frontiers
in Built Environment

ORIGINAL RESEARCH
published: 03 May 2021
doi: 10.3389/fbuil.2021.679488

A Monte Carlo Simulation Approach in Non-linear Structural Dynamics Using Convolutional Neural Networks

Franz Bamer†, Denny Thaler*†, Marcus Stoffel and Bernd Markert*

Institute of General Mechanics, RWTH Aachen University, Aachen, Germany

The evaluation of the structural response statistics constitutes one of the principal tasks in engineering. However, in the tail region near structural failure, engineering structures behave highly non-linear, making an analytic or closed form of the response statistics difficult or even impossible. Evaluating a series of computer experiments, the Monte Carlo method has been proven a useful tool to provide an unbiased estimate of the response statistics. Naturally, we want structural failure to happen very rarely. Unfortunately, this leads to a disproportionately high number of Monte Carlo samples to be evaluated to ensure an estimation with high confidence for small probabilities. Thus, in this paper, we present a new Monte Carlo simulation method enhanced by a convolutional neural network. The sample-set used for this Monte Carlo approach is provided by artificially generating site-dependent ground motion time histories using a non-linear Kanai-Tajimi filter. Compared to several state-of-the-art studies, the convolutional neural network learns to extract the relevant input features and the structural response behavior autonomously from the entire time histories instead of learning from a set of hand-chosen intensity inputs. Training the neural network based on a chosen input sample set develops a meta-model that is then used as a meta-model to predict the response of the total Monte Carlo sample set. This paper presents two convolutional neural network-enhanced strategies that allow for a practical design approach of ground motion excited structures. The first strategy enables for an accurate response prediction around the mean of the distribution. It is, therefore, useful regarding structural serviceability. The second strategy enables for an accurate prediction around the tail end of the distribution. It is, therefore, beneficial for the prediction of the probability of failure.

Keywords: Monte Carlo method, non-linear structural mechanics, elastoplastic structure, convolutional neural networks, machine learning, earthquake engineering, probability of failure

OPEN ACCESS

Edited by:
Xinzheng Lu,
Tsinghua University, China

Reviewed by:
Kaiqi Lin,
Fuzhou University, China
Kaiming Bi,
Curtin University, Australia

***Correspondence:**
Franz Bamer
bamer@iam.rwth-aachen.de
Denny Thaler
thaler@iam.rwth-aachen.de

†These authors have contributed equally to this work and share first authorship

Specialty section:
This article was submitted to Earthquake Engineering, a section of the journal Frontiers in Built Environment

Received: *11 March 2021*
Accepted: *30 March 2021*
Published: *03 May 2021*

Citation:
Bamer F, Thaler D, Stoffel M and Markert B (2021) A Monte Carlo Simulation Approach in Non-linear Structural Dynamics Using Convolutional Neural Networks. Front. Built Environ. 7:679488. doi: 10.3389/fbuil.2021.679488

1. INTRODUCTION

Deciding whether a structure subjected to ground motion is or is not safe is a delicate task in engineering. One big issue is the high level of uncertainty when considering a ground excitation time history relevant to structural design. This fact forces us to apply probabilistic methodologies to design structures and infrastructures (Der Kiureghian, 1996; Bucher, 2009).

Failure usually occurs in the non-linear range of structural behavior, making an analytic or closed form of the response statistics difficult or even impossible. In this context, the equivalent linearization method was proposed to approximate the response statistics, which is generally quite accurate around the mean of the distribution (Roberts and Spanos, 1990). To also provide accurate response statistics for infrequent events, the tail equivalent linearization method was proposed (Fujimura and Der Kiureghian, 2007).

The general Monte Carlo method is helpful to provide the response statistics of non-linear systems. The strategy is based on a series of computer experiments and provides an unbiased estimation for the probability of failure of a non-linear system (Hurtado and Barbat, 1998).

However, considering that each computer experiment includes a time history analysis of a complex high-dimensional finite element model, the Monte Carlo method turns out to be computationally expensive. As engineers, we have to ensure that structural failure should happen extremely rare, as it is generally accompanied by severe societal and economic loss. Thus, a disproportionately high number of computer experiments must be performed to estimate the occurrence of low-probability events with high confidence. This fact makes the application of the crude Monte Carlo method unfeasible for non-linear high-dimensional problems. A promising strategy to overcome this issue is to use model order reduction methods (Bamer and Bucher, 2012; Bamer and Markert, 2017; Bamer et al., 2018). This strategy enables one to decrease the computational burden for each sample computation significantly, and it is particularly efficient if only one reduced-order basis is applied during the whole Monte Carlo simulation run (Bamer et al., 2017).

In the context of computational speed-up and meta-modeling, the application of neural networks has proven to be a promising strategy (Stoffel et al., 2018, 2019, 2020a,b) for problems in non-linear structural mechanics. In particular, the incorporation of recurrent neural network architectures for elastoplastic problems leads to representations of the non-linear structural behavior revealing accurate and efficient solution functions (Koeppe et al., 2019, 2020).

With special emphasis on problems in earthquake engineering, Sun et al. (2021) has summarized the incorporation of machine learning approaches to response, damage, and failure prediction. Convolutional neural networks and deep learning have been used to predict the whole response time history of the structure subjected to a transient excitation (Zhang et al., 2020) and wavelet transformation-based response prediction (Lu et al., 2020, 2021; Liao et al., 2021). Thaler et al. (2021a,b) proposed a machine leaning-enhanced Monte Carlo method using a simple feed-forward neural network. In doing so, they used a selected choice of intensity measures from the time history data as a set of input parameters to predict the response of the structure. However, representing the whole time history data by a few hand-chosen scalar values only, much information regarding the excitation function can get lost.

In this paper, we present an extended Monte Carlo approach using convolutional neural networks so that it is not necessary to extract hand-designed input features, e.g., intensity measures, for the neural network. Furthermore, in order to assure high reliability of the neural network predictions in the tail of the distribution, an extended training strategy is applied.

The structure of this paper is goal-oriented and straightforward. In section 2, we firstly present an overview of convolutional neural networks followed by the presentation of the new Monte Carlo simulation strategy. Subsequently, in section 3, we present the new strategy using an illustrative numerical example. The efficiency, as well as advantages and disadvantages, are discussed. Finally, in section 4, the conclusion is drawn.

