

Article

Interpretable Machine Learning for Assessing the Cumulative Damage of a Reinforced Concrete Frame Induced by Seismic Sequences

Petros C. Lazaridis ^{1,*}, Ioannis E. Kavvadias ¹, Konstantinos Demertzis ², Lazaros Iliadis ¹, and Lazaros K. Vasiliadis ¹

¹ Department of Civil Engineering, Democritus University of Thrace, Campus of Kimméria, 67100 Xanthi, Greece

² Department of Forestry and Management of the Environment and Natural Resources, Democritus University of Thrace, 25520 Orestiada, Greece

* Correspondence: plazarid@civil.duth.gr (P.C.L.)

Abstract: This study investigates the interpretability of machine learning (ML) models applied to cumulative damage prediction during a sequence of earthquakes, emphasizing the use of techniques such as SHapley Additive exPlanations (SHAP), Partial Dependence Plots (PDPs), Local Interpretable Model-agnostic Explanations (LIME), Accumulated Local Effects (ALE), Permutation and Impurity-based technique. The research explores the cumulative damage during seismic sequences, aiming to identify critical predictors and assess their influence on the cumulative damage. Moreover, the predictors contribution in respect with the range of final damage is evaluated. Nonlinear time history analyses are applied to extract the seismic response of an eight-story Reinforced Concrete (RC) frame. The regression problem's input variables are divided into two distinct physical classes: pre-existing damage from the initial seismic event and seismic parameters representing the intensity of the subsequent earthquake, expressed by Park and Ang damage index (DI_{PA}) and Intensity Measures (IMs), respectively. The study offers a comprehensive review of cutting-edge ML methods, hyperparameter tuning, and ML method comparisons. A LightGBM model emerges as the most efficient, among 15 different ML methods examined, with critical predictors for final damage being the initial damage caused by the first shock and the IMs of the subsequent shock: I_{FVF} and SI_H . The importance of these predictors is supported by feature importance analysis and local/global explanation methods, enhancing the interpretability and practical utility of the developed model.

Keywords: seismic sequence; interpretable machine learning; successive earthquakes; seismic damage prediction; seismic damage accumulation; machine learning; explainable machine learning

1. Introduction

A series of successive occurring earthquakes can cause additional damage to already damaged buildings and consequently increasing the risk of collapse. Historical examples of such events include the New Madrid earthquake series in the central United States from 1811 to 1812, and the 1960 Chilean seismic sequence. The New Madrid sequence [1] comprised three primary shocks and numerous aftershocks, causing widespread structural damage and affecting regions as distant as Canada and the eastern United States. The 1960 Chilean earthquake sequence [2], initiated by a record-setting 9.5 magnitude mainshock, led to substantial destruction and casualties in Chile, as well as a series of tsunamis. One notable example of an earthquake sequence occurred in central Italy in 2016 [3,4]. Over a span of several months, a succession of earthquakes, with a 6.2 magnitude mainshock, affected the area, leading to significant destruction and casualties. Experts studying the series of seismic events found that they transpired on pre-existing fault lines that had been under stress for a considerable period. Moreover, they ascertained that the earthquake sequence had shifted the stress distribution within the region, potentially increasing the probability of subsequent seismic events. Another example is the

2019 Ridgecrest earthquake sequence [5] in California, which included a 6.4 magnitude mainshock followed by several aftershocks, including a 7.1 magnitude event. This sequence caused significant damage to infrastructure and homes in the region, and it demonstrated the potential for aftershocks to be almost as powerful as the mainshock. In February 2023, southeastern Türkiye and parts of Syria were hit by two strong earthquakes that resulted in numerous aftershocks and significant loss of life and injuries [6,7]. The earthquakes had magnitudes of 7.8 and 7.5 and occurred approximately nine hours apart. As an outcome of all the above examples, it is important to assess the cumulative damage and vulnerability of buildings to multiple earthquakes to prevent or minimize the potential losses.

