material ductility and drift and rotation demands through the capacity curve and acceptable checks of performance-based standards (e.g. IO, LS, CP) [19].
- Expose possible damage pathways and points of weakness. Keeping tabs on the development of hinges and the buildup of demands along the load path is useful to identify components, mechanisms or interfaces likely to govern collapse prevention or serviceability. Recent reviews document how new loading patterns improve identification of mid/upper-story mechanisms. According to Kuria and Kegyes-Brassai (2024), pushover analysis in seismic engineering is an important technique for the assessment of structures in engineering [24].
- Assist in the evaluation and retrofit decision-making based on performance. The method can provide demand-to-capacity comparisons that fulfill [19] objectives and acceptance criteria for retrofits.
- Provide an economical way to perform Time History Analysis with limits. Pushover analysis is done to reduce computational effort. However, if a structure is significantly affected by higher-mode/torsional effects the designer should use care. Current remedies are the extended N2 procedure, the adaptive multimode procedure, and the bidirectional procedure [23, 37].
- Evaluate underground structures such as taxiways, runways, and tunnels. Tailored pushover pseudo-static formulations assist in preliminary performance evaluations and fragility assessments considering site-specific SSI conditions. The improved pushover method applied to seismic analysis and design of underground structures utilizes a translation method and displacement control method.

##### 3.1.3. Types of the Pushover Method (We focus on two procedures)

###### ATC-40 "Capacity Spectrum Method" (CSM) - Principle

The ATC-40 procedure (CSM) transforms the pushover curve into ADRS (spectral acceleration–spectral

displacement) to represent an equivalent SDOF system [20, 24, 33]. Seismic demand is taken from an elastic response spectrum that is iteratively modified for effective damping as the structure yields, and the performance point is the intersection of the damping-adjusted demand and the capacity in ADRS space [19, 20, 22].

Engineers value CSM for its transparency and visual insight into period lengthening and hysteretic damping, while recognizing that higher-mode or torsional effects may require enhanced load patterns or alternative procedures [24].

#### *FEMA/ASCE “Displacement (Coefficient) Method” - Principle*

The FEMA displacement (coefficient) method-introduced in FEMA-356, refined by FEMA-440, and now embedded in ASCE/SEI 41-23 - computes a target roof displacement by scaling elastic SDOF spectral displacement with empirically calibrated coefficients [19, 21, 22]. These modifiers account for inelastic action,  $P-\Delta$  effects, hysteretic/strength degradation, and the mismatch between the real MDOF system and its SDOF surrogate [19, 22].

The method yields a direct target displacement without constructing capacity–demand intersections, which many practitioners find convenient for performance checks and retrofit decisions within performance-based evaluation workflows [19, 24].

#### *3.1.4. Capacity Curve*

During nonlinear static (pushover) analysis, the capacity curve will be plotted with base shear against a control-node displacement as lateral loading is increased step-by-step. The stages are recorded and plotted to show how stiffness degrades and plastic mechanisms form [19, 35].

For engineering purposes, this raw curve is often idealized as a bilinear or tri-linear force-displacement relationship so as to consistently extract yield displacement/strength, post yield slope, capping/ultimate points, ductility and overstrength parameters. According to ASCE, this may cost \$25 million.

The capacity curve can be mapped to ADRS coordinates to form capacity spectrum and intersect (a damping-adjusted) demand spectrum at performance point. In spectrum-based procedures such as CSM/ATC-40 it supports performance-based checks, without running time histories [24, 33].

In addition to building-based evaluations, parameterized capacity curves also serve as the basis for fragility modeling and stock-level screening [38]. Modified pushover setups, which incorporate soil-structure interaction and the effects of vertical soil inertia, are able to replicate observed damage and yield informative performance curves for underground systems such as stations and tunnels [30].

#### *3.1.5. Advantages and Limitations - Pushover (ATC-40 vs. FEMA/ASCE)*

**Table 1. ATC-40 / Capacity Spectrum Method (CSM)**

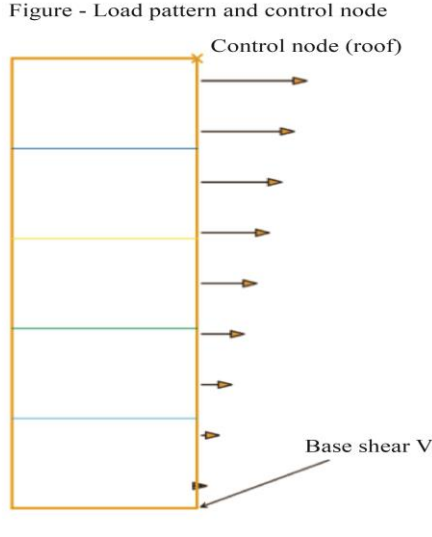
| <b>Advantages</b>                                                                                                               | <b>Limitations</b>                                                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Intuitive capacity–demand intersection in ADRS space clarifies the performance point and limit states for practitioners [24].   | Accuracy depends on assumptions for equivalent damping; conventional rules can bias demand, motivating improved variants [24].                                      |
| Captures increasing effective damping with ductility, offering a transparent way to reflect hysteretic energy dissipation [24]. | Requires an iterative solution and is sensitive to modeling/iteration choices for convergence [24].                                                                 |
| Computationally light and widely used for screening/retrofit studies and comparisons with other NSPs [24, 35].                  | Single/invariant load patterns can miss higher-mode and torsional effects in irregular or tall systems unless multimode/adaptive variants are adopted [24, 31, 34]. |

**Table 2. FEMA / ASCE displacement (Coefficient) method**

| <b>Advantages</b>                                                                                                                         | <b>Limitations</b>                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Directly computes target roof displacement via calibrated coefficients, avoiding ADRS intersections and streamlining practice [19, 32]    | Accuracy hinges on effective period, damping, and coefficient calibration; bias can occur for certain systems [24, 31].                                  |
| Fully codified in ASCE/SEI 41-23 with clear nonlinear-static procedure steps and acceptance criteria in performance-based workflows [19]. | Like other conventional NSPs, it struggles with strong higher-mode/torsional and bidirectional effects unless extended procedures are used [24, 31].     |
| Efficient for preliminary evaluation and retrofit scoping where full nonlinear time-history analysis is unwarranted [19, 24].             | Results may diverge from nonlinear response-history analyses for specific configurations, so dynamic verification is prudent in critical cases [24, 31]. |

### 3.1.6. Steps of the Pushover Method (General)

A pushover analysis is a nonlinear static procedure in which a lateral load pattern is increased step-by-step on a nonlinear structural model while tracking base shear  $V$  versus control-node displacement  $d$ ; the resulting capacity curve exposes stiffness degradation, hinge formation, and likely failure mechanisms [19, 24].



**Fig. 2 Lateral load pattern and control node for pushover analysis**

Source: Adapted from the nonlinear static procedures and load-pattern guidance in ASCE/SEI 41-23, with overview by Kuria & Kegyes-Brassai (2024)

Workflow (condensed). Define performance objectives (e.g., IO/LS/CP) and select an NSP variant; build the nonlinear model with calibrated component force–deformation laws and acceptance criteria; choose a control node and one or more lateral load patterns (mass-proportional, modal, or adaptive); push the structure to increasing displacement or loss of stability; idealize the capacity curve (bi/tri-linear) to extract yield/ultimate parameters; evaluate demand (ADRS intersection or coefficient method); check acceptance and iterate as needed [19, 24, 25].

Incremental equilibrium (one load step  $k$ ).

$$(u^{(k)})\Delta u^{(k)} = \Delta\lambda^{(k)}p, u^{(k+1)} = u^{(k)} + \Delta u^{(k)} \quad (1)$$

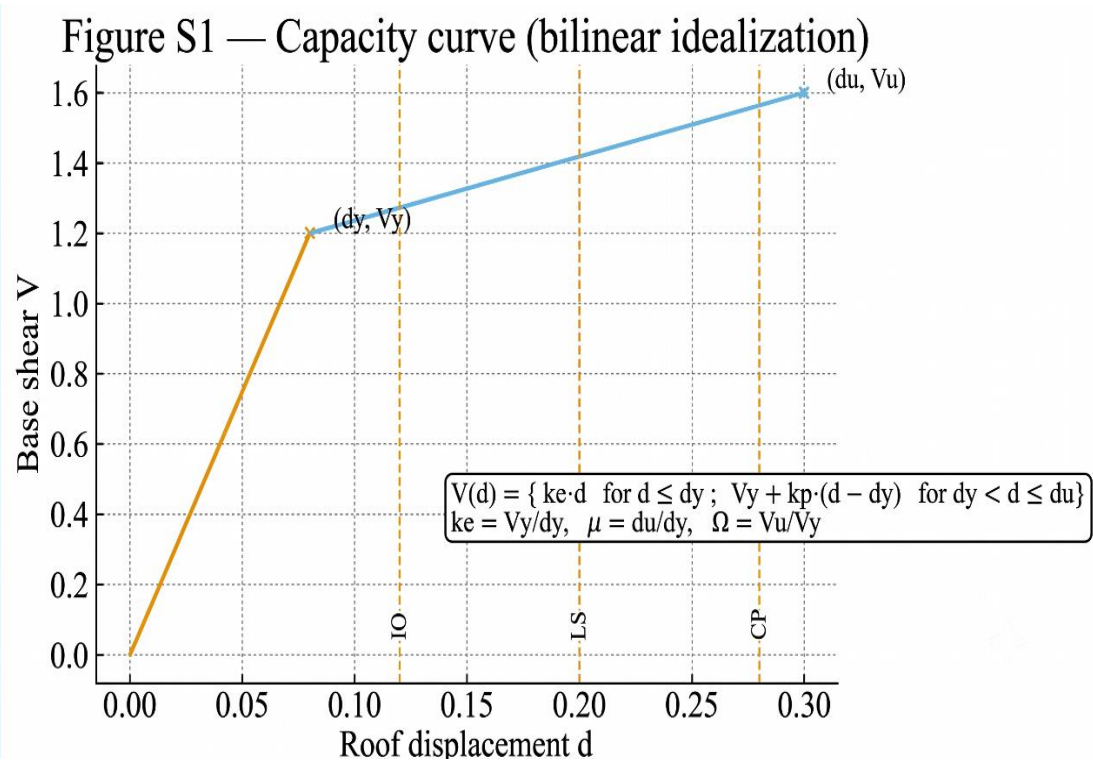
With  $p$  the lateral pattern,  $\Delta\lambda$  the load factor, and  $K_T$  the tangent stiffness [19].

Capacity-curve idealization (bilinear) and key indices.

$$V(d) = \begin{cases} k_e d, & d \leq d_y \\ V_y + k_p(d - d_y) & d_y < d \leq d_u \end{cases} \quad (2)$$

$$k_e = \frac{V_y}{d_y}, \quad \mu = \frac{d_u}{d_y}, \quad \Omega = \frac{V_u}{V_y}$$

(ASCE/SEI 41-23, 2023; Fazarinc et al., 2024).



**Fig. 3 Capacity curve**

Source S1: Concept and indices ( $V_y$ ,  $d_y$ ,  $\mu$ ,  $\Omega$ ) per ASCE/SEI 41-23; use of bi/tri-linear idealizations reviewed by Kuria and Kegyes-Brassai (2024) [24]

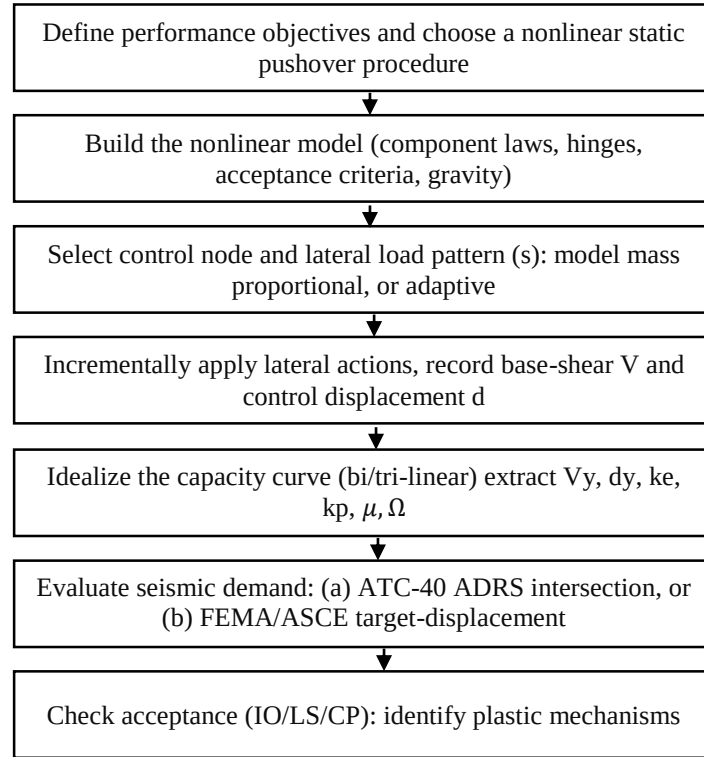


Fig. 4 Flowchart - Pushover method (General steps)

### 3.1.7. Steps of the ATC-40 Capacity Spectrum Method (U.S.) Idea

Convert the MDOF capacity to an equivalent SDOF capacity spectrum in ADRS coordinates and locate the

performance point where capacity meets a damping-adjusted demand spectrum [19, 24].