2. METHODS

2.1. Feedforward and Convolutional Neural Network Architectures in a Nutshell

Machine learning strategies have been applied to a wide range of tasks in engineering. This section briefly introduces the basic theory of the two machine learning methods applied in this paper, i.e., feedforward and convolutional neural networks.

The feedforward neural network consists of at least three layers: input, hidden, and output layer. Every layer contains a set of neurons. The input layer receives the initial data. The data is passed forward, layer by layer, until the output layer is reached from this layer. Every neuron of a certain layer, which is not the output layer, is connected to every other neuron of the next layer during one forward step. As a result of this, each neuron-to-neuron link has its corresponding weight value. A weight matrix can then represent the set of all connections. Thus, the output of the previous layer $\mathbf{y}_{i-1}$ is the input of the actual layer $\mathbf{x}_i$. The input is then multiplied by the weight matrix $\mathbf{W}_i$, and additionally, summed up by a bias vector $\mathbf{b}_i$. The sum is transferred by an activation function f_{act} which is then the output $\mathbf{y}_i$ of a layer:

$$\mathbf{y}_i = f_{act}(\mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i) \, . \tag{1}$$

This operation is repeated until the output layer, which constitutes the prediction of the neural network. In our paper, we apply supervised learning, where the prediction is compared to a known target output for regression tasks. The error is used to optimize the weight matrices and the bias of all the layers using back-propagation.

The convolutional neural network is a specific type of architecture that is used to process data. It is specialized for input data that show a correlation of neighboring data points. Thus, typical applications of convolutional neural networks are image data. However, as presented during this paper, convolutional neural networks can also effectively be applied to extract the main features of earthquake time histories. Using convolutional layers improves the machine learning system by sparse interaction and parameter sharing. Consequently, a reduction of the total number of trainable parameters, and therefore, decreased computational effort. Convolutional neural networks are composed of two main operations: convolution and pooling.

The convolution is the basic mathematical operation of this neural network type. Regarding one dimensional discrete data, such as earthquake time histories for example, the convolution operation $< * >$ of two vectors or one-dimensional tensors, $\mathbf{g}$ and $\mathbf{h}$, is written as:

$$(\mathbf{g} * \mathbf{h})_i = \sum_n g_{i-n} h_n \, . \tag{2}$$

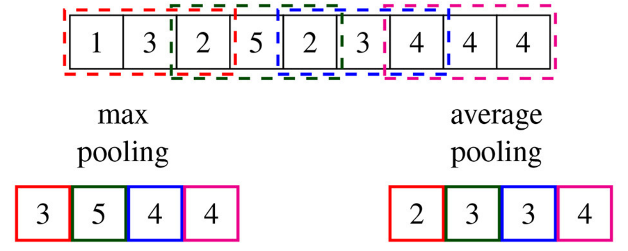

FIGURE 1 | Exemplary spatial reduction of the top data array using pooling functions (pool size = 3, stride size = 2); output data reduced by maximum pooling (bottom left); output data reduced by average pooling (bottom right).

The convolutional operation is usually followed by a non-linear activation function f_{act}. The output $\mathbf{y}_i$ of the convolution operation, the so-called feature map, is written as:

$$\mathbf{y}_i = f_{act}(\mathbf{x}_i * \mathbf{w}_i + \mathbf{b}_i) \ . \tag{3}$$

In this equation, $\mathbf{w}_i$ denotes a weight vector with the spatial size k of the kernel, which slides over the one-dimensional input data array $\mathbf{x}_i$.

In the next step, pooling improves the statistical efficiency by reducing the spatial size of the data. As shown in **Figure 1**, pooling summarizes the input space. Looking at this illustrative example, one can see that the pooling procedure summarizes three values to one output value. As a result of this, the decrease in the total number of values, i.e., the compression of information, is dependent on the stride size, which is the step increment at which the kernel moves over the data array. For the example in **Figure 1**, the stride size is two. Two different types of pooling procedures are considered in this paper: max-pooling and average-pooling. As expected from the nomenclature, max-pooling extracts the maximum of all values within the moving kernel, and average pooling extracts the average of the values within the moving kernel. Even though convolutions and pooling are individual operations, it is often referred to as a convolutional layer, whereby the operations are carried out in different stages.

Similar to the weight optimization procedure of fully connected layers, the values of the convolutional kernels are optimized during training, which allows for the extraction of features of the input data. Different features can be extracted from the initial data by using multiple kernels within the convolutional layer. A convolutional neural network usually consists of several convolutional layers, followed by feedforward layers. The choice of the architecture depends on the complexity of the data and the task to solve (Kruse et al., 2015; Goodfellow et al., 2016).

2.2. Machine Learning Enhanced Monte Carlo Simulation

The probability of structural failure P_f can be expressed in terms of a multiple integral:

$$P_f = \int \cdots \int_{g(\mathbf{x})<0} f_X(\mathbf{x}) \, \mathrm{d}x_1 \ldots \mathrm{d}x_n \ , \tag{4}$$

where $g(x)$ is the limit state function that divides the entire domain into the save region $g(x) \geq 0$ and the failure region $g(x) < 0$. Dependent on the complexity of the limit state function, the solution of this integral is generally not straightforward. However, one approach to solve this integral is provided by the Monte Carlo simulation. One performs a series of computer experiments by artificially generating a set of random inputs (Hurtado and Barbat, 1998). The integral describing the probability of failure in Equation (4) is rewritten as:

$$p_f = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} I_g(x_1, \ldots, x_n) \\ f_{X_1,\ldots,X_n}(x_1, \ldots, x_n) \, \mathrm{d}x_1 \ldots \mathrm{d}x_n \ , \tag{5}$$

with $I_g(x_1, \ldots, x_n)$ being an indicator function defining a safe or failure state:

$$I_g(x_1, \ldots, x_n) = 1 \quad \text{if} \quad g(\mathbf{x}) \leq 0 \quad \text{(structural failure)} \tag{6}$$

$$I_g(\mathbf{x}) = 0 \quad \text{if} \quad g(\mathbf{x}) > 0 \quad \text{(save)} \ . \tag{7}$$