Several studies focus on the effect of multiple earthquake events on the seismic performance of Reinforced Concrete (RC) structures. Abdelnaby in his PhD thesis [8] examines the consequences that successive seismic events have on RC buildings' degradation. Later, Hatzivassiliou and Hatzigeorgiou [9] investigate the effect of seismic sequences on the response of three-dimensional RC buildings, while Kavvadias et al. [10] examine the impact of aftershock severity characteristics on the seismic damage of a RC frame. Trevelopoulos and Guéguen [11] propose a framework to assess the vulnerability of RC structures during aftershock sequences based on period elongation. Other studies in this area include Shokrabadi et al. [12] who investigate the impact of mainshock-aftershock (MS-AS) ground motion pairing on the seismic performance, and Furtado et al. [13] assess the damage of infilled RC structures under MS-AS conditions. Di Sarno and Pugliese [14] study the seismic fragility of existing RC buildings with corroded bars under earthquake sequences, while Iervolino et al. [15] examine the accumulation of seismic damage in multiple MS-AS sequences. Rajabi and Ghodrati Amiri [16] develop behavior factor prediction equations for RC frames under critical MS-AS sequences using Artificial Neural Networks (ANNs). Soureshjani and Massumi [17] investigate the seismic behaviour of RC moment-resisting structures with concrete shear walls under MS-AS seismic sequences. Khansefid [18] studies the effect of structural nonlinearity on the risk estimation of buildings under MS-AS sequences in Tehran metro city. Also, Hu et al. [19] propose a framework for the seismic resilience assessment of buildings that considers the effects of both mainshock and multiple aftershocks. Finally, Askouni [20] investigated how repeated earthquakes affect RC buildings with in-plan irregularities. Overall, the research findings demonstrate the importance of considering the effects of multiple earthquakes in the design and assessment of RC structures.

Machine learning (ML) has gained increasing attention in earthquake and structural engineering due to its ability to predict the behaviour of structures. As a result, a number of studies have focused on exploring the application of ML in this field. The early papers, such as Zhao et al. [21], used ANNs to predict the response of simple simulated structures, and Stavroulakis and Antes [22] applied ANNs in nondestructive elastostatic identification of unilateral cracks. Later papers, such as De Lautour and Omenzetter [23] and Lagaros and Papadrakakis [24], focused on predicting the non-linear seismic response of 2D and 3D buildings, respectively, using ANNs. These papers demonstrated the potential of this ML method to accurately predict the behaviour of complex structures. Other papers have investigated the application of hybrid ML techniques, such as neuro-fuzzy methods, in predicting structural damage under earthquake excitation, as shown in Sánchez Silva and García [25], Alvanitopoulos et al. [26] and Vrochidou et al. [27]. Meanwhile, Mangalathu et al. [28] used ML techniques to classify buildings based on data from post-earthquake observations and Wang et al. [29] to predict the progressive building collapse. Recent papers also have explored the use of deep learning, which have shown great promise in rapid seismic response prediction of RC frames, as seen in Wen et al. [30]. In addition, Zhang et al. [31] proposed a physics-guided convolutional Neural Network (NN) for data-driven seismic response modelling, while Muradova and Stavroulakis [32] developed physics-informed NN for elastic plate problems with bending and Winkler-type contact effects, furthermore, Katsikis et al. [33] used the same type of NN in static rod and beam problems. Several papers explore the use of ML for seismic response prediction, including Morfidis and Kostinakis [34], who used ANNs to predict the damage state of RC buildings, and Hwang et al. [35], who used ML to predict the seismic demand and collapse of ductile RC building frames.