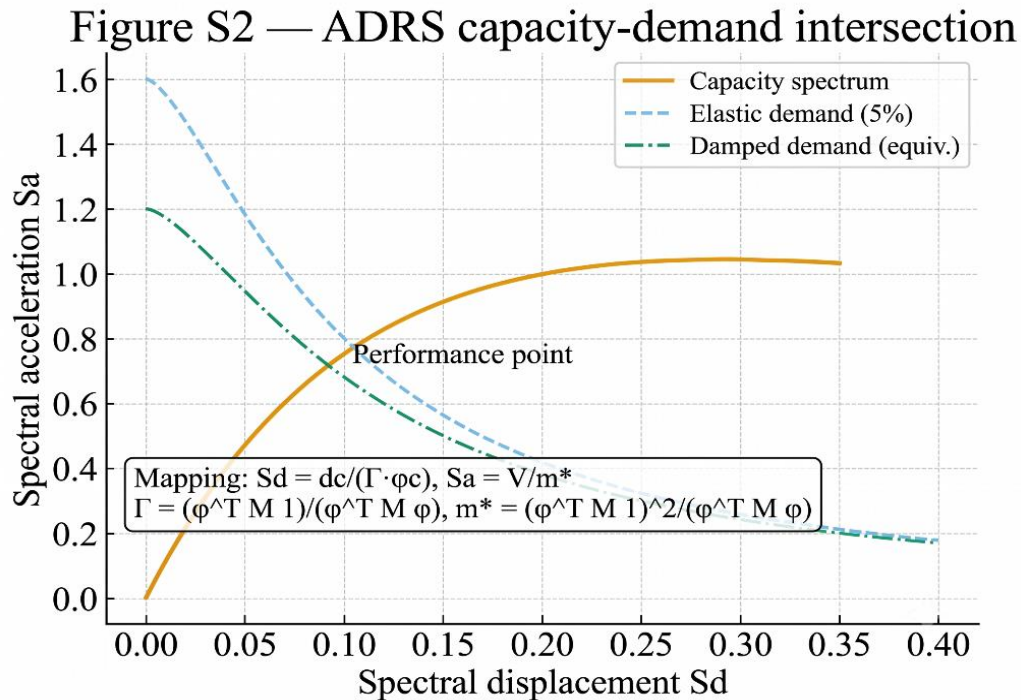


Fig. 5 ADRS capacity

Source S2: Capacity Spectrum Method per ATC-40 and ASCE/SEI 41-23; contemporary summary in Kuria & Kegyes-Brassai (2024)

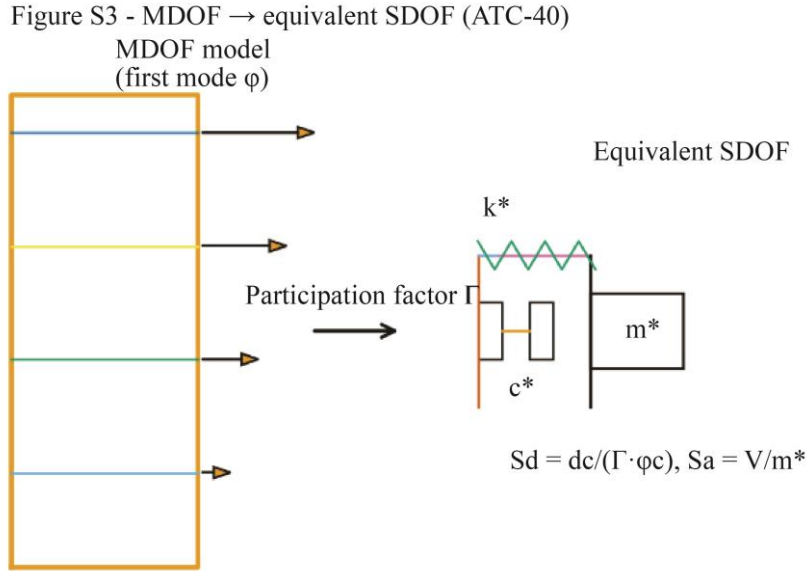


Fig. 6 MDOF

Source S3 : “Modal participation factor  $\Gamma$  and effective mass  $m$  per ATC-40/Capacity Spectrum framework and ASCE/SEI 41-23.”

#### Step-by-Step (with Core Relations)

1. Nonlinear pushover → obtain V–d for the chosen load pattern [24].
2. MDOF→SDOF transformation (first mode  $\phi$ , participation  $\Gamma$ , effective mass  $m^*$ ):

$$\Gamma = \frac{\phi^T M_1}{\phi^T M_0}, \quad m^* = \frac{(\phi^T M_1)^2}{\phi^T M_0}$$

3. ADRS mapping using control node c:

$$S_d = \frac{d_c}{\Gamma \phi_c}, \quad S_a = \frac{V}{m^*}$$

4. Demand spectra in ADRS; apply equivalent-viscous damping rules (ductility-dependent) to reduce the elastic demand.
5. Iterate to the performance point (capacity–demand intersection); back-compute story drifts, hinge rotations, etc., and check acceptance [19].
6. Higher-mode/torsion → use multimode/adaptive variants or verify with nonlinear response-history analysis [25].

#### 3.1.8. Steps of the FEMA/ASCE Coefficient (Target Displacement) Method

Concept Determine the target roof displacement directly from the elastic spectral displacement multiplied by code factors that account for the effects of inelasticity, P- $\Delta$ , hysteresis, and MDOF→SDOF effects [19].

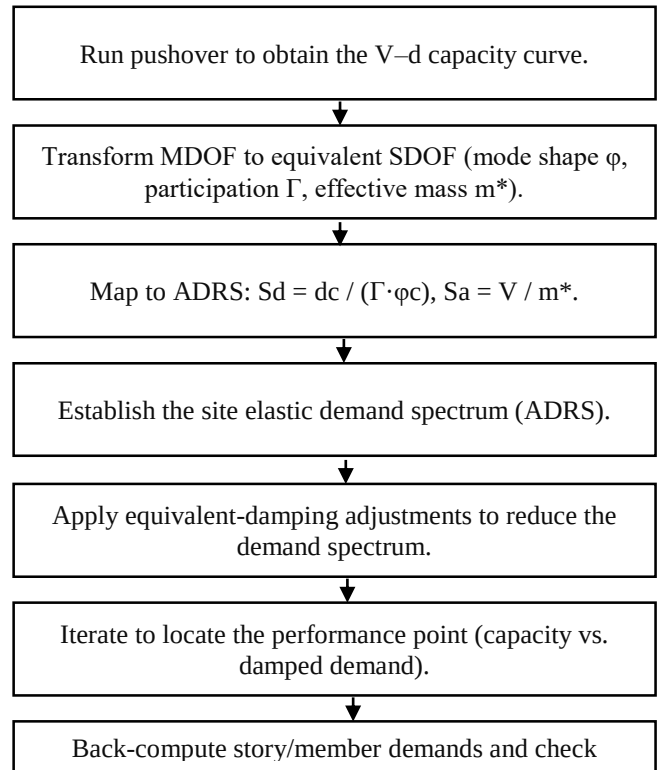


Fig. 7 Flowchart - ATC-40 capacity spectrum method (ADRS)

#### Step-by-step (with core relations)

1. Run pushover and idealize the capacity curve to identify  $d_y$ ,  $V_y$ ,  $k_e$ ,  $k_p$  and ultimate/capping points [19].
2. Compute effective properties ( $T_{eff}$ ,  $\xi_{eff}$ ) from the idealized curve [19].

3. Elastic spectral displacement (reference damping):

$$S_{d,el} = S_a(T_{eff}, \xi_{eff}) \left( \frac{T_{eff}}{2\pi} \right)^2 g, \quad (3)$$

Next, use the code's damping adjustment to determine  $S_d$  [19].

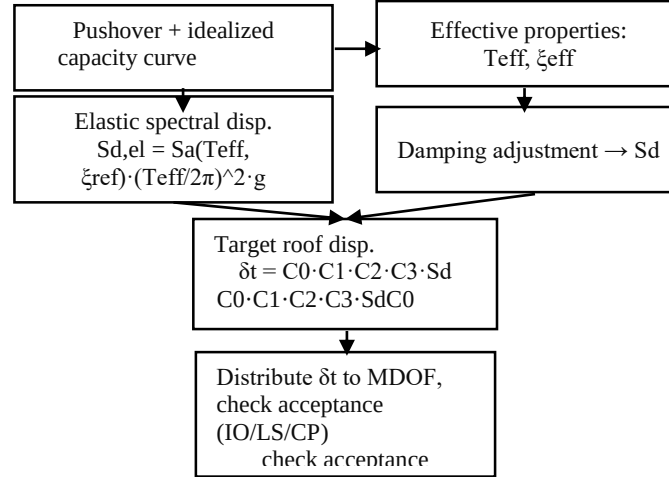
4. Target displacement (code coefficients  $C_i$ ):

$$\delta_t = C_0 C_1 C_2 C_3 C_d \quad (4)$$

Where  $C_i$  account for MDOF/SDOF mapping, inelastic amplification, and stability/hysteretic effects [19].

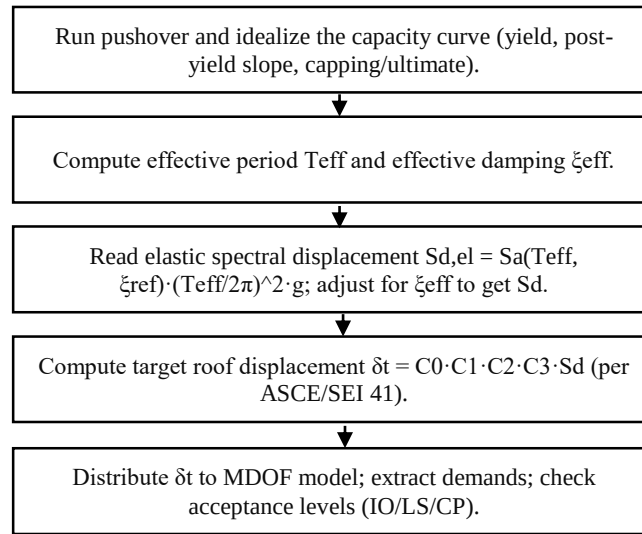
5. Distribute  $\delta_t$  to the MDOF model, extract demands, and check acceptance for the intended performance level [19].

6. Irregular or higher-mode-sensitive systems → corroborate with multimode/adaptive pushover or time-history analysis [24, 25].



**Fig. 8 FEMA/ASCE coefficient method (target-displacement)**

(Source S4: "Target-displacement formula  $\delta_t = C_0 \cdot C_1 \cdot C_2 \cdot C_3 \cdot S_d$  and damping adjustments per ASCE/SEI 41-23.")



**Fig. 9 Flowchart - FEMA/ASCE coefficient (Target-Displacement) method**

### 3.2. The Monte Carlo Method

#### 3.2.1. Principle of the Monte Carlo method

The Monte-Carlo method is one of the sampling-based numerical methods to compute failure probability, expected value or integral (in other words generally a quantity) associated with a model by repeatedly evaluating the model

using inputs drawn at random from specified probability distributions. In reality, the procedure consists of: (i) defining the uncertain input variables in consideration and assign Probability Density Functions (PDFs), (ii) generating a large number of random samples of these inputs, (iii) running the deterministic model for each sample, and (iv) calculating

statistics (mean response, variance, probability of failure) of the ensemble of outputs. The Monte Carlo simulation is accepted as a "gold standard" reference method in structural and geotechnical reliability analysis because it works directly in the original random space and does not rely on linearization or small-variance assumptions.

### 3.2.2. Advantages and Disadvantages of Monte Carlo

Main Advantages:

- The algorithm is easy to implement and applicable to almost any model linear or nonlinear, static or dynamic.
- Even when the performance function is non-linear, discontinuous or implicit (finite-element models for example) validity is retained.
- The method can handle multiple random variables without requiring major changes to the algorithm. Some analytical reliability methods become inapplicable when dimensionality increases.
- Simulations in parallelization take place straight away. Runs being independent Monte Carlo lends itself well to supercomputer and cloud computing.

Main Disadvantages :

- Many thousands or millions of model evaluations may be needed for accurate estimates, especially when the failure probability is very small.
- As the number of samples increases, the sampling error decreases slowly, precisely in accordance with its square root. Thus, if we want to halve the error, we need to perform four times as many simulations.
- When each deterministic analysis is costly (e.g. 3-D nonlinear FEM), direct Monte Carlo may be impractical unless surrogates or variance-reduction strategies are introduced.
- For most practical applications (31 words): Assumes known probability distributions of input variables, or at least reasonable approximations; poor assumptions at this stage can dominate the results of the method.

### 3.2.3. Probability Density Function

A Probability Density Function (PDF) describes how the probability of a continuous random variable is distributed over its possible values. For a scalar random variable  $X$  with PDF  $f_X(x)$ :

- The probability that  $X$  lies between two values  $a$  and  $b$  is

$$P(a \leq X \leq b) = \int_a^b f_X(x) dx \quad (5)$$

- The PDF is always non-negative,  $f_X(x) > 0$ , and its integral over the entire real line equals 1:

$$\int_{-\infty}^{\infty} f_X(x) dx = 1 \quad (6)$$

- The cumulative distribution function (CDF) is obtained by integrating the PDF from  $-\infty$  to  $x$ :

$$F_X(x) = P(X \leq x) = \int_{-\infty}^x f_X(t) dt \quad (7)$$

In engineering reliability, PDFs (e.g. normal, lognormal, gamma) are assigned to material strengths, loads, geometric parameters or soil properties; Monte Carlo simulation then draws random samples from these PDFs to propagate uncertainty through the structural or geotechnical model.

### 3.3. Neural Network

Artificial Neural Networks (ANNs) are inspired by the biological neural architecture and consist of nodes which are connected as layers. The weightage of each connection between these nodes can be adjusted. Each artificial neuron computes the weighted sum of its input after applying a non-linear activation function.

The weights of a neural network are trained using an optimization algorithm, for example gradient backpropagation that optimally changes these weights during learning so that cost function which measures how far the predicted values are from actual target values is minimized.

Through this iterative procedure, the network acquires progressively abstract and hierarchical representations of the data, thus capturing complex patterns and modeling non-linear relationships. Neural networks' most powerful feature is this ability.

Neural networks are a computational model made up of highly interconnected processing units called artificial neurons organized in a graph. Neural networks vary in type based on the following four features.

1. Architecture: The organization of the graph (e.g., layered or fully connected).
2. Complexity: Determined by the number of neurons and the presence or absence of cycles (feedback).
3. Neuron Type: Defined by its mathematical computation function, particularly its activation function.
4. Purpose: The intended application, such as supervised learning, optimization, or the modeling of dynamic systems.

#### 3.3.1. Types of Neural Networks Multilayer Perceptron (MLP) / Feedforward Neural Networks (FNN)

Description: The simplest kind of neural network. The network consists of an input layer, one or more hidden layers, and one output layer. Data is flowing in one direction, from input to output and there are no cycles! These are great for learning non-linear mappings between two datasets. Common Applications: Classification of images, regression modelling, feature learning.

### Convolutional Neural Networks (CNN or ConvNet)

**Description:** Generally refers to data that can be structured in a grid. The convolutional layers of filters utilize the dot product operation with a sliding window that extracts local spatial features (edges, textures, shapes). They often use pooling layers to lower dimensions.

**Common Applications:** Computer vision, image and video recognition, medical image processing, autonomous vehicles.

### Recurrent Neural Networks (RNN)

**Description:** Parsed sequential data (time series, text, audio). A key characteristic of theirs is cyclic connections that form a “memory” to maintain information from previous time marks. They analyze series one piece at a time.

**Common Applications:** Machine translation, language modeling, sentiment analysis, time series prediction.

### Long Short-Term Memory (LSTM) & Gated Recurrent Units (GRU)

**Description:** Advanced RNN architectures created to solve the “vanishing gradient” problem of simple RNNs, which prevented them from learning long-range dependencies. They use sophisticated “gating” mechanisms to regulate the flow of information, allowing them to remember or forget information over long periods selectively.

**Common Applications:** All RNN applications, particularly those where long-term context is crucial (e.g., translating long sentences, speech synthesis).

### Auto Encoders (AE)

**Description:** Unsupervised networks designed to learn a compressed, efficient representation (encoding) of data. They consist of two parts: an encoder that compresses the input into a lower-dimensional latent representation, and a decoder that attempts to reconstruct the original input from this representation.