Thus, the probability of failure is provided by a consistent unbiased estimate in terms of an expected value:

$$p_f = \frac{1}{m} \sum_{k=1}^{m} I_g(\mathbf{x}^{(k)}) \ . \tag{8}$$

The standard deviation of the estimation of the probability of failure is evaluated as:

$$\sigma_{p_f}^2 = \frac{p_f}{m} - \frac{p_f^2}{m} \approx \frac{p_f}{m} \Rightarrow \sigma_{p_f} = \sqrt{\frac{p_f}{m}} \ , \tag{9}$$

which means that the variance of the probability of failure increases with a decreasing number of samples m. In engineering, we obviously want the probability of failure to remain a low value. Thus, the reliability of the estimate of a small probability of failure is considerably low using the Monte Carlo method and m must be chosen to be a disproportionately high number. In other words, extensive relevant earthquake record data must be collected in order to reliably design structures. A rule of thumb may be that a number of 10^5 excitation records should be considered for structural design with a desired failure probability of 10^{-4}—obviously an impossible task, as the number of recorded relevant ground excitations around the respective building site is inherently limited. Therefore, we suggest to artificially generate a sufficiently high number of random ground excitations using the metadata from one relevant record. In doing so, site-dependent structural design is possible. In order to realize artificial site-dependency, we consider the incorporation of a non-linear Kanai-Tajimi filter (Bamer and Markert, 2017). To extract the important information from a ground excitation record, a time window of size t_w is defined that moves over the excitation time history, as shown in **Figure 2**. For the numerical example in this paper, a severe earthquake, recorded in Kobe Takarazuka in 1995 (Kobe, 2016), is chosen as benchmark time history.

In this paper, we extract two time-dependent identification parameters, which are the intensity $\hat{e}(t)$ and the number of zero

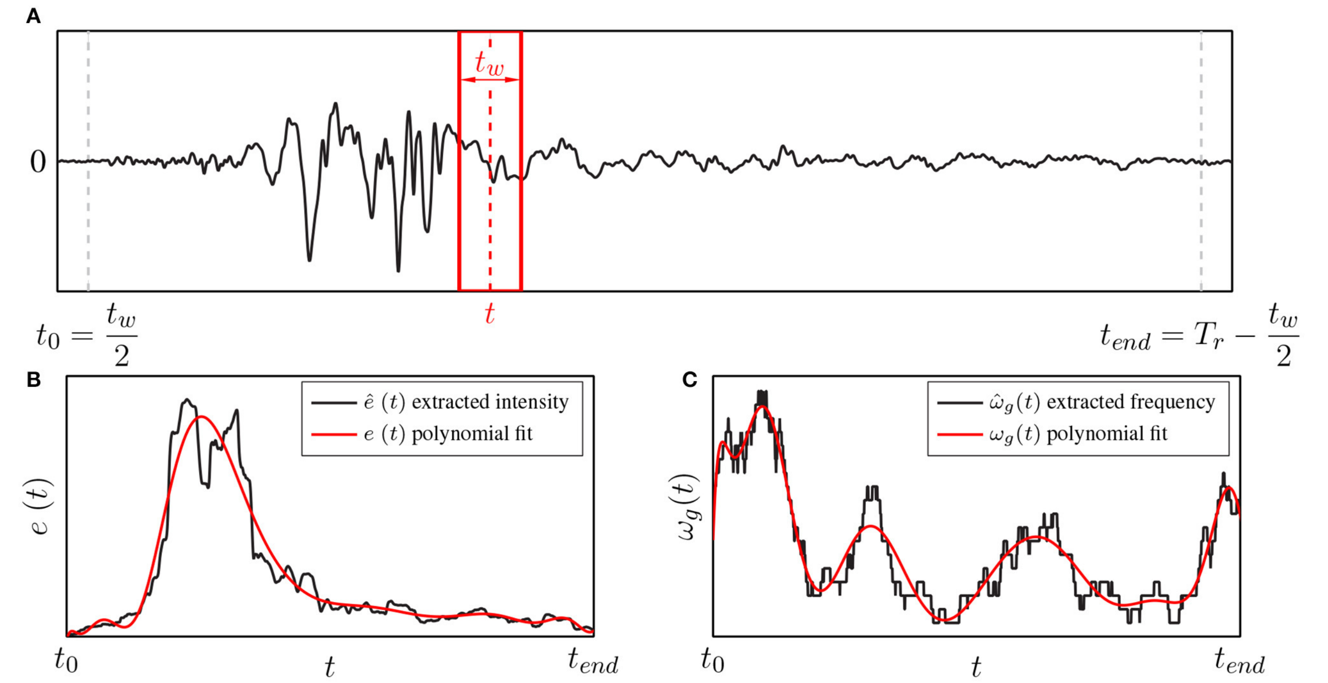

FIGURE 2 | (A) Moving time window to extract relevant metadata from acceleration records; **(B)** extracted intensity and **(C)** extracted lowest frequency content.

crossings $\hat{n}(t)$ of the time history, as shown in **Figure 2**. The first parameter is evaluated by the integration of the squared ground acceleration $\ddot{x}_r$ of the benchmark record from $t - \frac{t_w}{2}$ to $t + \frac{t_w}{2}$:

$$\hat{e}(t) = \int_{t-\frac{t_w}{2}}^{t+\frac{t_w}{2}} (\ddot{x}_r)^2 \ \mathrm{d}t \ . \tag{10}$$

The second parameter leads to the time history of the lowest ground frequency of the benchmark record. Equivalently to the procedure in Equation (10), the ground frequency $\hat{\omega}_g(t)$ is evaluated by considering the number of zero-crossings $\hat{n}(t)$ in the time period $\left[t - \frac{t_w}{2}, t + \frac{t_w}{2}\right]$:

$$\hat{\omega}_g(t) = \frac{\hat{n}(t)}{t_w} 2\pi \ . \tag{11}$$

Polynomial functions approximate the extracted data, which results in the representations of the intensity $e(t)$ and the frequency content $\omega(t)$, highlighted in red in the respective plot of **Figure 2**.