Other papers focus on developing earthquake scenarios for building portfolios [36], rapid seismic response prediction [37], and multivariate seismic classification [38]. Kazemi et al. [39] presented a ML-based approach for classification of the structural behaviour of tall buildings with a diagrid structure. Karbassi et al. [40] and Ghiasi et al. [41] proposed decision tree and support vector machine algorithms, respectively, to predict damage in RC buildings. Thaler et al. [42] used ML techniques to enhance tail-end prediction of seismic response statistics, while Chen et al. [43], and Jia and Wu [44] investigated probabilistic ML methods for predicting the performance of structures and infrastructures. The current knowledge also cover various application of ML on seismic fragility and vulnerability prediction, including the development of seismic fragility curves, as in Kiani et al. [45], fragility and vulnerability assessment of RC structures as in Kazemi et al. [46], fragility curve parameters prediction for RC buildings as in Dabiri et al. [47], assessing seismic risk and fragility of underground structures as in Yang et al. [48] and efficient and precise seismic vulnerability assessment of urban structures as in De-Miguel-Rodríguez et al. [49]. Moreover, Morfidis et al. [50] developed a user-friendly software application that leverages ANNs for rapid damage assessment of RC buildings in earthquake scenarios. Lastly, Lazaridis et al. [51,52] presented studies on forecasting the structural damage experienced by a RC frame subjected to both individual and successive seismic events using ML methods. Additional information regarding the application of ML in the field of earthquake and structural engineering can be found in the respective review papers [53–62].

In general, ML models are utilized in earthquake engineering to deliver rapid and accurate seismic vulnerability assessment. Despite this, the advanced of these methods have been criticized as “black boxes”, due to lack of transparency and reasonable explanation of their predictions, questioning their reliability. To address this issue several techniques have been developed recently [63]. Global methods focus on capture the general patterns and trends of models, whereas local approaches target specific data points and provide insights for individual predictions. These kind of techniques have been applied by Mangalathu et al. [64] for seismic performance assessment of infrastructure systems, by Wakjira et al. [65] in flexural capacity prediction of RC beams strengthened with FRCM, by Junda et al. [66] in seismic drifts estimation of CLT buildings and by Demertzis et al. [67] in damage prediction of RC buildings following a single earthquake.

In this study, the expansion of interpretation techniques to the concept of cumulative damage during a sequence of earthquakes is carried out, with the pre-existing seismic damage being taken into account. The primary questions addressed in the present paper include: identifying the most important predictors for damage accumulation, understanding how these variables impact the final damage, and determining the range of cumulative damage in which their contributions lie. Furthermore, a thorough review of employed cutting-edge ML methods is offered, with hyperparameter tuning and ML methods comparisons being incorporated. To this end, the transformation of the problem from the perspective of earthquake engineering to that of ML through the adopted workflow and methodology is discussed.

2. Feature Selection and Dataset Configuration

The input variables for our regression problem fall into two distinct physical classes: the pre-existing damage from the initial seismic event, along with the seismic parameters representing the intensity of the subsequent earthquake, with the goal of forecasting the cumulative damage occurred by two consecutive earthquakes. The first class of feature, as well as the target, is expressed in terms of Park and Ang damage index (DI_{PA}) [68]. For the second category, a preliminary feature selection process has been conducted. During this process, the outcomes of several studies [34,69–77] investigating the interdependence between seismic parameters and damage indicators of RC structures has been taken into account. As a result, 16 prominent Intensity Measures (IMs) have been selected as seismic damage predictors to express the severity of the second shock. The selected IMs, as well as the damage index, are described in the following section.

2.1. Ground Motion IMs and Damage Index

2.1.1. Ground Motion IMs

Seismic ground motion IMs are metrics employed to quantify the intensity or severity of seismic acceleration signals. These measures play a crucial role in evaluating a site's seismic hazard, predicting seismic demand on structures, and designing earthquake-resistant structures. Various IMs have been suggested over time, each with its own pros and cons. Peak Ground Acceleration (PGA) [78] is among the most commonly used ground motion signal IMs. PGA represents the maximum absolute acceleration of ground motion during an earthquake and is extensively used in seismic hazard analysis and building design, as it offers a straightforward indication of ground shaking intensity. Furthermore, our suit of IMs encompass amplitude parameters such as the maximum absolute values of ground velocity (PGV), and ground displacement (PGD) signals. The Arias intensity (I_A) [79] and Cumulative Absolute Velocity (CAV) [78] are additional seismic ground motion IMs that supply information about the overall amount of ground motion energy during an earthquake. Both I_A and CAV are determined as integrals of ground motion acceleration over time, offering a more comprehensive depiction of the seismic signal compared to PGA or PGV, as they account for both the amplitude and duration of the signal.