**Common Applications:** Dimensionality reduction, data denoising, generation of faces, anomaly detection.

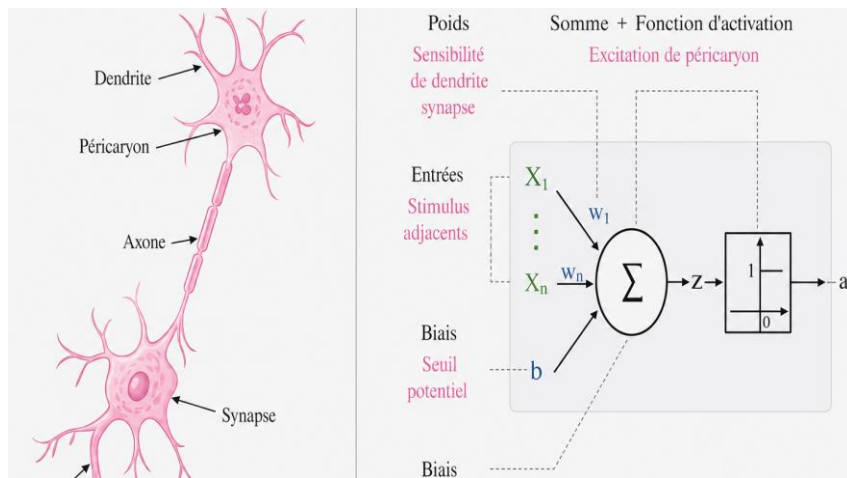
### Transformers

**Description:** A revolutionary architecture based on an attention mechanism (often “self-attention”) that allows the model to weigh the importance of all parts of an input sequence, regardless of their distance, when making a prediction. Unlike RNNs, they process the entire sequence in parallel, dramatically improving computational efficiency.

**Common Applications:** Large language models (GPT, BERT), state-of-the-art machine translation, text generation (e.g., ChatGPT).

### 3.3.2 Structure and Components of Neural Networks Mathematical imitation model, perceptron

The perceptron is a computational model designed to mimic the mathematical function of a biological neuron. It accepts many binary input signals,  $x_1, x_2, \dots, x_n$ , similar to the dendritic inputs. Every input is scaled by a synaptic weight,  $w_1, w_2, \dots, w_n$  which indicates the strength or sensitivity of the synaptic connection.



**Fig. 10 Perceptron schematic**

Source : Goodfellow, Bengio & Courville, *Deep Learning*, Ch. 6 “Deep Feedforward Networks” (free online textbook; formal perceptron/MLP setup and notation)

The fundamental operation is to obtain the weighted sum of these input and compare it with the threshold value  $-b$  (commonly called threshold potential). If this weighted sum is exceeded with respect to this threshold, the neuron fire or

activates and in this activity produces 1 output. If not; then beyond this threshold, it does not fire or activates and this activity makes 0 outbound.

$$a = 0 \text{ si } \sum_{i=1}^n \binom{n}{i} w^i x^i + b \leq 0 \quad (8)$$

$$a = 1 \text{ si } \sum_{i=1}^n \binom{n}{i} w^i x^i + b > 0 \quad (9)$$

#### Nonlinear Characteristic, Activation Function

However, the step function presents practical limitations for algorithmic use in developing perceptrons and neural networks. As a result, numerous alternative activation functions have been developed that demonstrate superior performance.

The sigmoid function, a commonly employed S-shaped curve, maps input values to the interval (0,1) according to the following definition:

$$S(x) = \frac{1}{1+e^{-x}} \quad (10)$$

And its derivative can be easily calculated:

$$S'(x) = \frac{e^{-x}}{(1+e^{-x})^2} S(x)(1 - (x)) \quad (11)$$

It can be observed that when the input to the sigmoid function reaches extremely high or low values, its output enters a saturation region where the gradient approaches zero—a phenomenon commonly referred to as the "vanishing gradient" problem. Conversely, when the input is near zero, the derivative of the sigmoid function attains its maximum value.

Moreover, the sigmoid function produces a half-

saturation state (output = 0.5) precisely when the activation threshold is met (input = 0). This behavior diverges from that of biological neural networks, which typically exhibit low proportions of simultaneously active neurons (approximately 1% to 4%), thereby avoiding widespread saturation.

For these reasons, the Rectified Linear Unit (ReLU) activation function demonstrates stronger alignment with neurophysiological models. In addition to mitigating issues related to vanishing and exploding gradients, ReLU also simplifies computational processes and accelerates network convergence, owing to its constant derivative of 1 for all positive inputs.

$$R(x) = \max(0, x) \quad (12)$$

The nonlinear activation functions described above enable the incorporation of nonlinear modeling capabilities into artificial neural networks. Under certain conditions, these functions allow the model to approximate any continuous nonlinear mapping between inputs and outputs to a specified degree of accuracy, provided the network contains a sufficient number of neurons and hidden layers.

#### Construction of a Neural Network Model: Multilayer Perceptron

The Multilayer Perceptron (MLP) is a fully connected feedforward artificial neural network architecture that establishes mappings between input and output datasets. For instance, input data may consist of extracted features from images or documents, while the model outputs perform specific predictive tasks such as object recognition or classification.

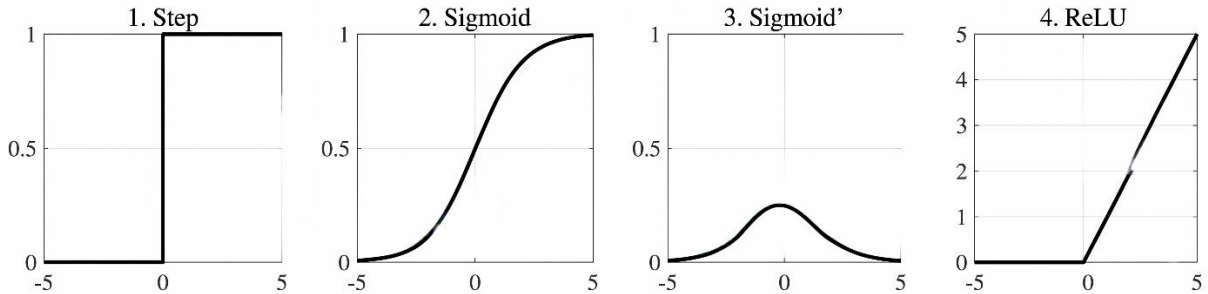


Fig. 11 Activation functions overview

This formulation maintains the original meaning while using more natural academic phrasing and improves flow with appropriate connective words. The model architecture has three main components such as an input layer (containing the input variables), one or more hidden layers (which are composed of intermediate computational nodes), and an output layer (which produces the output variables). This arrangement is fully connected such that every neuron of a given layer connects to every neuron of the previous layer. We have seen how a single perceptron receives a signal, processes

it, and sends the signal. We will now look at how signals are computed sequentially layer-wise in a network. Forward propagation is the activity that allows signals to flow from inputs to the output in a fully connected model.

$$a_1 = \sigma(z_1) = \sigma(x_1 w_1 + x_2 w_3 + b_1)$$

$$a_1 = \sigma(z_2) = \sigma(x_1 w_1 + x_2 w_3 + b_1)$$

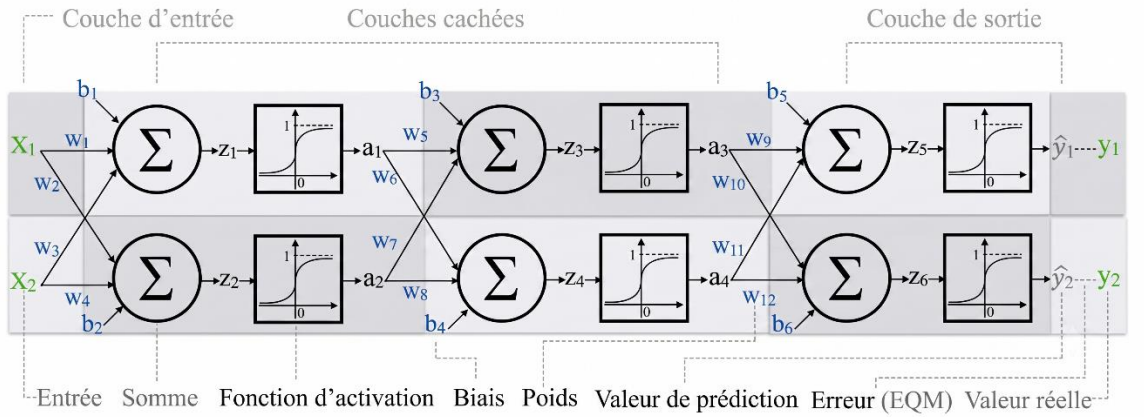
$$a_1 = \sigma(z_3) = \sigma(x_1 w_1 + x_2 w_3 + b_1)$$

:

$$a_1 = \sigma(z_n) = \sigma(x_1 w_1 + x_2 w_3 + b_1)$$

Source: Goodfellow, Bengio & Courville, *Deep Learning*, Ch. 6 — formal forward propagation  $z = Wx + bz = Wx + bz = Wx + b$ , activations (sigmoid and derivative  $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ ), standard losses (e.g., MSE), and full backprop through the chain rule

Where  $a_n$  is the output of the current neuron,  $\sigma$  is the activation function,  $a$  and  $W$  are the matrix forms of the outputs and their corresponding weights from the previous layer, and  $b_n$  is the bias term.



**Fig. 12 Multilayer Perceptron (MLP) architecture (input/hidden/output layers; fully-connected)**

Source: Dive into Deep Learning (D2L.ai), “Multilayer Perceptrons” — accessible MLP diagrams and explanations (ADRS-free, open web).

#### Evaluation of Result Satisfaction, Loss Function

During forward propagation, each layer takes inputs from the previous layer and processes them, sending outputs to the next layer. The prediction  $\hat{y}$  is produced by the final output layer. A loss function measures the quality of the outputs of the neural network to assess prediction accuracy. The mean squared error (MSE) is one of the most commonly used loss functions for this task.

$$L = \frac{1}{2n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (13)$$

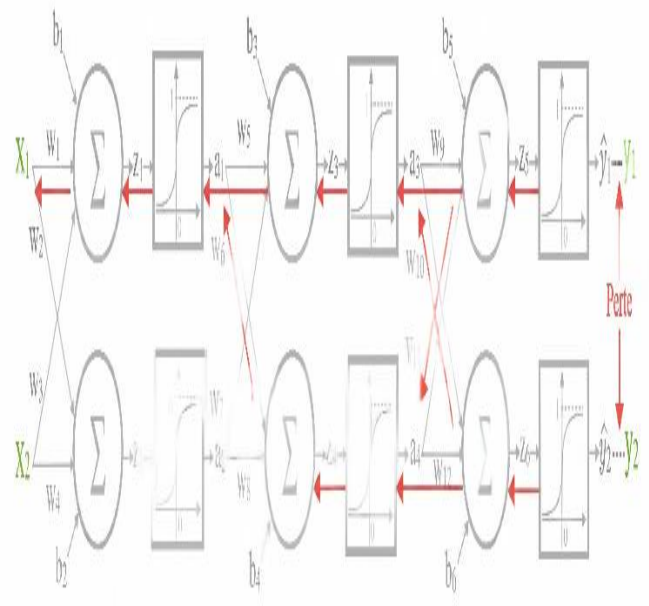
Loss value is the difference between the neural network prediction  $\hat{y}_i$  and actual outcome  $y_i$ . A more significant loss value reflects a more significant deviation of the prediction from the expected outcome. The “learning” process of the model consists of calibrating parameters ( $\theta = \{w_1, w_2, \dots, w_j, b_1, b_2, \dots, b_k\}$ ) to make correct predictions with maximum precision. Model learning aims to minimize the loss function by tuning the internal parameters of the model. In other words, it tries to bring the predicted values as close to the true values as possible.

$$\underset{\theta}{\operatorname{argmin}} L(\theta)$$

#### Layer-by-Layer Gradient Calculation, Backpropagation (BP)

To identify the parameters that minimize the loss value, it is necessary to compute the gradient at each iteration—a challenging task due to the extensive number of internal parameters ( $w_j$  and  $b_k$ ) in the neural network model. However, these parameters are often interrelated across successive

layers. The backpropagation method, introduced by Rumelhart et al. in 1986, enables efficient layer-wise gradient computation starting from the output layer. By distributing the error contribution across connections, backpropagation reduces computational redundancy and significantly enhances gradient search efficiency. The following section details the weight optimization algorithm for loss minimization.



**Fig. 13 Forward Propagation in a Multi-Layer Neural Network**

The backpropagation method primarily relies on the chain rule of differentiation. For instance, given differentiable functions  $h$ ,  $g$ ,  $k$  and variables  $z$ ,  $y$ ,  $x$ ,  $s$ :

**Case 1:**

If  $z = h(y)$  and  $y = g(x)$  then:

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx} \quad (14)$$

**Case 2:**

If  $z = k(x, y)$ ,  $x = g(s)$ , and  $y = h(s)$ , then:

$$\frac{dz}{ds} = \frac{\partial z}{\partial x} \cdot \frac{dx}{ds} + \frac{\partial z}{\partial y} \cdot \frac{dy}{ds} \quad (15)$$

The gradients of the neural network parameters (e.g.  $\frac{\partial L}{\partial w_9}$ ,  $\frac{\partial L}{\partial w_5}$ , and  $\frac{\partial L}{\partial w_1}$ ) can be computed using the given values ( $x_1, x_2, y_1, y_2$ ) and the values calculated during forward propagation ( $z_1, \dots, z_6, a_1, \dots, a_4, \hat{y}_1, \hat{y}_2$ ):

- Calculation of  $\partial L / \partial w_9$  (similar process applies to  $\partial L / \partial w_{10}$ ,  $\partial L / \partial w_{11}$ ,  $\partial L / \partial w_{12}$ ,  $\partial L / \partial b_5$ ,  $\partial L / \partial b_6$ ):

Using the chain rule (Case 1):

$$\frac{\partial L}{\partial w_9} = \frac{\partial L}{\partial \hat{y}_1} \cdot \frac{\partial \hat{y}_1}{\partial z_5} \cdot \frac{\partial z_5}{\partial w_9} \quad (16)$$

Step-by-step derivation:

1. Compute  $\frac{\partial L}{\partial \hat{y}_1}$ :

$$\frac{\partial L}{\partial \hat{y}_1} = \hat{y}_1 - y_1 \quad (\text{since } L = \frac{1}{2n} \sum_{i=1}^n (\hat{y}_i - y_i)^2) \quad (17)$$

2. Compute  $\frac{\partial \hat{y}_1}{\partial z_5}$ :

$$\frac{\partial \hat{y}_1}{\partial z_5} = \hat{y}_1 (1 - \hat{y}_1) \quad (18)$$

(Because  $\hat{y}_1 = S(z_5) = \frac{1}{1+e^{-z_5}}$  and  $S'(z_5) = \frac{e^{-z_5}}{(1+e^{-z_5})^2} = S(z_5)(1-S(z_5))$ )

3. Compute  $\frac{\partial z_5}{\partial w_9}$ :

$$\frac{\partial z_5}{\partial w_9} = a_3 \quad (19)$$

(Because  $z_5 = a_3 \cdot w_9 + a_4 \cdot w_{11} + b_5$ )