The non-linear Kanai-Tajimi filter is used to model the movement of the ground. In this context, the filter is represented by a non-linear single degree of freedom system that is subjected to a stationary Gaussian random white noise $w(t)$:

$$\ddot{x}_f + 2\zeta_g \omega_g(t)\dot{x}_f + \omega_g^2(t) x_f = w(t) \ . \tag{12}$$

The damping parameter ζ_g and the time-dependent frequency $\omega_g(t)$ are both related to site-dependent ground properties. The response of the filter x_f is evaluated using numeric integration, the filtered ground acceleration is then written as:

$$\ddot{\tilde{x}}_g = -2\zeta_g \omega_g(t)\dot{x}_f - \omega_g^2(t) x_f \ . \tag{13}$$

In order to consider the extracted intensity, the obtained filter acceleration $\ddot{\tilde{x}}_g$ is finally multiplied by the intensity polynomial $e(t)$:

$$\ddot{x}_g = \ddot{\tilde{x}}_g \, e(t) \ . \tag{14}$$

The result of this procedure, using our benchmark ground acceleration, is depicted in **Figure 3**. Here the benchmark excitation, recorded in Kobe, is depicted in **Figure 3A** and one respective sample excitation is presented in **Figure 3B**. One can observe the similarity regarding both frequency and intensity time histories while preserving the required level of randomness required for the Monte Carlo simulation.

Theoretically, the procedure is straightforward, as shown in **Figure 4**. One creates a set of ground excitation time histories $\mathcal{S}$ based on which the response set $\mathcal{R}$ is evaluated. Considering the limit state function $g(\mathbf{x})$, every response function leads then to its corresponding indicator function value $\mathcal{I}$:

$$\begin{aligned} \mathcal{S} = &< \ddot{x}_g^{(1)}, \cdots, \ddot{x}_g^{(m)} > \longrightarrow \mathcal{R} < \mathbf{x}^{(1)}(t), \cdots, \\ &\mathbf{x}^{(m)}(t) > \longrightarrow \mathcal{I} < I_g(\mathbf{x}^{(1)}), \cdots, I_g(\mathbf{x}^{(m)}) > \end{aligned} \tag{15}$$

The evaluation of the response set $\mathcal{R}$ is realized by numeric integration. In this paper, the Newmark method is used to evaluate the response function sample-by-sample (Bamer et al.,

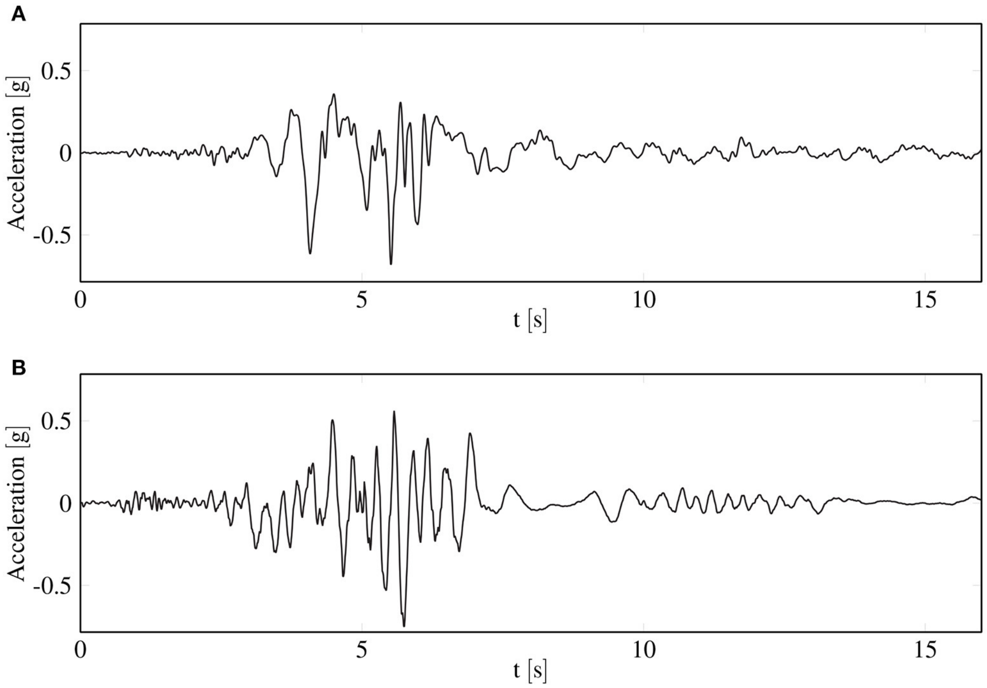

FIGURE 3 | Ground acceleration time histories; **(A)** real acceleration record measured in Kobe (2016) as the benchmark earthquake; **(B)** artificially generated earthquake that inherits the properties from the benchmark earthquake.

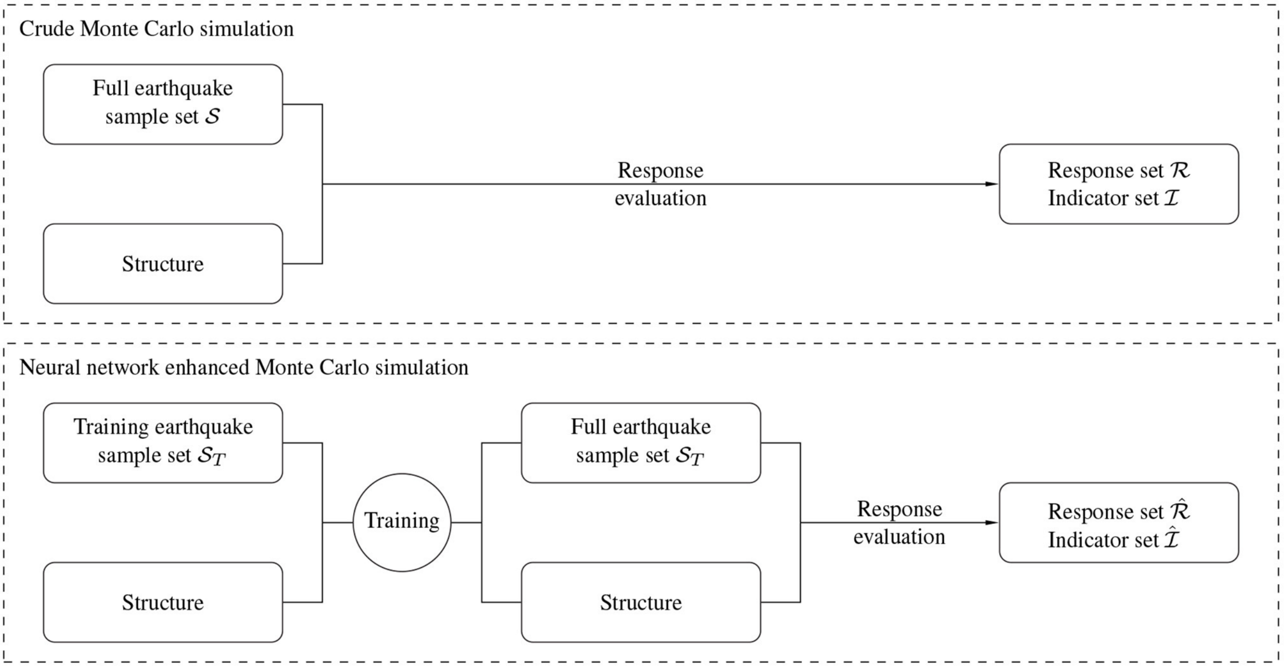
Crude Monte Carlo simulation
Full earthquake sample set $\mathcal{S}$
Structure
Response evaluation
Response set $\mathcal{R}$ Indicator set $\mathcal{I}$
Neural network enhanced Monte Carlo simulation
Training earthquake sample set $\mathcal{S}_T$
Structure
Training
Full earthquake sample set $\mathcal{S}_T$
Structure
Response evaluation
Response set $\hat{\mathcal{R}}$ Indicator set $\hat{\mathcal{I}}$