The frequency content of ground motion signals significantly influences a structure's response. This content can be assessed in a simplified manner using the corresponding frequency PGA/PGV or by calculating the average zero-crossing count of the acceleration time history per unit time. If the number of zero-crossings is denoted as u_0 , the fraction of I_A over u_0^2 is recognized as the potential destructiveness measure, according to Araya and Saragoni (I_{AS}) [80]. The strong motion duration of seismic excitation considering as the time interval during which most of the total intensity is released, is another vital parameter. Two definitions of strong motion duration, Trifunac and Brady (SMD_{TB}) [81] and Reinoso, Ordaz, and Guerrero (SMD_{ROG}) [82], are based on the Arias intensity's time evolution. The bracketed duration after Bolt (SMD_{Bolt}) [83], which is calculated based on the initial and final instances when the acceleration exceeds 5% of g , is also utilized.

Advanced measures can be obtained by merging the above parameters, such as power P_{90} [78], a_{rms} [78], characteristic intensity (I_c) [78], the potential damage metric by Fajfar, Vidic, and Fischinger (I_{FVF}) [84], and the intensity measure by Riddell and Garcia (I_{RG}) [85]. However, spectral values reliant on the fundamental structural period are not applicable due to the increase in the fundamental period during the initial earthquake. As an alternative, the Housner intensity (SI_H) [86], which aggregates the values of pseudo-velocity spectrum (PSV) to a constant interval of periods and exhibits a strong correlation with seismic damage, is utilized. A brief depiction of the formulas for the studied IMs is given in Table 1. IMs values are derived with Python [87] and NumPy [88] code, whereas the computation of acceleration spectra is performed using OpenSeismoMatlab [89].

Table 1. Mathematical Formulas of the examined IMs.

Num	Name	Expression	Ref.	Num	Name	Expression	Ref.
1	PGA	$\max a_g(t) $	[78]	9	SMD _{ROG}	$t(H_d = 97.5\%) - t(H_d = 2.5\%)$	[82]
2	PGV	$\max v_g(t) $	[78]	10	SMD _{Bolt}	$t_{last}^{a_g > 0.05g} - t_{1st}^{a_g > 0.05g}$	[83]
3	PGD	$\max d_g(t) $	[78]	11	P ₉₀	$\frac{I_A(H_d=95\%) - I_A(H_d=5\%)}{SMD_{TB}}$	[78]
4	I _A	$\frac{\pi}{2g} \int_0^{t_{end}} a_g^2(t) dt$	[79]	12	a _{rms}	$\sqrt{\frac{1}{SMD_{TB}} \int_{t_{5\%}}^{t_{95\%}} a_g(t)^2 dt}$	[78]
5	CAV	$\int_0^{t_{end}} a_g(t) dt$	[78]	13	I _c	$a_{rms}^{1.5} \cdot SMD_{TB}^{0.5}$	[78]
6	PGA/PGV	$\frac{PGA}{PGV}$	[78]	14	I _{FVF}	$PGV \cdot SMD_{TB}^{0.25}$	[84]
7	I _{AS}	$\frac{I_A}{u_c}$	[80]	15	I _{RG}	$PGD \cdot SMD_{TB}^{\frac{1}{3}}$	[85]
8	SMD _{TB}	$t(H_d = 95\%) - t(H_d = 5\%)$	[81]	16	SI _H	$\int_{0.1}^{2.5} PSV(T, \zeta = 0.05) dT$	[86]

* $a_g(t)$, $v_g(t)$ and $d_g(t)$: ground acceleration, velocity and displacement signals, H_d : Husid Diagram [90].