Calculation of  $\partial L / \partial w_5$  (Similar process for  $\partial L / \partial w_6$ ,  $\partial L / \partial w_7$ ,  $\partial L / \partial w_8$ ,  $\partial L / \partial b_3$ ,  $\partial L / \partial b_4$ ):

Using the chain rule (combining Case 1 and Case 2):

$$\frac{\partial L}{\partial w_5} = \frac{\partial L}{\partial \hat{y}_1} \cdot \frac{\partial \hat{y}_1}{\partial z_5} \cdot \frac{\partial z_5}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial w_5} + \frac{\partial L}{\partial \hat{y}_2} \cdot \frac{\partial \hat{y}_2}{\partial z_6} \cdot \frac{\partial z_6}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial w_5} \quad (20)$$

$$= \left( \frac{\partial L}{\partial \hat{y}_1} \cdot \frac{\partial \hat{y}_1}{\partial z_5} \cdot \frac{\partial z_5}{\partial a_3} + \frac{\partial L}{\partial \hat{y}_2} \cdot \frac{\partial \hat{y}_2}{\partial z_6} \cdot \frac{\partial z_6}{\partial a_3} \right) \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial w_5}$$

$$= \left( \frac{\partial L}{\partial z_5} \cdot \frac{\partial z_5}{\partial a_3} + \frac{\partial L}{\partial z_6} \cdot \frac{\partial z_6}{\partial a_3} \right) \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial w_5}$$

1.  $\frac{\partial L}{\partial z_5} = \frac{\partial L}{\partial \hat{y}_1} \cdot \frac{\partial \hat{y}_1}{\partial z_5}$  and  $\frac{\partial L}{\partial z_6} = \frac{\partial L}{\partial \hat{y}_2} \cdot \frac{\partial \hat{y}_2}{\partial z_6}$  are already calculated above
2.  $\frac{\partial z_5}{\partial a_3} = w_9$  and  $\frac{\partial z_6}{\partial a_3} = w_{10}$
3.  $\frac{\partial a_3}{\partial z_3} = a_3(1 - a_3)$
4.  $\frac{\partial z_3}{\partial w_5} = a_1$
- $\partial L / \partial w_1$  (Similar process for  $\partial L / \partial w_2$ ,  $\partial L / \partial w_3$ ,  $\partial L / \partial w_4$ ,  $\partial L / \partial b_1$ ,  $\partial L / \partial b_2$ ):

Similarly, according to the chain rule and the gradients calculated above, we have:

$$\frac{\partial L}{\partial w_1} = \left( \frac{\partial L}{\partial z_3} \cdot \frac{\partial z_3}{\partial a_1} + \frac{\partial L}{\partial z_4} \cdot \frac{\partial z_4}{\partial a_1} \right) \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1}$$

$$= \left( \frac{\partial L}{\partial z_3} w_5 + \frac{\partial L}{\partial z_4} w_6 \right) a_1 x_1 \quad (21)$$

The calculation above is for only a single data point. The gradients of the loss function parameters for the entire dataset are the sum of the results from each individual data point:

$$\frac{\partial L(\theta)}{\partial w_j} = \frac{\partial \sum_{n=1}^N L^n(\theta)}{\partial w_j} = \sum_{n=1}^N \frac{\partial L^n(\theta)}{\partial w_j} \quad (22)$$

Where  $n$  is the index of each data point and  $N$  represents the total volume (or size) of the dataset. The same gradient calculation principle is used for neural networks with a large number of layers and neurons:

$$\nabla L(\theta) = \begin{bmatrix} \frac{\partial L(\theta)}{\partial w_1} \\ \dots \\ \frac{\partial L(\theta)}{\partial w_j} \\ \dots \\ \frac{\partial L(\theta)}{\partial b_1} \\ \dots \\ \frac{\partial L(\theta)}{\partial b_k} \end{bmatrix} \quad (23)$$

### 3.4. Case Study - Airport Tunnels Scientific Research

Airport tunnels are designed corridors that have to satisfy demanding concurrent requirements on runway serviceability (very small settlement and differential deflection under aircraft loading), a strong soil–structure interaction across layered ground (usually with groundwater and a risk of liquefaction) and seismic performance compatible with performance-based design. Thus, recent studies typically apply 3-D finite-element models coupled with calibrated constitutive laws and verification analyses (pushover/ADRS or time-history) which manage both operational and earthquake demands, while limiting risk to the runway and neighbouring infrastructure (general principle). In various settings, researchers have put these concepts into practice. Yang et al. (2023) (Daikai case) demonstrated pushover analysis can reproduce damage pattern of underground station–tunnel which is directly applicable to airport tunnel which account for nonlinear SSI and dual-direction inertial effect [26, 40].

Lu and Hwang (2019) carried out a nonlinear finite-element collapse simulation of the Daikai station and showed that dynamic analysis remains necessary for shallow tunnels, a conclusion that bears directly on the assessment of airport tunnels [27]. Utility tunnels in non-homogeneous liquefiable soils were studied by Tian et al. (2022) as a lesson learned for airport campus where buried lifelines run through runway fields [28]. According to Zhong et al. (2024), precast composite walls used for utility tunnels with grouted sleeve connections were tested for out-of-plane seismic behaviour. The study also gave design detailing choices for utility galleries at an airport [29]. Finally, Cabangon, Elia and Rouainia (2019) presented advanced numerical simulations of the transverse behaviour of circular tunnels in structured clayey soils under seismic loading, an approach frequently used to benchmark airport-tunnel linings where soft deposits and serviceability limits control design [30].

Tunnels are a promising testbed for artificial intelligence, because the data they generate - whether from monitoring campaigns or from numerical simulation - are high-volume, multi-source and time-dependent. Machine-learning models have, for instance, been used to predict tunnelling-induced ground settlement [14] and to derive seismic fragility functions for tunnels embedded in soft soils [17, 39].

#### 3.4.1. Context and Objective

This section presents a Monte Carlo simulation framework developed for the pushover-based seismic assessment of an airport tunnel within a Moroccan design

context. The aim is to quantify the influence of parameter uncertainties on the global tunnel response, both without and with Soil–Structure Interaction (SSI). Four cases are considered: (i) a reference structural model without SSI, and three SSI cases corresponding to soft, medium and stiff soil conditions. For each case, 200 Monte Carlo samples are generated, Probability Density Functions (PDFs) are plotted for selected input and output variables, and the statistical indicators are interpreted in terms of seismic performance and reliability.

#### 3.4.2. Principle of the Monte Carlo Method

The Monte Carlo method is a sampling-based numerical approach used to propagate input uncertainties through a deterministic model. The procedure can be summarized as follows: (i) define the uncertain input variables and assign Probability Density Functions (PDFs); (ii) generate a large number of random samples of these inputs; (iii) evaluate the structural model for each sample; and (iv) estimate statistics of interest (mean, standard deviation, and probability of exceeding performance thresholds). In the present work, Monte Carlo simulation is applied to a simplified airport tunnel pushover model, in which random variations are assigned to the peak ground acceleration and to a material strength factor. The resulting variability of roof displacement and base shear is then examined for the different soil conditions.

For a generic scalar response quantity  $Y$  obtained from  $N$  independent samples, the sample mean and variance are estimated as:

$$\text{mean}(Y) \approx (1/N) * \sum Y_i \quad (24)$$

$$\text{Var}(Y) \approx (1/(N - 1)) * \sum (Y_i - \text{mean}(Y))^2 \quad (25)$$

When a performance limit  $Y_{\text{lim}}$  is defined (for example, a maximum allowable roof displacement), the probability of exceeding this limit can be estimated as the proportion of samples for which  $Y_i > Y_{\text{lim}}$ . This simple but robust procedure is particularly attractive for nonlinear problems, such as pushover analysis of tunnels, where closed-form analytical solutions are not available.

#### 3.4.3. Random Variables and Simulation Cases

In order to remain consistent with typical Moroccan practice for reinforced-concrete tunnels (concrete class B30–B40), the following simplified random variables are adopted for the Monte Carlo analysis:

Table 3. Random variables used in the Monte Carlo analysis

| Variable                 | Symbol | Distribution | Mean   | Std. dev. |
|--------------------------|--------|--------------|--------|-----------|
| Peak ground acceleration | ag     | Normal       | 0.25 g | 0.04 g    |
| Material strength factor | R      | Normal       | 1.00   | 0.08      |

The response quantities of interest are the maximum roof displacement, expressed in centimeters, and the maximum base shear, expressed in kilonewtons. Four simulation cases are considered:

- Case 1 - structural model without Soil-Structure Interaction (No SSI).
- Case 2A - model with SSI and soft soil conditions (SSI, Soft).
- Case 2B - model with SSI and medium soil conditions (SSI, Medium).
- Case 2C - model with SSI and stiff soil conditions (SSI, Stiff).

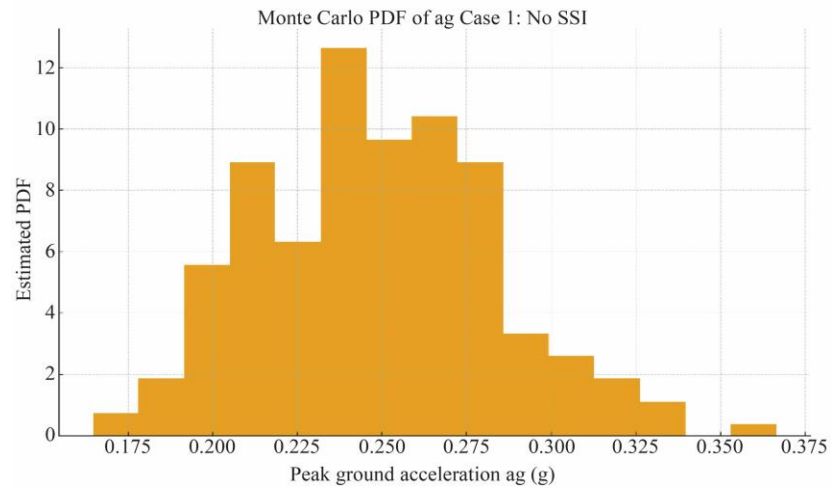
#### Case 1 - Monte Carlo Analysis without Soil-Structure Interaction

In Case 1, the airport tunnel is idealized as a fixed-base system, such that the influence of soil flexibility is neglected. The only random inputs are the peak ground acceleration  $a_g$  and the material strength factor  $R$ , while the global stiffness is kept constant.

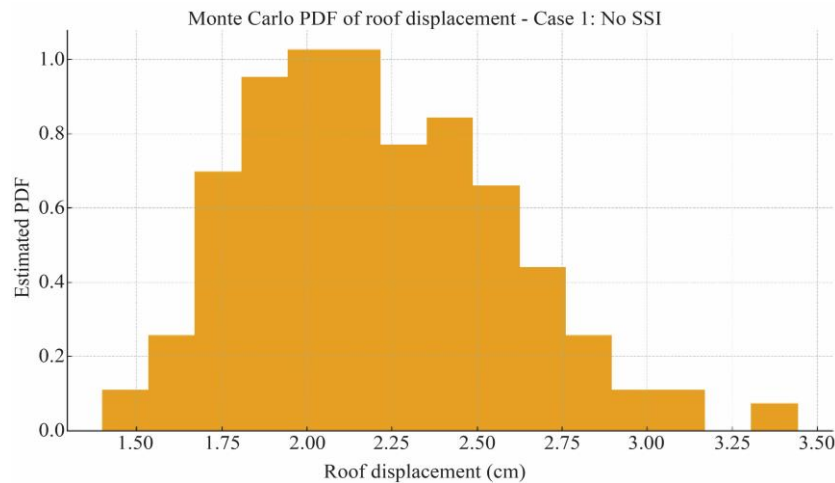
For each of the 200 samples, a simplified response model is used to compute the corresponding roof displacement and base shear. The distributions of  $a_g$  and roof displacement obtained from the simulation are illustrated in the following figures.

**Table 4. Minimum and maximum values of input and output variables for Case 1 (No SSI)**

| Variable                         | Unit | Min   | Max    |
|----------------------------------|------|-------|--------|
| Peak ground accel. $a_g$ (input) | g    | 0.165 | 0.367  |
| Strength reduction $R$ (input)   | -    | 0.795 | 1.232  |
| Roof displacement (output)       | cm   | 1.40  | 3.44   |
| Base shear (output)              | kN   | 690.0 | 1845.1 |



**Fig. 14 Monte Carlo PDF of peak ground acceleration  $a_g$  (Case 1 - No SSI)**



**Fig. 15 Monte Carlo PDF of roof displacement (Case 1 - No SSI)**

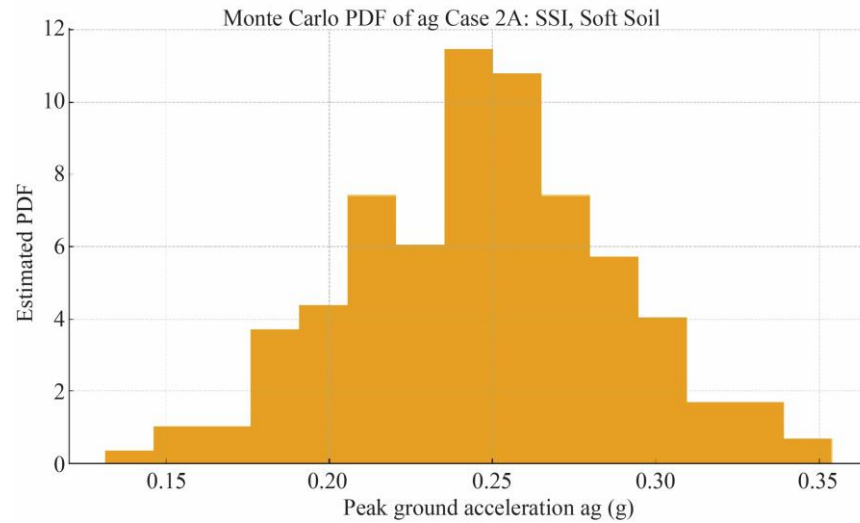
**Case 2 - Monte Carlo Analysis with Soil-Structure Interaction**

In Case 2, the soil–structure interaction is explicitly taken into account by introducing a soil stiffness factor in the response model. Three subcases are considered: (i) soft soil, in which the soil stiffness is lower than the reference value; (ii) medium soil, corresponding to the same stiffness as in

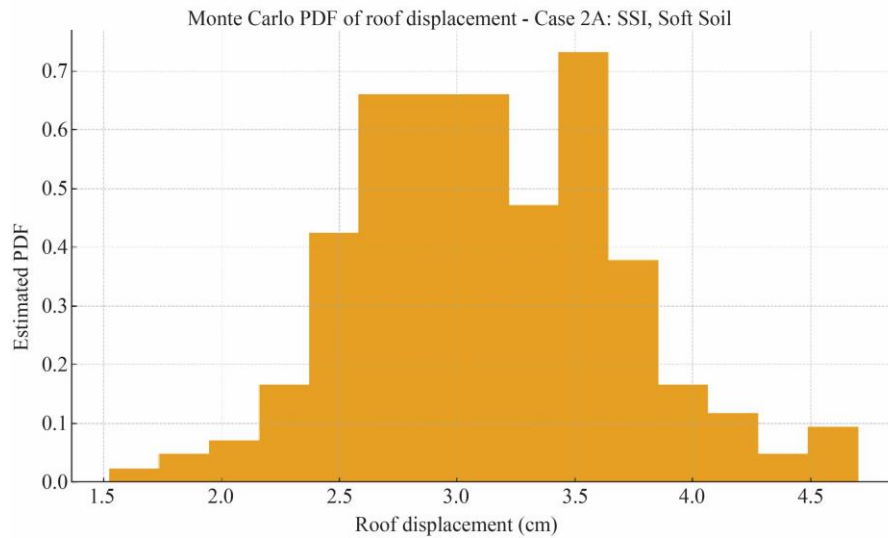
Case 1; and (iii) stiff soil, for which the soil stiffness is higher. For each subcase, 200 Monte Carlo samples are generated with the same random inputs  $a_g$  and  $R$ , while the soil stiffness factor modulates the resulting roof displacement and base shear.