FIGURE 4 | (Top) Crude Monte Carlo simulation workflow; **(Bottom)** Machine-learning-enhanced Monte Carlo simulation workflow.

2021). However, due to relation (9), the number of samples m must be chosen disproportionately high to obtain a reliable response statistics for small probabilities of structural failure. Therefore, a realization of the crude Monte Carlo method becomes unfeasible if the system is high-dimensional and complex. In order to reliably estimate small probabilities of structural failure, we propose a strategy where a neural network is trained to learn the response behavior using a smaller training sample subset $\mathcal{S}_T \subset \mathcal{S}$, as shown in the bottom plot of **Figure 4**. Subsequently, the neural network can predict the full response and indicator sample sets, $\mathcal{R}$ and $\mathcal{I}$. One significantly benefits from the superior computational efficiency of the neural network meta-model, as it practically constitutes a series of subsequent multiplications. This strategy has one downside: the occurrence of extreme events in the sample subset $\mathcal{S}_T$ is unlikely if the number of training subset samples is considerably smaller than the number of total samples. Neural networks will then not accurately predict the most relevant extreme events for a reliable estimation of structural failure. One measure to avoid this problem is to significantly increase the number of training subset samples, which would significantly decrease the numerical efficiency of the proposed strategy. Therefore, we pursue a different approach: we extend the training sample subset $\mathcal{S}_T$ by introducing an additional random parameter α for the extracted intensity $\ddot{x}_r$, presented in Equation (10). The extension is written as:

$$\hat{e}(t) = \int_{t-\frac{t_w}{2}}^{t+\frac{t_w}{2}} (\alpha \ddot{x}_r)^2 \, \mathrm{d}t \,. \tag{16}$$

As shown later in the results, a unified distribution for α between 0.8 and 1.5 is a good choice for the problems in this paper. Using this extension allows us to train the neural network with a significantly higher portion of extreme events in the tail end of the distribution. The neural network is then later able to predict such relevant extreme events accurately. Instead of using a few hand-designed intensity parameters (Morfidis and Kostinakis, 2018), we use the whole excitation history as input for the neural network, as shown at the left side of **Figure 5**. As a result, any loss of information that could influence the structural response is avoided. In this paper, the output quantity of interest is chosen as the peak story drift ratio (PSDR) of the ground floor. However, if necessary, one can adopt the output quantity of interest. A convolutional neural network architecture is chosen for the enhanced Monte Carlo simulation method. The convolutional layers allow for automatic extraction of the time-dependent relevant patterns of the excitation time input and a breakdown of the time history information to flattened characterization parameters (see **Figure 5**). The subsequent fully-connected layers lead then to one output quantity of interest—the PSDR.

3. NUMERICAL RESULTS

The numerical demonstration of the new strategies is presented on a three-story two-bay frame structure subjected to the set of ground motions, as depicted in **Figure 6**. One ground motion sample is presented in **Figure 6A** and the structure is presented in **Figure 6B**. For the columns, beam elements with a squared hollow cross section (0.3 × 0.3 m, thickness 0.03 m) are chosen. For the beams, beam elements with a rectangular cross section (0.3 × 0.4 m, thickness 0.03 m) are chosen. Every column and beam is discretized by four fiber beam elements leading to a total number of 188 degrees of freedom (Bamer and Bucher, 2012; Bamer and Markert, 2017; Bamer et al., 2017). An elastoplastic material law with kinematic hardening is considered using the following material parameters: initial Young's modulus ... $2.1 \times 10^{11}\,\mathrm{N\,m^{-2}}$, hardening stiffness ... $2.1 \times 10^{10}\,\mathrm{N\,m^{-2}}$, yielding stiffness ... $2.4 \times 10^{8}\,\mathrm{N\,m^{-2}}$, density ... $7.850\,\mathrm{kg\,m^{-3}}$. Additionally, point masses of a value of 5.000 kg are added to every finite element node belonging to the beams of the frame structure in order to roughly consider the structural setting. We used our python and C++ based in-house tool to perform the numerical calculations.

Using the Newmark algorithm for numeric integration, the response of the structure to the generated ground excitation (**Figure 6A**) is evaluated. The absolute displacement of the left corners of all three stories is depicted in **Figure 6C**, based on which the PSDR is extracted. One can see that the structure experiences plastic deformation due to this sample excitation, as a plastic drift-off remains after the whole integration time period. We confirmed the level of plasticity by observing the material hystereses on the Gauss-Integration point level. Plasticity mainly occurs around the frame corners of the first floors. For this example, the major displacement always occurs on the first floor, leading to the decision to take the PSDR of the first floor as the design quantity of interest.