2.1.2. Damage Index

Seismic damage in structures manifests as a reduction in resistance to external forces, resulting in instability. The Park and Ang damage index (DI_{PA}) is a reliable seismic damage metric that represents structural damage as a linear combination of excessive deformation and damage developed by repeated cyclic loading effects. This index is calculated by summing the maximum bending responses and the energy absorbed by plastic hinges during an earthquake, as shown in Equation (1). Kunnath's modified version of the index [91] is calculated using Equation (2). The total damage index ($DI_{G,PA}$) is obtained as an adjusted mean of sub-damage values, where each sub-damage belongs to each structural member. The weight of each sub-damage is proportional to the energy used by its corresponding structural member, according to Equation (3). A low value of $DI_{G,PA}$, close to zero, indicates that the structure has experienced minimal damage and exhibited an elastic response. Conversely, a value around the unity and larger signifies that the structure is nearing collapse. Overall damage indices, such as $DI_{G,PA}$, offers a quantitative assessment of a structure's seismic damage and has been utilized in several studies [34,69–72,76,92] to evaluate the post-earthquake condition of buildings.

$$DI_{PA} = \frac{\delta_m}{\delta_u} + \frac{\beta}{Q_y \delta_u} \int dE \quad (1)$$

$$DI_{PA,component} = \frac{\theta_m - \theta_r}{\theta_u - \theta_r} + \frac{\beta}{\theta_u M_y} E_h \quad (2)$$

$$DI_{G,PA} = \frac{\sum E_i DI_{PA,component}}{\sum E_i} \quad (3)$$

The damage index equation comprises several variables associated with structural element capacity and response, such as the maximum displacement response (δ_m), the ultimate displacement capacity (δ_u), the strength deterioration model constant (β) [93], the absorbed cumulative hysteretic energy ($\int dE$), the yield strength (Q_y), the maximum rotation of the member throughout the response (θ_m), the member's ultimate rotation capacity (θ_u), and the recoverable rotation during unloading (θ_r).

2.2. Dataset Configuration

For this study, a comprehensive dataset containing both artificial and natural seismic sequences is used. The former are generated by randomly pairing 318 real acceleration records, while the latter comprises 111 real pairs of sequential shocks. Further information about the suite of seismic sequences can be found in [52]. All seismic records are sourced from the ESM [94] and PEER NGA West [95] databases, with a 20-second intermediate time gap introduced between consecutive records (Figure 1)

to eliminate overlapping structural responses. Nonlinear time history analyses are conducted on both seismic sequences and individual ground motion records to investigate the seismic structural response. The real seismic sequences are presented in Table A1 in Appendix A, while the histograms and the probability density curves for all variables across the total dataset are provided in Figure A1 in the same appendix. In this study an eight-story RC frame (Figure 2) designed by Hatzigeorgiou and Liolios [96], without seismic provisions [97] or retrofitting [98–103] is examined. The finite element simulation is performed using IDARC 2D [104], and the distributed plasticity concept with a three-parameter Park hysteretic model [105] is employed. The concrete and steel materials are modeled based on their effective strength and strain properties, and the structure's initial elastic fundamental period is 1.27 seconds. GNU Octave [106,107] code is also written to automate the creation of IDARC 2D input files and the subsequent processing of the results.

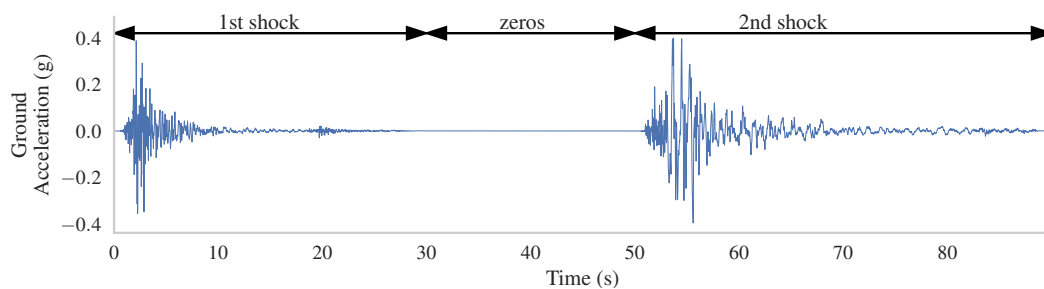


Figure 1. Indicative seismic signal comprises two successive ground motions.