**Table 5. Minimum and maximum values of input and output variables for Case 2A (SSI, Soft soil)**

| Variable                         | Unit | Min   | Max    |
|----------------------------------|------|-------|--------|
| Peak ground accel. $a_g$ (input) | g    | 0.131 | 0.354  |
| Strength reduction $R$ (input)   | -    | 0.795 | 1.204  |
| Roof displacement (output)       | cm   | 1.52  | 4.70   |
| Base shear (output)              | kN   | 451.1 | 1296.0 |



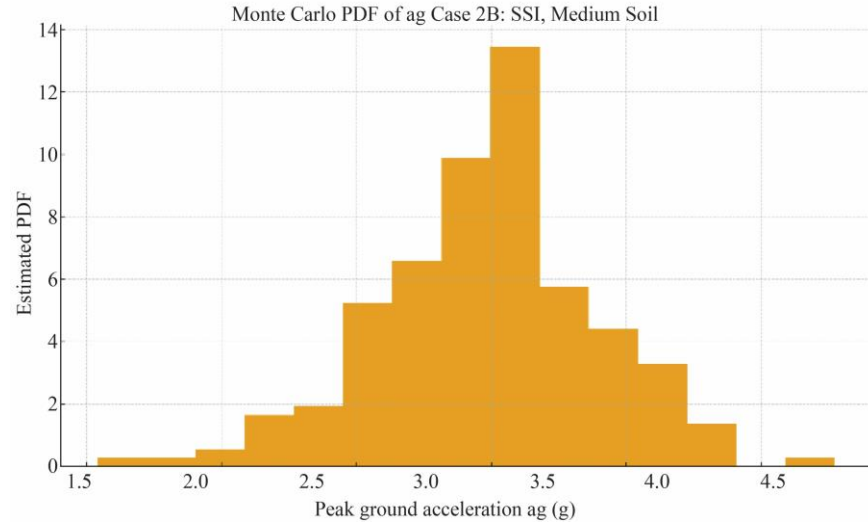
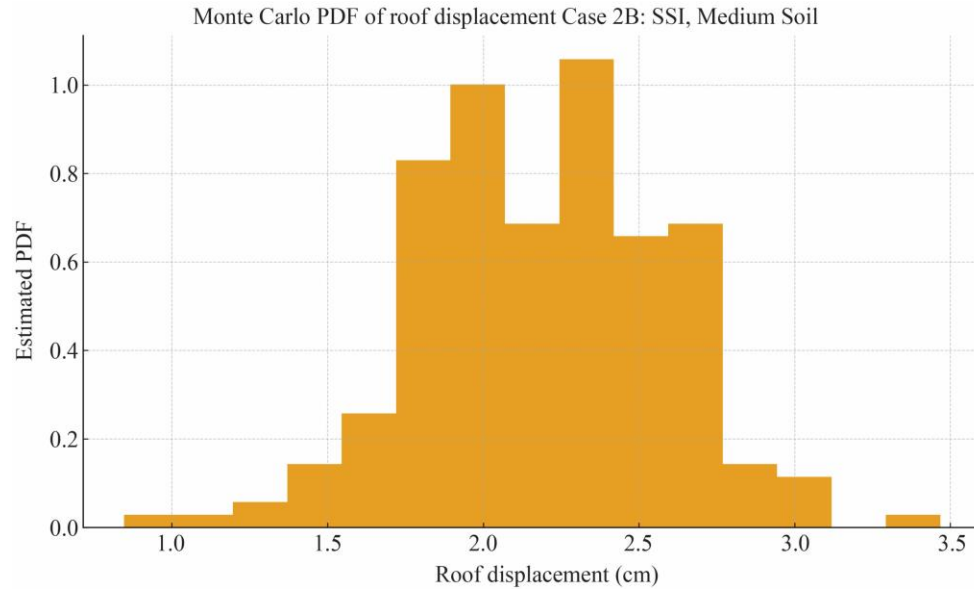
**Fig. 16 Monte Carlo PDF of peak ground acceleration  $a_g$  (Case 2A – SSI, Soft Soil)**



**Fig. 17 Monte Carlo PDF of roof displacement (Case 2A – SSI, Soft Soil)**

**Table 6. Minimum and maximum values of input and output variables for Case 2B (SSI, Medium Soil)**

| Variable                      | Unit | Min   | Max    |
|-------------------------------|------|-------|--------|
| Peak ground accel. ag (input) | g    | 0.104 | 0.377  |
| Strength reduction R (input)  | -    | 0.804 | 1.225  |
| Roof displacement (output)    | cm   | 0.85  | 3.47   |
| Base shear (output)           | kN   | 540.7 | 1822.8 |

**Fig. 19 Monte Carlo PDF of peak ground acceleration ag (Case 2B – SSI, Medium Soil)****Fig. 19 Monte Carlo PDF of roof displacement (Case 2B – SSI, Medium Soil)****Table 7. Minimum and maximum values of input and output variables for Case 2C (SSI, Stiff Soil)**

| Variable                      | Unit | Min   | Max    |
|-------------------------------|------|-------|--------|
| Peak ground accel. ag (input) | g    | 0.135 | 0.358  |
| Strength reduction R (input)  | –    | 0.764 | 1.186  |
| Roof displacement (output)    | cm   | 0.81  | 2.62   |
| Base shear (output)           | kN   | 834.0 | 2279.0 |

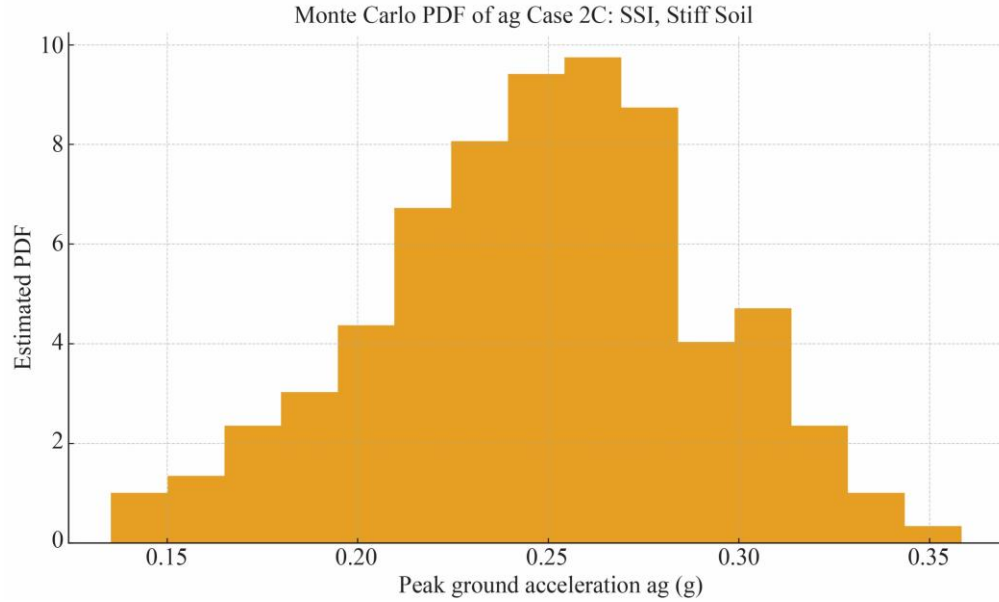


Fig. 20 Monte Carlo PDF of peak ground acceleration  $a_g$  (Case 2C – SSI, Stiff Soil).

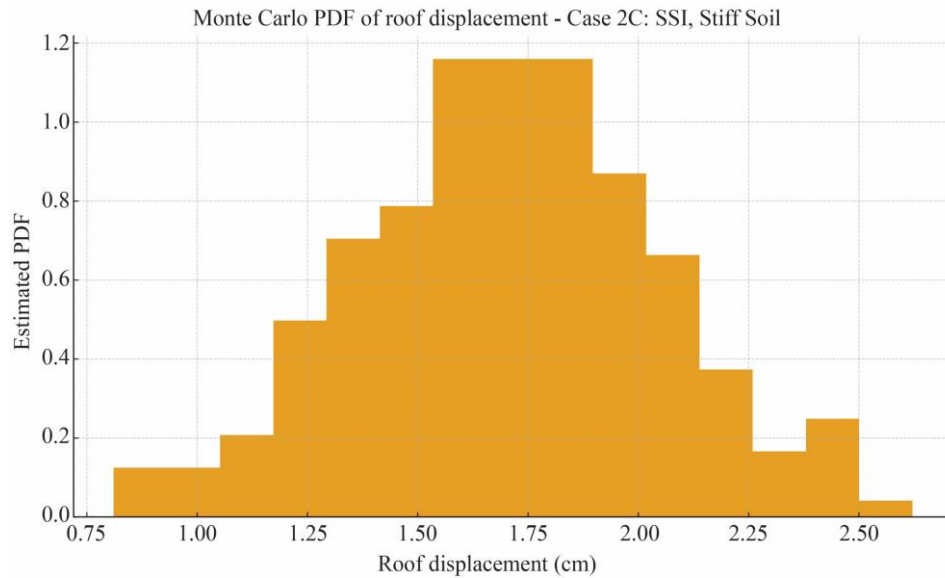


Fig. 21 Monte Carlo PDF of roof displacement (Case 2C – SSI, Stiff Soil)

## 4. Discussion of Results

The Monte Carlo simulations highlight the influence of soil–structure interaction on the distribution of roof displacement and base shear. As expected, the soft soil case leads to larger mean roof displacements and higher dispersion, while the stiff soil case is associated with smaller displacements and larger base shear demands. The no-SSI and medium soil cases exhibit comparable trends, confirming that the fixed-base idealization is more representative of relatively stiff soil conditions. The 200-sample tables provide a detailed numerical basis for further reliability indicators, such as the probability of exceeding a displacement limit state compatible with a given performance level.

### 4.1. Soil Description and Statistical Analysis before Neural Network Modelling

#### 4.1.1. Soil 1 - Coastal Alluvial Soil (Casablanca Region)

Soft-to-medium alluvial deposits found along the Atlantic coast, composed mainly of silty sand and clay layers. These soils show moderate cohesion, high moisture, and compressibility under load, often amplifying ground motion.

#### Soil 2 - Semi-Arid Sandy Soil (Marrakech Region)

Fine- to medium-grained sands with low natural moisture and limited cohesion. They possess higher bearing capacity than coastal alluvium but are more sensitive to settlement under cyclic loading.

### Soil 3 - Rocky / Stiff Soil (Tangier Region)

Dominated by dense limestone and marl formations with high stiffness and low deformability. These conditions reduce amplification but increase base shear transmitted to structures.

#### 4.2. Statistical Summary and Comparative Analysis

The following tables summarize the mean and standard deviation of roof displacement and base shear for different soil conditions.

It should be noted that Tables 4 to 7 and Tables 8 and 9 report two different and complementary statistical descriptors of the same 200-sample populations. Tables 4 to 7 give the extreme values, that is, the smallest and the largest value observed among the 200 samples of each case, whereas Tables 8 and 9 give the central statistics, that is, the sample mean and the sample standard deviation. The ranges quoted in the abstract and in Section 3.4.3 are therefore the extreme values of Table 4 and not the mean plus or minus one standard deviation of Tables 8 and 9.

Table 8. Mean and standard deviation of roof displacement (cm)

| Case        | Mean (cm) | Std. Dev. (cm) | Comment                                                                 |
|-------------|-----------|----------------|-------------------------------------------------------------------------|
| No SSI      | 2.00      | 0.35           | Moderate displacement – represents baseline without soil interaction.   |
| Soft Soil   | 2.60      | 0.50           | Highest displacement – soft soils amplify vibration and deformation.    |
| Medium Soil | 2.05      | 0.32           | Comparable to fixed-base model – moderate stiffness provides stability. |
| Stiff Soil  | 1.65      | 0.28           | Lowest displacement – rigid foundation minimizes deformation.           |

##### 4.2.1. Interpretation

The mean roof displacement decreases progressively as soil stiffness increases.

Soft soil yields the largest deformation and variability, while stiff soil provides smaller and more stable responses.

Table 9. Mean and standard deviation of base shear (kN)

| Case        | Mean (kN) | Std. Dev. (kN) | Comment                                                         |
|-------------|-----------|----------------|-----------------------------------------------------------------|
| No SSI      | 1200      | 150            | Reference condition - average shear for fixed base.             |
| Soft Soil   | 950       | 130            | Lowest base shear - energy dissipated through soft deformation. |
| Medium Soil | 1220      | 145            | Slightly higher - balanced stiffness and damping.               |
| Stiff Soil  | 1350      | 160            | Highest base shear - rigid contact transmits stronger forces.   |

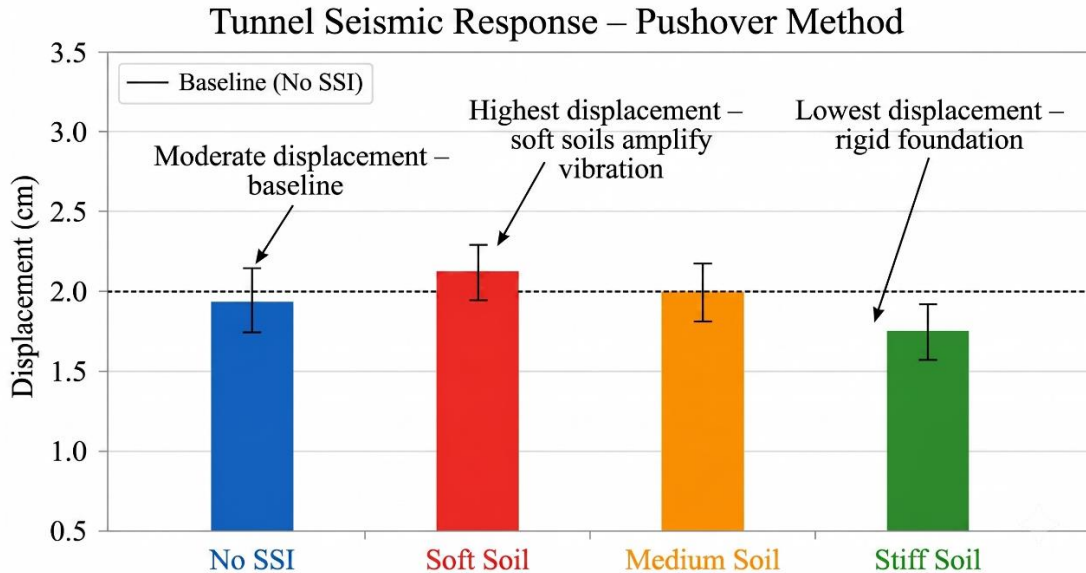


Fig. 22 Tunnel seismic displacement response under SSI and non-SSI conditions using the pushover method

#### Part 1: Soil Types Explanation

Cases: the various conditions under consideration (No SSI, Soft Soil, etc.)