We performed a rigorous hyperparameter search to find a neural network setup that leads to predictions of a high level of accuracy. In doing so, we found that an optimum architecture is dependent on the number of samples. If the chosen neural network architecture is chosen complex enough to extract all relevant features, a larger number of samples leads to higher accuracy of the neural network predictions. Concerning the efficiency of the proposed approach, we decided to keep the number of samples small. Therefore, the convolutional neural network architecture can also be kept simple to extract only the most relevant features. The chosen neural network architecture receives the sample excitation time history as input. Subsequently, three convolutional layers are implemented, followed by two fully connected layers, which lead to the PSDR prediction. Each convolutional layer consists of eight filters. As a result, a kernel size of 32 and a stride size of 3 have been found useful for the first convolutional layer. Regarding the second convolutional layer, the kernel size reduces to 16 with a stride size of 2. The last convolutional layer has a kernel size of 8 and a stride size of 1. After each convolutional stage, max-pooling summarizes the outputs using a pooling size of 2 and the same stride size. After passing the filters of the last convolutional layer, the outputs are averaged before the values are flattened and forwarded to the fully connected layers. The two fully connected layers consist of ten neurons each. The rectified linear activation function is applied to the convolutional stages and the fully connected layers. This architecture has shown the best accuracy if the number of samples within the sample subset of $\mathcal{S}_\mathcal{T}$ is 5×10^3. Using the standard training strategy, the mean absolute error of a validation set using 10^3 samples shrinks down to 1.9×10^{-3} after 200 epochs of training. Using the extended set for the training procedure, the mean absolute error after 200 training epochs is slightly higher with a value of 2.1×10^{-3}. During the hyperparameter search, the number of convolutional

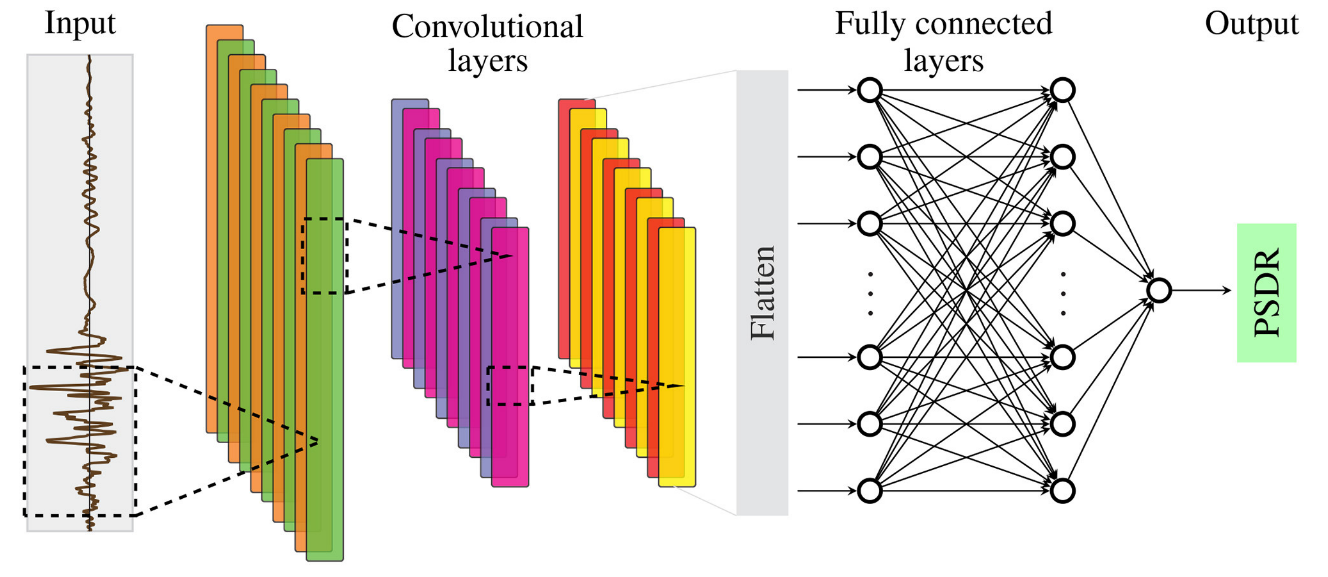

FIGURE 5 | Convolutional neural network architecture for the enhanced Monte Carlo simulation method.

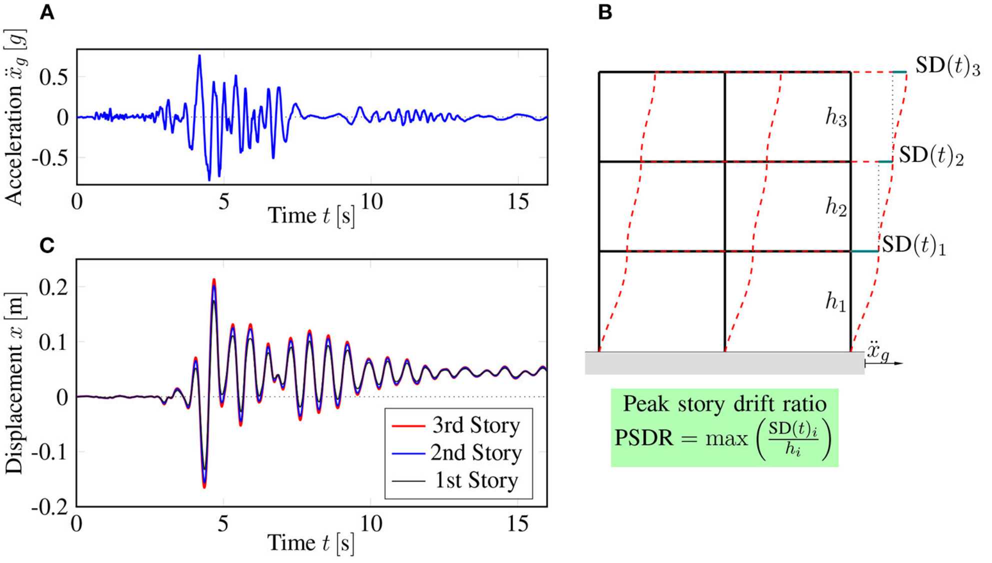

FIGURE 6 | Structural response of a frame structure subjected to ground excitation; **(A)** time history of one artificially generated earthquake; **(B)** illustrative frame structure used for the numerical demonstration and visualization of the story drifts (SD); **(C)** displacement time history of the left corner of every story of the frame structure.

layers was varied between one and six using up to 32 filters with kernel sizes between 3 and 64. Furthermore, the pooling sizes within the convolutional layers were varied, and both previously presented pooling functions were applied. A more extensive hyperparameter search might result in better prediction accuracy of the neural network, with high probability, if using an increased number of training samples. However, this convolutional neural network architecture can predict the PSDR accurately enough for this study. We used the Python-based library TensorFlow to implement the neural network meta-model.