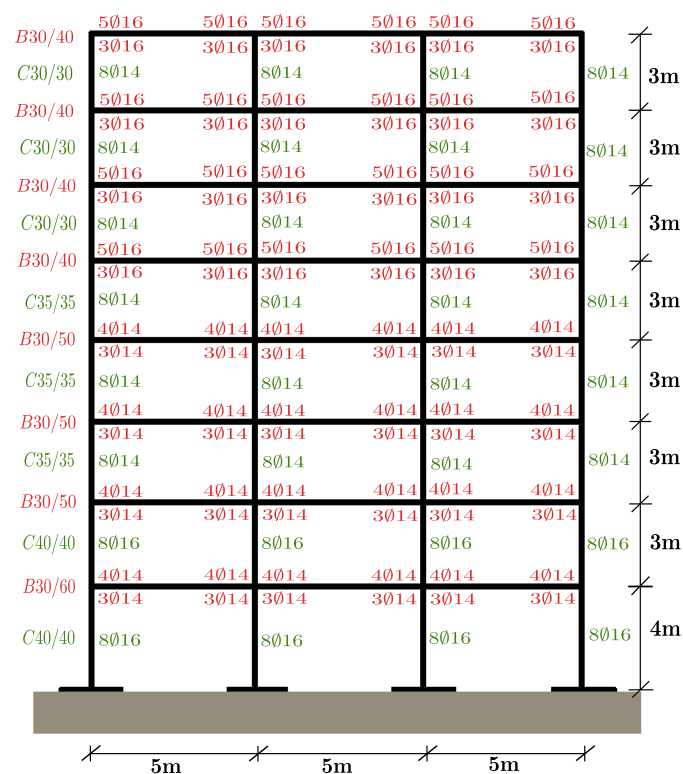


Figure 2. The examined RC frame.

3. Machine Learning Methods, Hyperparameter Tuning and Interpretation

Machine Learning (ML) is a branch of artificial intelligence that focuses on create algorithms capable to learn from and make predictions based on data, with the ultimate goal to generalize effectively to new and unseen examples. A subcategory of ML is the Supervised Learning which deals with building models that predict a target output using input features. By leverage the training dataset,

model parameters are adjusted to minimize the cost function and thereby achieving an effective model fit. A loss function (\mathcal{L}) measures the difference between the predicted and actual outputs for an individual data point. Conversely, a cost function \mathcal{J} computes the total error between the predicted and true outputs for all data points in the training dataset. A general form of a cost function \mathcal{J} provided in Equation (4) in which the total error estimated as the average of total m data point losses (\mathcal{L}).

The main two problems which the ML techniques could be suffered from are the high variance and high bias problem or in other words overfitting and underfitting respectively. Overfitting define the weakness of the model to generalize its prediction after training to new unseen data as those of the test set, in case of high bias the model does not have the required complexity or the proper input features and behave poorly both to train and test set. To address the variance problem, the regularization technique is used in ML. It adds a penalty term to the cost function that discourages the model from having too many parameters or from having parameters with large magnitude. This helps to prevent overfitting and improve the generalization performance of the model.

$$\mathcal{J}(\text{model weights}) = \frac{1}{m} \sum_{k=1}^m \mathcal{L}(y_k, \hat{y}_k) + \text{regularization terms} \quad (4)$$

3.1. Linear Models

In linear models, the output variable is determined by calculating the weighted average of the input features and adding a constant, known as the bias term or intercept (Equation 5). Various types of Linear Regression (LR) [108,109] exist, which are further explained below.