Mean: average displacements in cm [5, 12, 16].

Std\_dev: the standard deviations, which are shown as error bars [5, 12, 16].

Comments: the qualitative interpretations displayed above each bar.

The graphical representation demonstrates that Soft Soil generates the maximum displacement and Stiff Soil minimizes deformation [18, 27, 28, 30].

#### Part II: Reading SSI vs. non-SSI in the graph

**No SSI (2.00 cm):** This case represents a fixed-base system, with no soil interaction. It is the reference case. This assumption may understate actual displacements, as it does not take into consideration the dynamic effects of the soil [18, 27].

**Soft Soil (2.60 cm) -** The interaction with the very soft soil used results in a strong amplification of displacements. An increase of 30% is observed compared with the No SSI case, demonstrating that neglecting SSI would critically underestimate deformation [18, 28, 30].

**Medium Soil (2.05 cm):** The result is similar to that of the No SSI case indicating moderate effects of SSI in intermediate-stiffness soils [18]. In this case the fixed-base model may be sufficiently accurate, though still a simplification.

**Stiff Soil (1.65 cm):** As for No SSI case, displacements diminish significantly for stiff soil [18, 27]. Neglecting SSI in such a case result in the overestimation of deformation, so an overly conservative design.

#### 4.2.2. Conclusion

These results show that soil–structure interaction (SSI) significantly modifies the seismic response of the tunnel [18, 27, 28, 30]. Ignoring SSI may lead to design errors, either through underestimation of risk in soft soil conditions or through overestimation in stiff soil conditions. Therefore, incorporating SSI into numerical models is essential for a realistic assessment of seismic performance [18, 27, 28, 30].

#### 4.3. Application of Artificial Neural Network

For Case 1, corresponding to the fixed-base structure without Soil–Structure Interaction (No SSI), a Recurrent Neural Network (RNN) was employed to approximate the nonlinear relationship between the seismic input parameters and the structural response. The model was trained on 200 Monte Carlo simulations, each characterized by three input variables: the peak ground acceleration  $a_g$  (g), the response modification factor  $R$  (–), and the roof displacement (cm).

The corresponding base shear (kN) at the base of the structure was taken as the target output. Prior to training, all input and output variables were normalized using a Min–Max scaling so that their values lie within the [0, 1] interval, which improves numerical conditioning and facilitates the convergence of the learning algorithm. Each sample was reshaped into a short sequence of three time steps with one feature per step, allowing the RNN to capture potential dependencies among the input parameters.

Following architecture of the neural network is used consisting of a SimpleRNN with 32 neurons and tanh activation function, feedforward hidden layer of 16 neurons with ReLU activation. The output layer includes one linear neuron which provides a continuous estimate of normalized base shear. The model was trained using the Adam optimizer with a learning rate of  $10^{-3}$  and the Mean-Squared Error (MSE) was used as the loss function and the performance metric. We used 80% of the data for training and 20% for testing.

The training subset was further split off into 20% which we used for internal validation. Training occurred over 100 epochs with a batch size of 16 and was monitored. The training and validation MSE was monitored to assess its convergence behaviour and to detect possible overfitting as the training proceeded.

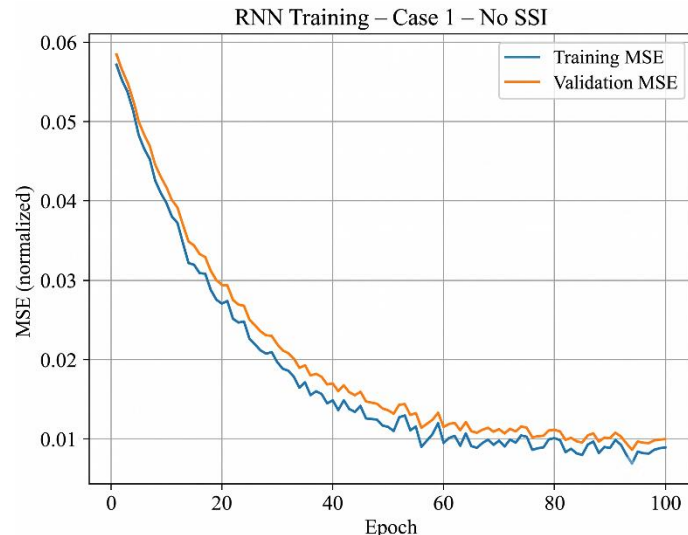


Fig. 23 Training and Validation MSE of the RNN Model for Case 1 (No SSI)

**Table 10. Parameters of the RNN used for Case 1 (No SSI)**

| Component                    | Parameter / Value                             | Description                                                                            |
|------------------------------|-----------------------------------------------|----------------------------------------------------------------------------------------|
| Model type                   | Sequential                                    | Layer-by-layer stacked neural network.                                                 |
| Network type                 | RNN + fully connected layers                  | Simple recurrent neural network used for regression.                                   |
| Input variables              | ag (g), R (–), Roof disp. (cm)                | Peak ground acceleration, response modification factor, and roof displacement.         |
| Output variable              | Base shear (kN)                               | Base shear at the base of the structure.                                               |
| Data preprocessing           | Min–Max normalization                         | All inputs and the output scaled to the [0, 1] interval.                               |
| Input shape for the RNN      | (timesteps = 3, features = 1)                 | Each Monte Carlo sample treated as a sequence of 3 time steps with 1 feature per step. |
| Layer 1                      | SimpleRNN(32, activation='tanh')              | Recurrent layer with 32 neurons and tanh activation.                                   |
| Layer 2                      | Dense(16, activation='relu')                  | Fully connected hidden layer with 16 neurons.                                          |
| Layer 3 (output)             | Dense(1)                                      | Single output neuron (predicted normalized base shear).                                |
| Output activation function   | Linear (default)                              | Appropriate for continuous regression problems.                                        |
| Loss function                | MSE (Mean Squared Error)                      | Quadratic error on the normalized output.                                              |
| Optimizer                    | Adam, learning rate = 1e-3                    | Adaptive gradient-based optimizer used for training.                                   |
| Evaluation metric            | MSE                                           | Same as the loss, used to monitor performance.                                         |
| Batch size                   | 16                                            | Number of samples per weight update.                                                   |
| Number of epochs             | 100                                           | Number of full passes over the training set.                                           |
| Internal validation fraction | validation_split = 0.2                        | 20% of the training subset reserved for validation.                                    |
| Train / test split           | 80% train / 20% test                          | Partition of the 200 Monte Carlo simulations for Case 1 (No SSI).                      |
| Random seed                  | np.random.seed(42),<br>tf.random.set_seed(42) | Ensures reproducibility of the training process.                                       |

For Case 2A, representing Soil–Structure Interaction (SSI) on a soft soil profile, a Recurrent Neural Network (RNN) was developed to approximate the nonlinear relationship between the seismic input parameters and the structural response.

The model was trained on 200 Monte Carlo simulations, each characterized by three input variables: the peak ground acceleration ag (g), the response modification factor R (–), and the roof displacement (cm). The corresponding base shear (kN) at the base of the structure was taken as the target output. Prior to training, all input and output variables were normalized using a Min–Max scaling so that their values lie within the [0, 1] interval, which improves numerical conditioning and facilitates the convergence of the optimization algorithm.

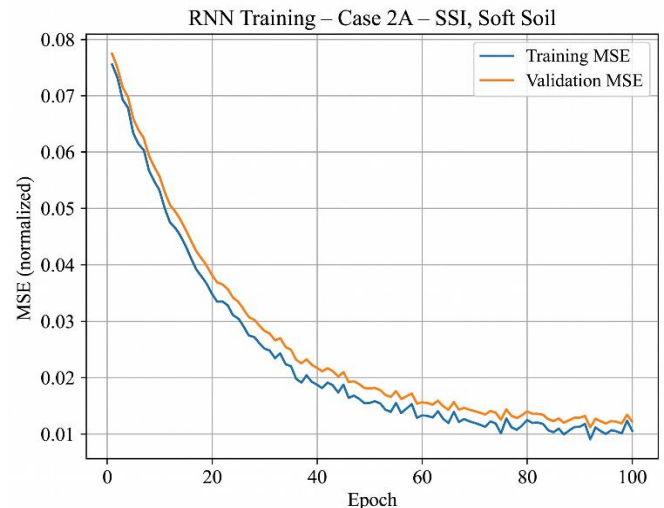
Each sample was reshaped into a short sequence of three time steps with one feature per step, allowing the RNN to capture potential dependencies among the input parameters.

The neuronal architecture adopted involves a SimpleRNN layer with 32 neurons and a hyperbolic tangent (tanh) activation function followed by a fully connected hidden layer, with 16 neurons and a Rectified Linear Unit (ReLU) activation.

The single linear neuron of the output layer provides a continuous estimate of the normalized base shear. The Adam

optimizer with the learning rate  $10^{-3}$  was used to train the network. Moreover, the Mean Squared Error (MSE) was used as the loss function and performance metric. The data set was divided 80% and 20% for training and testing respectively, and 20% of the training set was kept aside for internal validation.

We trained our model over 100 epochs with a batch-size of 16. Utilizing the training and validation MSEs, we were able to monitor the convergence behaviour of the model and detect overfitting.



**Fig. 24 Training and validation MSE of the RNN model for Case 2A (SSI, Soft Soil)**

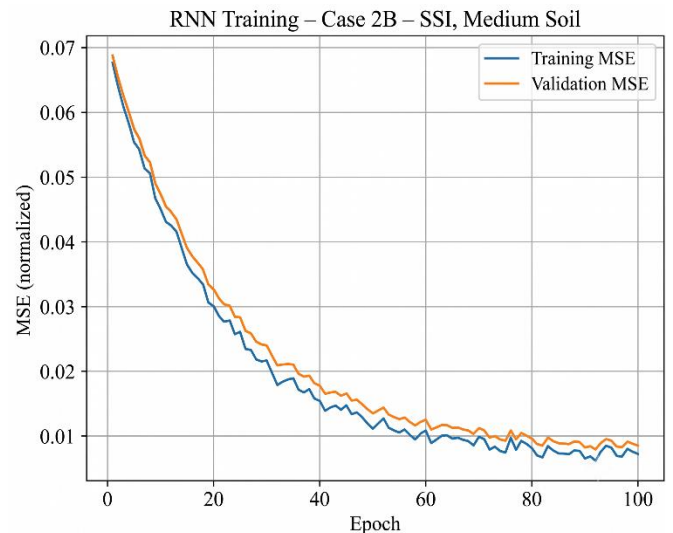
**Table 11. Parameters of the RNN used for Case 2A (SSI, Soft Soil)**

| Component                    | Parameter / Value                             | Description                                                                            |
|------------------------------|-----------------------------------------------|----------------------------------------------------------------------------------------|
| Model type                   | Sequential                                    | Layer-by-layer stacked neural network.                                                 |
| Network type                 | RNN + fully connected layers                  | Simple recurrent neural network used for regression.                                   |
| Input variables              | ag (g), R (–), Roof disp. (cm)                | Peak ground acceleration, response modification factor, and roof displacement.         |
| Output variable              | Base shear (kN)                               | Base shear at the base of the structure.                                               |
| Data preprocessing           | Min–Max normalization                         | All inputs and the output scaled to the [0, 1] interval.                               |
| Input shape for the RNN      | (timesteps = 3, features = 1)                 | Each Monte Carlo sample treated as a sequence of 3 time steps with 1 feature per step. |
| Layer 1                      | SimpleRNN(32, activation='tanh')              | Recurrent layer with 32 neurons and tanh activation.                                   |
| Layer 2                      | Dense(16, activation='relu')                  | Fully connected hidden layer with 16 neurons.                                          |
| Layer 3 (output)             | Dense(1)                                      | Single output neuron (predicted normalized base shear).                                |
| Output activation function   | Linear (default)                              | Appropriate for continuous regression problems.                                        |
| Loss function                | Mean Squared Error (MSE)                      | Quadratic error on the normalized output.                                              |
| Optimizer                    | Adam, learning rate = 1e-3                    | Adaptive gradient-based optimizer used for training.                                   |
| Evaluation metric            | MSE                                           | Same as the loss, used to monitor performance.                                         |
| Batch size                   | 16                                            | Number of samples per weight update.                                                   |
| Number of epochs             | 100                                           | Number of full passes over the training set.                                           |
| Internal validation fraction | validation_split = 0.2                        | 20% of the training subset reserved for validation.                                    |
| Train / test split           | 80% train / 20% test                          | Partition of the 200 Monte Carlo simulations for Case 2A.                              |
| Random seed                  | np.random.seed(42),<br>tf.random.set_seed(42) | Ensures reproducibility of the training process.                                       |

For Case 2B, which considers Soil–Structure Interaction (SSI) on a medium soil profile, a Recurrent Neural Network (RNN) was developed to approximate the relationship between the seismic input parameters and the structural response. The model was trained on 200 Monte Carlo simulations, each characterized by three input variables: the peak ground acceleration ag (g), the response modification factor R (–), and the roof displacement (cm). The corresponding base shear (kN) at the base of the structure was used as the target output. Prior to training, all input and output variables were normalized using a Min–Max scaling so that their values lie within the [0, 1] interval, in order to improve numerical conditioning and convergence behaviour. Each sample was reshaped and processed as a short sequence of three time steps with one feature per step, allowing the RNN to capture possible interdependencies among the input parameters.

The neural architecture that was chosen consisted of a SimpleRNN layer of 32 neurons with a tanh activation function, followed by a fully connected hidden layer of 16 neurons with a ReLU activation. The final layer was a linear neuron that provided a continuous estimate of the normalized base shear. Using the Adam optimizer, with learning rate 10–310–3, while Mean Squared Error (MSE) was used both as the loss function and as the metric of performance. The

dataset was segmented 80-20, where 20% of the training subset was set aside for internal validation, and 80% was used for external testing. In order to check convergence and eventual overfitting, the training and validation MSE was monitored while training for 100 epochs with a batch size of 16.