The Monte Carlo simulation results are depicted in **Figure 7**. Within the subplots of this figure, the crude Monte Carlo simulation using 10^5 samples is depicted by the gray dotted lines, the enhanced Monte Carlo simulation is depicted as the red lines, and the green lines depict the extended enhanced Monte Carlo simulation. We performed the numerical example

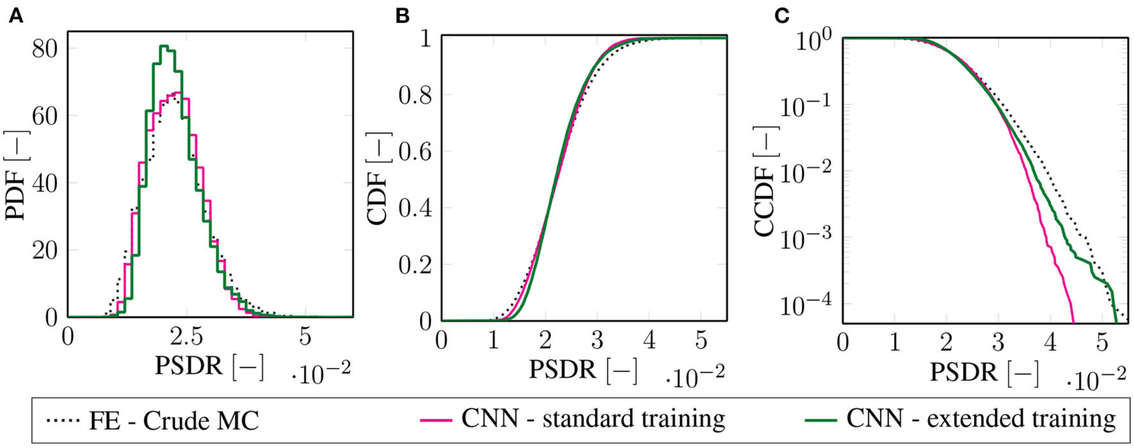

FIGURE 7 | (A) Probability density function (PDF), **(B)** cumulative distribution function (CDF), and **(C)** logarithmic plot of the complementary cumulative distribution function applying the crude Monte Carlo method (black, dotted line), the neural network-enhanced Monte Carlo method using the standard training strategy (red line), and the neural network-enhanced Monte Carlo method using the extended training strategy (green line).

on a computer with an Intel Xeon E5-2643 processor running at 3.3 GHz. The operating system is Linux Ubuntu 16.04 with 60 GB of working memory available, which is by no means required for the simulation. For the example in this paper, the crude Monte Carlo simulation requires a calculation time of ∼1 week, while the prediction procedure of the two neural network-enhanced strategies requires only a few seconds. However, to provide a fair speed-up measure, one must consider evaluating the training sample set, the hyperparameter fitting, and the training procedure. Considering those factors, we chose a fair speed-up measure for the new strategies by dividing the number of full samples by the number of training samples. The numerical demonstration results in a value of 10^5 samples for the full set and over 5×10^3 samples for the training set, which results in a speed-up factor of 20.

Figure 7A presents the probability density function (PDF) in terms of a histogram over the displacement intervals. At first sight, one can see that the enhanced Monte Carlo simulation method using the extended scheme is not as accurate as the enhanced Monte Carlo using the standard training scheme. Also, regarding the cumulative distribution function (CDF) prediction, shown in **Figure 7B**, the enhanced Monte Carlo simulation scheme using the standard training algorithm seems to be more accurate than the enhanced Monte Carlo method using the extended training scheme. This is not surprising as the neural network is better trained around the mean using standard sample subset $\mathcal{S}_T$ then using the extended subset $\hat{\mathcal{S}}_T$. However, accuracy around the mean is not that relevant in structural engineering design, as, for obvious reasons, we want structural failure to happen very rarely. Therefore, we present the tail end of the complementary cumulative distribution function (CCDF) in **Figure 7C**, which shows the occurrence of exceeding a certain PSDR in a logarithmic plot. Here, the clear advantage of the enhanced Monte Carlo simulation using an extended sample subset becomes clear, as it outperforms the standard, enhanced Monte Carlo simulation in the crucial region around the tail end of the distribution. Evaluating the probability of exceedance for a PSDR over 3.5×10^{-2} m results in a reliable approximation using the extended neural network enhanced strategy, while the neural network enhanced method using the standard training becomes gradually more inaccurate until the method entirely fails for predicting the probability of a PSDR exceeding a value of 4.2×10^{-2} m.

4. CONCLUSION

In this paper, we proposed a new Monte Carlo simulation method enhanced by a convolutional neural network. Using a non-linear Kanai-Tajimi filter, site-dependent ground conditions were taken into account. Two convolutional neural network-enhanced strategies were proposed. The first strategy uses a standard training procedure by considering samples from the full excitation sample set. The second strategy uses a sample set with increased variance regarding intensity to train the neural network more around the tail end of the distribution. In the presented numerical example, the PSDR is the output quantity. However, the proposed strategy can be adapted to predict other output features. The investigation on a more sophisticated damage prediction using multiple output quantities is of high interest for future research.

We have shown that both strategies reveal outstanding efficiency when compared to the crude Monte Carlo simulation. Structural design that incorporates non-linear behavior has practically not been realized using the Monte Carlo method as it was yet not possible in a feasible amount of time. However, the new strategies allow for Monte Carlo simulations in a feasible amount of time that can now be incorporated into a practical design environment. The accuracy of the convolutional neural network is dependent on the number of training samples. Accordingly, one always makes a trade-off between the accuracy of the predictions and the computing time for evaluating the training samples.

The two types of convolutional neural network enhanced Monte Carlo simulation strategies have different advantages and disadvantages. The first type of the new method applying the standard training strategy shows a high accuracy around the mean of the distribution. One can conclude that they will be more useful for estimations of the response regarding structural serviceability. The second type of the new method applying the extended training strategy reveals to be rather more inaccurate around the mean of the distribution than the first type. However, it reveals an outstanding accuracy around the tail end of the distribution, where the first method fails. It is, therefore, highly appropriate to be applied to practical engineering procedures that involve the design of structures considering the low target probability of failure that is, especially, required for engineering structures in an urban built environment.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

FB and DT developed the new method, performed the numerical simulations, and wrote the manuscript. MS and BM reviewed the manuscript and supervised the research project. All authors contributed to the article and approved the submitted version.