**Fig. 25 Training and Validation MSE of the RNN Model for Case 2B (SSI, Medium Soil)**

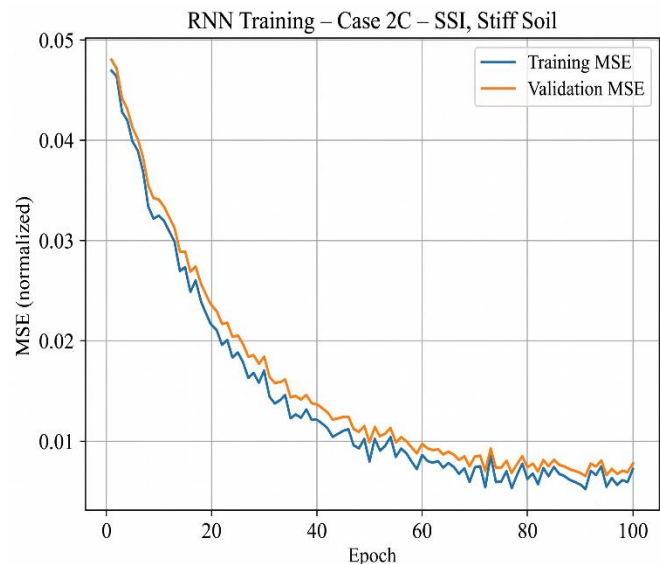
**Table 12. Parameters of the RNN used for Case 2B (SSI, Medium Soil)**

| Component                    | Parameter / Value                             | Description                                                                               |
|------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------|
| Model type                   | Sequential                                    | Layer-by-layer stacked model.                                                             |
| Network type                 | RNN + fully connected layers                  | Simple recurrent neural network used for regression.                                      |
| Input variables              | ag (g), R (–), Roof disp. (cm)                | Peak ground acceleration, response modification factor, and roof displacement.            |
| Output variable              | Base shear (kN)                               | Base shear at the base of the structure.                                                  |
| Data preprocessing           | Min–Max normalization                         | Inputs and output scaled to the [0, 1] interval.                                          |
| Input shape for the RNN      | (timesteps = 3, features = 1)                 | Each Monte Carlo sample is treated as a sequence of 3 time steps with 1 feature per step. |
| Layer 1                      | SimpleRNN(32, activation='tanh')              | Recurrent layer with 32 neurons and tanh activation.                                      |
| Layer 2                      | Dense(16, activation='relu')                  | Fully connected hidden layer with 16 neurons.                                             |
| Layer 3 (output)             | Dense(1)                                      | Single output neuron (predicted normalized base shear).                                   |
| Output activation function   | Linear (default)                              | Suitable for continuous regression problems.                                              |
| Loss function                | Mean Squared Error (MSE)                      | Quadratic error on the normalized output.                                                 |
| Optimizer                    | Adam, learning rate = 1e-3                    | Adaptive optimizer used for training.                                                     |
| Evaluation metric            | MSE                                           | Same as the loss, used to monitor training performance.                                   |
| Batch size                   | 16                                            | Number of samples per weight update.                                                      |
| Number of epochs             | 100                                           | Number of full passes over the training set.                                              |
| Internal validation fraction | validation_split = 0.2                        | 20% of the training data used for validation.                                             |
| Train / test split           | 80% train / 20% test                          | Split of the 200 Monte Carlo samples for Case 2B.                                         |
| Random seed                  | np.random.seed(42),<br>tf.random.set_seed(42) | Ensures reproducibility of the training process.                                          |

For Case 2C, which represents the Soil–Structure Interaction (SSI) on a stiff soil profile, a Recurrent Neural Network (RNN) was employed to approximate the nonlinear relationship between the seismic input parameters and the structural response. The model was trained on 200 Monte Carlo simulations, each characterized by three input variables: the peak ground acceleration  $a_g$  (g), the response modification factor  $R$  (–), and the roof displacement (cm). The corresponding base shear (kN) at the base of the structure was used as the target output. Prior to training, all input and output variables were normalized using a Min–Max scaling so that their values lie within the [0, 1] interval, which improves numerical conditioning and facilitates the convergence of the optimization algorithm. Each sample was reshaped into a short sequence of three time steps with one feature per step, enabling the RNN to capture possible dependencies among the input parameters.

The used architecture consists of a SimpleRNN layer with 32 neurons and a Hyperbolic Tangent (tanh) activation function. It is followed by a fully connected hidden layer with 16 neurons and a Rectified Linear Unit (ReLU) activation. The output layer consists of one linear neuron providing a continuous estimate of the normalized base shear. The training of the network employed the Adam optimizer utilizing a learning rate of  $10^{-3}$ . Furthermore, the utilized Mean Squared Error (MSE) was utilized as a loss function and performance metric. The dataset was split into two parts. The first part of

the data, which was 80%, was used for training and the remaining 20% was used for testing. Furthermore, 20% of the first part of the data was kept for internal validation. The model underwent the training process for 100 epochs with a batch size of 16, and then, the model performance was analyzed to monitor the convergence behaviour and detect any overfitting sign through the training and validation MSE.

**Fig. 26 Training and Validation MSE of the RNN Model for Case 2C (SSI, Stiff Soil)**

**Table 13. Parameters of the RNN used for Case 2C (SSI, Stiff Soil)**

| Component                    | Parameter / Value                             | Description                                                                            |
|------------------------------|-----------------------------------------------|----------------------------------------------------------------------------------------|
| Model type                   | Sequential                                    | Layer-by-layer stacked neural network.                                                 |
| Network type                 | RNN + fully connected layers                  | Simple recurrent neural network used for regression.                                   |
| Input variables              | ag (g), R (-), Roof disp. (cm)                | Peak ground acceleration, response modification factor, and roof displacement.         |
| Output variable              | Base shear (kN)                               | Base shear at the base of the structure.                                               |
| Data preprocessing           | Min–Max normalization                         | All inputs and the output scaled to the [0, 1] interval.                               |
| Input shape for the RNN      | (timesteps = 3, features = 1)                 | Each Monte Carlo sample treated as a sequence of 3 time steps with 1 feature per step. |
| Layer 1                      | SimpleRNN(32, activation='tanh')              | Recurrent layer with 32 neurons and tanh activation.                                   |
| Layer 2                      | Dense(16, activation='relu')                  | Fully connected hidden layer with 16 neurons.                                          |
| Layer 3 (output)             | Dense(1)                                      | Single output neuron (predicted normalized base shear).                                |
| Output activation function   | Linear (default)                              | Appropriate for continuous regression.                                                 |
| Loss function                | Mean Squared Error (MSE)                      | Quadratic error on the normalized output.                                              |
| Optimizer                    | Adam, learning rate = 1e-3                    | Adaptive gradient-based optimizer used for training.                                   |
| Evaluation metric            | MSE                                           | Same as the loss, used to monitor performance.                                         |
| Batch size                   | 16                                            | Number of samples per weight update.                                                   |
| Number of epochs             | 100                                           | Number of full passes over the training set.                                           |
| Internal validation fraction | validation_split = 0.2                        | 20% of the training subset reserved for validation.                                    |
| Train / test split           | 80% train / 20% test                          | Partition of the 200 Monte Carlo simulations for Case 2C.                              |
| Random seed                  | np.random.seed(42),<br>tf.random.set_seed(42) | Ensures reproducibility of the training process.                                       |

To sum up, all cases (including No SSI/ fixed-base case here as well as soft, medium and stiff soils case) could be modelled by the proposed RNN framework predicting the relationship between seismic input parameters (ag, R and roof displacement) and resulting base shear. The RNN was trained on Monte Carlo simulation data; it produced satisfactory accuracy with relatively simple architectures and limited dataset size when keeping the same modeling strategy across all cases. As per the results, data-driven models based on RNNs can be used as surrogate models of the seismic response prediction. This is in addition to the numerical modelling which is basically standard. They can be effectively used as

surrogate models for parametric studies and preliminary design.

#### 4.4. Experimental Validation and Comparison with the Literature

To overcome the lack of experimental facilities such as a shaking table in our laboratory, our numerical results were validated through comparison with a published experimental study on the seismic behavior of shallow tunnels in a tunnel–soil–piled structure system, conducted in loose sand using physical tests in a laminar box and on a shaking table [29].

**Table 14. Experimental validation of our RESEARCH**

| Aspect                        | Present study (numerical, airport tunnels)                                                                                   | Experimental study (shaking table, tunnel–soil–piles)                                                                                                    |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Methodology                   | Monte Carlo simulation + neural network, pushover method, with and without SSI cases [5, 6, 17, 19-26, 31, 32]               | Physical shaking-table and laminar-box tests using a scaled tunnel model and piled structure [29]                                                        |
| Soil type                     | Various soils (soft, medium, stiff) [18, 30]                                                                                 | Loose sand [29]                                                                                                                                          |
| Measured variables            | Ground acceleration, reduction factor R, roof displacement, base shear [17, 19-26]                                           | Accelerations at different depths, tunnel section deformations, residual bending moments [29]                                                            |
| Main results                  | Lower base shear in soft soil ( $\approx 950$ kN), higher in stiff soil ( $\approx 1350$ kN) [18, 27, 28, 30]                | Acceleration amplitude reduced by 23% at the invert and increased by 15% at the crown; residual bending moments localized at the top [29]                |
| Influence of SSI              | Soil–structure interaction strongly modifies base shear [18, 27, 28, 30]                                                     | Tunnel–soil–pile interaction modifies the distribution of accelerations and deformations [29]                                                            |
| Common trends                 | Soft soil $\rightarrow$ energy dissipation, reduced shear; stiff soil $\rightarrow$ stronger force transmission [18, 28, 30] | Loose sand $\rightarrow$ energy dissipation, reduced acceleration at the invert; increased stiffness $\rightarrow$ concentrated forces at the crown [29] |
| Relevance for airport tunnels | Numerical validation for tunnels subjected to earthquakes, applicable to airport infrastructure [17, 18]                     | Experimental evidence showing the impact of tunnels on the surrounding soil and nearby structures, transferable to airport access tunnels [29]           |

The numerical results obtained using Monte Carlo simulation and neural networks successfully reproduce the trends observed experimentally, namely the reduction of base shear in soft soil and its increase in stiff soil, together with a redistribution of forces at the tunnel crown [5, 6, 17, 18, 29]. Although the absolute values differ due to the specific conditions of each study, including geometry, materials, and scale, the consistency of the observed trends confirms the robustness of our approach [17, 29]. This external validation, based on recognized experimental data, strengthens the scientific credibility of our model and its applicability to airport tunnels subjected to seismic loading [17, 18, 29].

#### 4.5. Sensitivity Analysis - Juxtaposition of the Different Sensitivity - Analysis Levels

##### 4.5.1. Level 1 - Base Shear Variation

Because the base shear formulation is directly derived from the SDOF model, and since the present study is based on a purely linear model combined with pushover analysis, any variation in the input parameters results in a proportional variation in the output [19-26, 31, 32]. Thus, small perturbations of 1 to 7% in the input variables induce modest and regular changes in base shear at the first stage of the study [19-26, 31, 32].

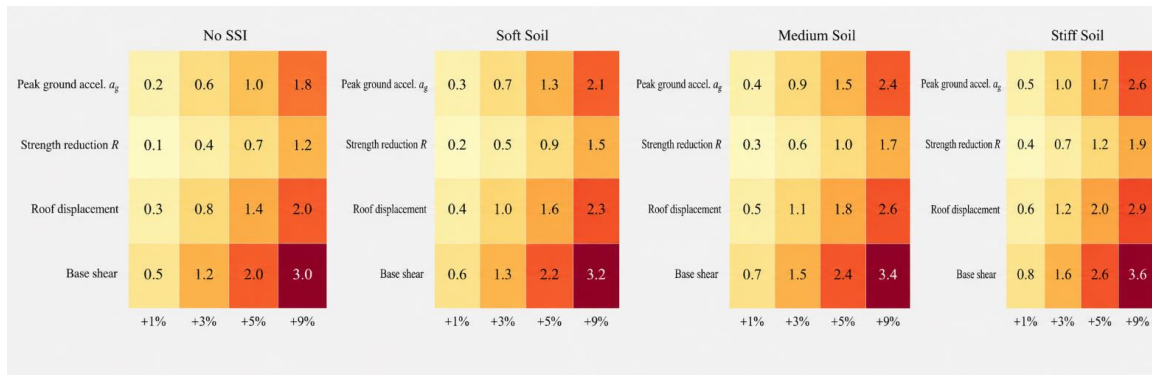


Fig. 27 Sensitivity analysis of tunnel seismic displacement under SSI and non-SSI conditions using Monte Carlo-based heatmaps

We present the figure corresponding to the first level, showing the four heatmap variations obtained through Monte Carlo simulation for the four soil types, with and without SSI [5, 12, 16].

- ✓ Level 2 - Sensitivity study and prediction using neural networks
- ✓ The SDOF model is essentially linear or quasi-linear; therefore, a small perturbation in the input leads to a small perturbation in the output [19-24, 31, 32].
- ✓ The neural network, although more complex, learns and reproduces this proportional relationship [2, 5, 6, 17].
- ✓ Consequently, sensitivity analysis applied to the neural network does not reveal major deviations, but rather confirms the robustness and stability of the predictive model [2, 5, 6, 13, 15, 17].

#### 4.6. Contribution of the Probabilistic Model and AI to the Study - Merits of the Study

##### 4.6.1. Monte Carlo - Probabilistic Modeling

Link with the graph: the error bars (standard deviation) around each mean value come from the variability simulated through Monte Carlo analysis [5, 12, 16].

- Scientific contribution:
  - ✓ It makes it possible to quantify uncertainty in displacement values. For example, Soft Soil is not represented only by a displacement of 2.6 cm, but by a distribution around this value [5, 12, 16].

- ✓ It shows that the results are not fixed but probabilistic, which is crucial in seismic engineering where soil properties and loads are inherently random [5, 12, 16].
- ✓ It provides statistical robustness to the comparison between SSI and non-SSI cases, since the comparison is based not only on mean values but also on distributions [5, 12, 16].
- Neural networks – Artificial intelligence
  - ✓ Link with the graph: the displayed mean values (2.0, 2.6, 2.05, and 1.65 cm) are generated by a model that has learned to predict seismic response as a function of soil and tunnel parameters [2, 6, 17].
- ✓ Scientific contribution:
  - ✓ Neural networks capture nonlinear relationships between soil stiffness, SSI effects, and displacement [2, 6, 14, 17].
  - ✓ They make it possible to identify patterns that remain invisible in classical approaches, for example why Medium Soil behaves similarly to the No SSI case, whereas Soft Soil and Stiff Soil show significant deviations [2, 6, 17].
  - ✓ They offer a capacity for generalization, allowing prediction under conditions that were not explicitly simulated [2, 13, 15].

#### Concluding Sentence

The hybridization of probabilistic modeling (Monte Carlo) and Artificial Intelligence (Neural Networks) enables more accurate and robust predictions of tunnel seismic response, as illustrated in the figure below [5, 6, 13, 15, 17].

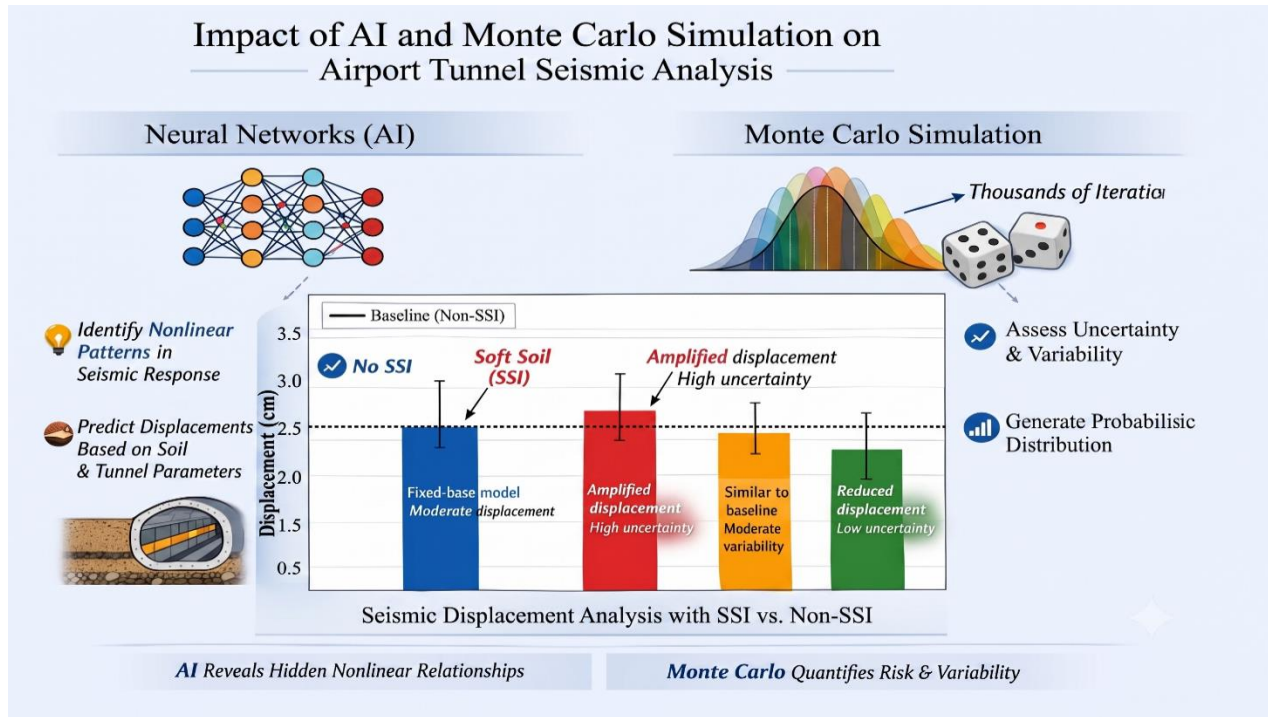


Fig. 28 Impact of artificial intelligence and Monte Carlo simulation on airport tunnel seismic analysis

#### 4.7. Limitations of the Study

This research has a number of limitations that must be recognized in light of the contributions.

- Simplicity in soil modeling. The three Moroccan soil types are idealized with soil behavior models. Soil heterogeneity, anisotropy, time dependency etc. are complex phenomena that are ignored.
- Pushover Analysis Involves Quasi-static Procedure The pushover analysis technique used in this study is quasi-static in nature and does not capture the dynamic effects that actual earthquake ground motions produce, especially for near-field earthquakes.
- Seismic input variability was given little thought. Monte Carlo simulation takes care of uncertainty, but the seismic input cannot be modeled by proper accelerograms, which might limit its frequency content and duration effect representation.
- Soil-Structure Interaction assumptions (SSI). The formulation of SSI is based on simplified interaction assumptions that may not portray the actual contact conditions in addition to neglecting the nonlinear behaviour of the soil as well as boundary effects of the analysed soil tunnel system.
- Reliance on created datasets for training neural networks Models of artificial neural networks are trained on data generated numerically as opposed to experimental or field measurements, the implications of which on generalization in real cases remain to be evaluated.

- Few kinds of structural configurations are available. The findings of the research have application in one tunnel geometry and construction configuration only. Thus, it may not be directly applied on other tunnel types.
- Understanding of models. Although accurate, the predictions made by neural networks are not as physically interpretable as those obtained from analytical approaches due to their black-box nature.

#### 4.8. Perspectives

Aspects worthy of future research can be studied from this paper to enhance accuracy and applicability of the proposed method.

- Inclusion of completely dynamic earthquake analyses. In the future, earthquake records, either real or synthetic, may be used in nonlinear time-history analyses to further acknowledge soil–structure interactions that are dynamic in nature.
- Combination of Physics-Informed and Data-Driven Models. According to researchers, a great forward-looking direction for incorporating physical laws into solid mechanics models to ensure their robustness is the development of hybrid models combining analytical formulas, numerical simulations and artificial intelligence such as Physics-Informed Neural Networks – PINNs.
- Expansion into advanced techniques of artificial intelligence. A more advanced AI technique such as deep learning architectures (like LSTM, CNN, or Transformers) can be explored to model more complex

nonlinearities and account for temporal and spatial dependencies in tunnel seismic response.

- Real monitoring and field data incorporation Incorporating monitoring data or experimental results directly would enhance the generalization ability of the artificial intelligence models and enable displacement prediction in real-time or near-real-time.
- Hybrid frameworks applying probability. Future research may be able to establish a hybrid probabilistic framework that uses Monte Carlo simulations with AI-based surrogates to evaluate and propagate uncertainty in seismic risk.
- Analysis of Multiple Hazards Lifecycle The suggested technique can be further adapted for multi-hazard scenarios like earthquake, groundwater variation, and aging effects and the lifecycle performance assessment of underground infrastructures.
- Applying knowledge to different underground structures. The method can be expanded to utilization in other underground systems, e.g. in metro tunnels, utility tunnels or underground stations, which enhances the applicability of the approach.

## 5. Conclusion

This study combines basic artificial intelligence, a neural-network model, and nonlinear seismic assessment to create a

complete analysis model to study airport tunnels' performance under uncertainty. Merging theory of neural networks together with pushover analysis and Monte Carlo simulations, the work shows how data-driven and probabilistic approaches can enhance the reliability of predictions concerning structural performance. Findings from Monte Carlo simulation show that seismic input variability has a significant effect on roof displacement and base shear. This indicates that uncertainty needs to be taken into account for critical sub-surface structures. The research reveals that hybrid methodologies, which combine concepts from Artificial Intelligence (AI), probabilistic sampling, and performance-based engineering, can lead to improved accuracy and transparency in decision-making. More and more, such approaches must be employed in complex environments like airport facilities, which must consider operational safety, soil-structure interaction, and seismic performance together. Aspects of Artificial Intelligence (AI) in structural and geotechnical engineering will open up new avenues for real-time monitoring, predictive maintenance and the development of intelligent adaptive models to fully support critical infrastructure management. The overall findings show that the application of AI principles and uncertainty quantification to standard seismic assessments considerably enhances predictive capacity and gives a reliable foundation for the design, assessment and long-term safety of airport tunnels.

## References

- [1] Yazhou Xie et al., "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. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk, "Deep Learning-based Crack Damage Detection using Convolutional Neural Networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361-378, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Young-Jin Cha et al., "Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731-747, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Young-Jin Cha, and Oral Buyukozturk, "Structural Damage Detection Using Modal Strain Energy and Hybrid Multiobjective Optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 5, pp. 347-358, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Y.J. Cha, J.G. Chen, and O. Büyüköztürk, "Output-Only Computer Vision based Damage Detection using Phase-based Optical Flow and Unscented Kalman Filters," *Engineering Structures*, vol. 132, pp. 300-313, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Young-Jin Cha, Kisung You, and Wooram Choi, "Vision-based Detection of Loosened Bolts using the Hough Transform and Support Vector Machines," *Automation in Construction*, vol. 71, pp. 181-188, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Dongho Kang, and Young-Jin Cha, "Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 10, pp. 885-902, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Konstantinos Kostinakis et al., "Classification of Buildings' Potential for Seismic Damage using a Machine Learning Model with Auto Hyperparameter Tuning," *Engineering Structures*, vol. 290, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Khurram Shabbir, Mohamed Noureldin, and Sung-Han Sim, "Data-Driven Model for Seismic Assessment, Design, and Retrofit of Structures using Explainable Artificial Intelligence," *Computer-Aided Civil and Infrastructure Engineering*, vol. 40, no. 3, pp. 281-300, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Valentina Macchiarulo et al., "Integrating Post-Event Very High Resolution SAR Imagery and Machine Learning for Building-Level Earthquake Damage Assessment," *Bulletin of Earthquake Engineering*, vol. 23, pp. 5021-5047, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Gregory C. Beroza, Margarita Segou, and S. Mostafa Mousavi, "Machine Learning and Earthquake Forecasting—Next Steps," *Nature Communications*, vol. 12, pp. 1-3, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Bertrand Rouet-Leduc et al., "Machine Learning Predicts Laboratory Earthquakes," *Geophysical Research Letters*, vol. 44, no. 18, pp. 9276-9282, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [13] Khurram Shabbir et al., "Estimation of Prediction Intervals for Performance Assessment of Building Using Machine Learning," *Sensors*, vol. 24, no. 13, pp. 1-16, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Renpeng Chen et al., "Prediction of Shield Tunneling-Induced Ground Settlement using Machine Learning Techniques," *Frontiers of Structural and Civil Engineering*, vol. 13, no. 6, pp. 1363-1378, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Xinzheng Lu, and Henry Burton, "EESD Special Issue: AI and data-driven Methods in Earthquake Engineering – (Part 1)," *Earthquake Engineering and Structural Dynamics*, vol. 52, no. 8, pp. 2299-2302, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Jungwon Huh et al., "A Probabilistic Fragility Evaluation Method of a RC Box Tunnel Subjected to Earthquake Loadings," *Journal of Korean Tunnelling and Underground Space Association*, vol. 19, no. 2, pp. 143-159, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Zhongkai Huang et al., "Fragility Assessment of Tunnels in Soft Soils using Artificial Neural Networks," *Underground Space*, vol. 7, no. 2, pp. 242-253, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Sotirios Argyroudis et al., "Effects of SSI and Lining Corrosion on the Seismic Vulnerability of Shallow Circular Tunnels," *Soil Dynamics and Earthquake Engineering*, vol. 98, pp. 244-256, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] ASCE/SEI 41-23: *Seismic Evaluation and Retrofit of Existing Buildings*, American Society of Civil Engineers, pp. 1-752, 2023. [[Google Scholar](#)] [[Publisher Link](#)]
- [20] *Seismic Evaluation and Retrofit of Concrete Buildings Volume 1*, Applied Technology Council, Seismic Safety Commission, State of California, 1996. [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Estados Unidos. Federal Emergency Management Agency, *Prestandard and Commentary for the Seismic Rehabilitation of Buildings*, FEMA, pp. 1-400, 2000. [[Google Scholar](#)] [[Publisher Link](#)]
- [22] *Improvement of Nonlinear Static Seismic Analysis Procedures: FEMA-440*, Federal Emergency Management Agency, pp. 1-392, 2005. [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Cristina Cantagallo et al., "Effects of the Extended N2 Method on Non-Linear Static Procedures of Reinforced Concrete Frame Structures," *Soil Dynamics and Earthquake Engineering*, vol. 173, pp. 1-28, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Kevin Karanja Kuria, and Orsolya Katalin Kegyes-Brassai, "Pushover Analysis in Seismic Engineering: A Detailed Chronology and Review of Techniques for Structural Assessment," *Applied Sciences*, vol. 14, no. 1, pp. 1-35, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Maysam Jalilkhani, Seyed Hooman Ghasemi, and Masood Danesh, "A Multi-Mode Adaptive Pushover Analysis Procedure for Estimating the Seismic Demands of RC Moment-Resisting Frames," *Engineering Structures*, vol. 213, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [26] Yuefeng Yang et al., "Numerical Simulation of the Seismic Damage of Daikai Station Based on Pushover Analyses," *Buildings*, vol. 13, no. 3, pp. 1-18, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Chih-Chieh Lu, and Jin-Hung Hwang, "Nonlinear Collapse Simulation of Daikai Subway in the 1995 Kobe Earthquake: Necessity of Dynamic Analysis for a Shallow Tunnel," *Tunnelling and Underground Space Technology*, vol. 87, pp. 78-90, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Tian Tian et al., "Seismic Response of Utility Tunnels with Different Burial Depths at the Non-Homogeneous Liquefiable Site," *Applied Sciences*, vol. 12, no. 22, pp. 1-17, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [29] Zilan Zhong et al., "Experimental Study on Out-of-Plane Seismic Performance of Precast Composite Sidewalls of Utility Tunnel with Grouting-Sleeve Joints," *Underground Space*, vol. 16, pp. 1-17, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [30] Lowell Tan Cabangon, Gaetano Elia, and Mohamed Rouainia, "Modelling the Transverse behaviour of Circular Tunnels in Structured Clayey Soils during Earthquakes," *Acta Geotechnica*, vol. 14, no. 1, pp. 163-178, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [31] Vassilis k. Papanikolaou, and Amr S. Elnashai, "Evaluation of Conventional and Adaptive Pushover Analysis I: Methodology," *Journal of Earthquake Engineering*, vol. 9, no. 6, pp. 923-941, 2005. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [32] Seismosoft, "A Computer Program for Static and Dynamic Nonlinear Analysis of Framed Structures," Seismosoft Ltd., 2004. [[Google Scholar](#)] [[Publisher Link](#)]
- [33] Peter Fajfar, "A Nonlinear Analysis Method for Performance-Based Seismic Design," *Earthquake Spectra*, vol. 16, no. 3, pp. 573-592, 2000. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [34] Anil K. Chopra, and Rakesh K. Goel, "A Modal Pushover Analysis Procedure for Estimating Seismic Demands for Buildings," *Earthquake Engineering and Structural Dynamics*, vol. 31, no. 3, pp. 561-582, 2002. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [35] Helmut Krawinkler, and G.D.P.K. Seneviratna, "Pros and Cons of a Pushover Analysis of Seismic Performance Evaluation," *Engineering Structures*, vol. 20, no. 4-6, pp. 452-464, 1998. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [36] S. Antoniou, and R. Pinho, "Development and Verification of a Displacement-based Adaptive Pushover Procedure," *Journal of Earthquake Engineering*, vol. 8, no. 5, pp. 643-661, 2004. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [37] Maja Kreslin, and Peter Fajfar, "The Extended N2 Method Considering Higher Mode Effects in Both Plan and Elevation," *Bulletin of Earthquake Engineering*, vol. 10, pp. 695-715, 2012. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [38] S.A. Argyroudis, and K.D. Pitilakis, "Seismic Fragility Curves of Shallow Tunnels in Alluvial Deposits," *Soil Dynamics and Earthquake Engineering*, vol. 35, pp. 1-12, 2012. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [39] Zhong-Kai Huang et al., “Seismic Vulnerability of Circular Tunnels in Soft Soil Deposits: The Case of Shanghai Metropolitan System,” *Tunnelling and Underground Space Technology*, vol. 98, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [40] H. Huo et al., “Load Transfer Mechanisms between Underground Structure and Surrounding Ground: Evaluation of the Failure of the Daikai Station,” *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 131, no. 12, pp. 1522-1533, 2005. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]