REFERENCES

Bamer, F., Amiri, A., and Bucher, C. (2017). A new model order reduction strategy adapted to nonlinear problems in earthquake engineering. *Earthq. Eng. Struct. Dyn*. 46, 537–559. doi: 10.1002/eqe.2802

Bamer, F., and Bucher, C. (2012). Application of the proper orthogonal decomposition for linear and nonlinear structures under transient excitations. *Acta Mech*. 223, 2549–2563. doi: 10.1007/s00707-012-0726-9

Bamer, F., and Markert, B. (2017). An efficient response identification strategy for nonlinear structures subject to nonstationary generated seismic excitations. *Mech. Based Des. Struct. Mach*. 45, 313–330. doi: 10.1080/15397734.2017.1317269

Bamer, F., Shi, J., and Markert, B. (2018). Efficient solution of the multiple seismic pounding problem using hierarchical substructure techniques. *Comput. Mech*. 62, 761–782. doi: 10.1007/s00466-017-1525-x

Bamer, F., Shirafkan, N., Xiaodan, C., Oueslati, A., Stoffel, M., de Saxcé, G., et al. (2021). A newmark space-time formulation in structural dynamics. *Comput. Mech*. 1–19. doi: 10.1007/s00466-021-01989-4

Bucher, C. (2009). *Computational Analysis of Randomness in Structural Mechanics*, Vol. 3. London: Tayler & Francis.

Der Kiureghian, A. (1996). Structural reliability methods for seismic safety assessment: a review. *Eng. Struct*. 18, 412–424. doi: 10.1016/0141-0296(95)00005-4

Fujimura, K., and Der Kiureghian, A. (2007). Tail-equivalent linearization method for nonlinear random vibration. *Probabil. Eng. Mech*. 22, 63–76. doi: 10.1016/j.probengmech.2006.08.001

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press. Available online at: http://www.deeplearningbook.org

Hurtado, J., and Barbat, H. (1998). Monte Carlo techniques in computational stochastic mechanics. *Archiv. Comput. Methods Eng*. 5, 3–30. doi: 10.1007/BF02736747

Kobe (2016). *Kobe Takarzuka Earthquake Measurement 1995–01-16 20:46:52 UTC*. Center for Engineering Strong Motion Data. Virtual Data Center. Available online at: www.strongmotioncenter.org/

Koeppe, A., Bamer, F., and Markert, B. (2019). An efficient Monte Carlo strategy for elasto-plastic structures based on recurrent neural networks. *Acta Mech*. 230, 3279–3293. doi: 10.1007/s00707-019-02436-5

Koeppe, A., Bamer, F., and Markert, B. (2020). An intelligent nonlinear meta element for elastoplastic continua: deep learning using a new time-distributed residual U-Net architecture. *Comput. Methods Appl. Mech. Eng*. 366:113088. doi: 10.1016/j.cma.2020.113088

Kruse, R., Borgelt, C., Braune, C., Klawonn, F., Moewes, C., and Steinbrecher, M. (2015). *Computational Intelligence, Eine Methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. Wiesbaden: Springer Vieweg.

Liao, W., Chen, X., Lu, X., Huang, Y., and Tian, Y. (2021). Deep transfer learning and time-frequency characteristics-based identification method for structural seismic response. *Front. Built Environ*. 7:10. doi: 10.3389/fbuil.2021.627058

Lu, X., Liao, W., Huang, W., Xu, Y., and Chen, X. (2020). An improved linear quadratic regulator control method through convolutional neural network–based vibration identification. *J. Vib. Control* 27:107754632093375. doi: 10.1177/1077546320933756

Lu, X., Xu, Y., Tian, Y., Cetiner, B., and Taciroglu, E. (2021). A deep learning approach to rapid regional post-event seismic damage assessment using time–frequency distributions of ground motions. *Earthq. Eng. Struct. Dyn*. 50, 1612–1627. doi: 10.1002/eqe.3415

Morfidis, K., and Kostinakis, K. (2018). Approaches to the rapid seismic damage prediction of R/C buildings using artificial neural networks. *Eng. Struct*. 165, 120–141. doi: 10.1016/j.engstruct.2018.03.028

Roberts, J., and Spanos, P. (1990). *Random Vibration and Statistical Linearization*. New York, NY: John Wiley & Sons.

Stoffel, M., Bamer, F., and Markert, B. (2018). Artificial neural networks and intelligent finite elements in non-linear structural mechanics. *Thin Walled Struct*. 131, 102–106. doi: 10.1016/j.tws.2018.06.035

Stoffel, M., Bamer, F., and Markert, B. (2019). Neural network based constitutive modeling of nonlinear viscoplastic structural response. *Mech. Res. Commun*. 95, 85–88. doi: 10.1016/j.mechrescom.2019.01.004

Stoffel, M., Bamer, F., and Markert, B. (2020a). Deep convolutional neural networks in structural dynamics under consideration of viscoplastic material behaviour. *Mech. Res. Commun*. 108:103565. doi: 10.1016/j.mechrescom.2020.103565

Stoffel, M., Gulakala, R., Bamer, F., and Markert, B. (2020b). Artificial neural networks in structural dynamics: a new modular radial basis function approach vs. convolutional and feedforward topologies. *Comput. Methods Appl. Mech. Eng*. 364:112989. doi: 10.1016/j.cma.2020.112989

Sun, H., Burton, H. V., and Huang, H. (2021). Machine learning applications for building structural design and performance assessment: state-of-the-art review. *J. Build. Eng*. 33:101816. doi: 10.1016/j.jobe.2020.101816

Thaler, D., Bamer, F., and Markert, B. (2021a). A machine learning enhanced structural response prediction strategy due to seismic excitation. *PAMM* 20:e202000294. doi: 10.1002/pamm.202000294

Thaler, D., Stoffel, M., Markert, B., and Bamer, F. (2021b). Machine learning enhanced tail end prediction of structural response statistics in earthquake engineering. *Earthq. Eng. Struct. Dyn*. eqe3432. doi: 10.1002/eqe.3432

Zhang, R., Liu, Y., and Sun, H. (2020). Physics-guided convolutional neural network (PHYCNN) for data-driven seismic response modeling. *Eng. Struct.* 215:110704. doi: 10.1016/j.engstruct.2020.110704

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.