The most commonly used linear regression model based on Ordinary Least Squares (OLS), and establishes the association between input features and the output variable by minimizing the sum of squared differences between real and predicted values. As a fundamental and classic approach to linear regression, OLS is often utilized as a baseline against which more advanced models are evaluated. Lasso Regression [110] adds a ℓ_1 regularization term to the loss function \mathcal{J} , which results in sparse solutions, where many of the coefficients are exactly equal to zero. The regularization term is defined as the sum of the absolute values of the coefficients. This can lead to feature selection and interpretability, but can be too aggressive in shrinking the coefficients and result in a suboptimal solution. Ridge Regression [111] (RR) adds a ℓ_2 regularization term to the loss function \mathcal{J} , which results in non-sparse solutions, where the coefficients are shrunk towards zero, but not necessarily exactly equal to zero. The regularization term is expressed as the sum of the squared coefficients, which contributes to achieving more stable solutions and prevent overfitting. Elastic Net [112] (EN) is a combination of Lasso and Ridge Regression, which adds both ℓ_1 and ℓ_2 regularization penalties to the loss function \mathcal{J} (Equation (6)), weighted by a mixing parameter, ρ . The regularization parameter α determines the strength of the penalty applied to the coefficients, with larger values of α leading to stronger regularization. The mixing parameter ρ governs the balance between the ℓ_1 and ℓ_2 penalties in the model. When $\rho = 1$, it results in the Lasso penalty, whereas when $\rho = 0$, the Ridge penalty is applied. The adjustment of the mixing parameter, can fine-tuned the model and achieve the desired balance between the two penalties.

$$\hat{y}(x_j) = b + \sum_{j=1}^n w_j x_j \quad (5)$$

$$\mathcal{J}(w_j, b) = \underbrace{\frac{1}{2m} \sum_{k=1}^m (\hat{y}(x_{jk}) - y_k)^2}_{\text{OLS}} + \underbrace{\alpha \rho \sum_{j=1}^n |w_j|}_{\ell_1} + \underbrace{\frac{\alpha(1-\rho)}{2} \sum_{j=1}^n w_j^2}_{\ell_2} \quad (6)$$

3.2. Non-parametric Algorithms

The so-called non parametric algorithms have no trainable parameters and as a result, they get their name. An example of this kind of algorithm is the K-Nearest Neighbors [113] (KNN) is an instance-based technique that depends on memorizing the training dataset. The "K" in the method's name signifies the number of training data examples nearest to the point being examined, which are considered when making predictions. This is the primary hyperparameter of the method.

Decision Trees (DTs) [114] constitute a form of ML algorithm used to tackle classification and regression issues. The algorithm constructs a tree-like model illustrating the connections between input features and target variables. Every node in the tree represents a decision of input features, and each branch corresponds to a possible outcome of that decision. The leaves of the tree denote the final prediction or classification outcome. The algorithm's objective is to identify the optimal set of splits that most effectively separate the data, maximizing the homogeneity within each resulting subset. This is accomplished by using criteria such as Information gain or Gini impurity to determine the most suitable feature for data splitting at each node [114]. The depth at which a feature is used as a decision node in the tree is indicative of its relative importance, as it reflects the degree to which the feature contributes to the reduction of impurity. Additionally, a feature's importance increases as it is more frequently used in a tree's split points resulting in a greater impurity reduction. DTs provide several advantages such as interpretability, handling both continuous and categorical data, user-friendliness, and fast prediction times. However, they also have some drawbacks, such as being susceptible to overfitting and a tendency to generate complex trees that are challenging to interpret [114]. Addressing these challenges, various algorithms have been devised, such as Random Forests, which is an ensemble of DTs [115], and Gradient Boosted Decision Trees, a boosting technique for DTs [116].

3.3. Ensemble Trees

3.3.1. Random Forests

Random Forest [115] (RF) is an ensemble ML technique that relies on multiple DTs as its base learners, creating a multitude of trees and fusing their predictions to deliver a final result. This approach aims to tackle the overfitting issue often found in single DTs by aggregating the outputs of several trees. In the RF algorithm, each DT is formed using a random portion of the training data and a random set of features, expanding to its allowed full depth with the ultimate prediction determined by averaging the predictions of individual trees. Employing random data and feature subsets effectively addresses overfitting, as each tree has access to only a limited part of the data and is less prone to fit the data's noise. Nonetheless, random forests can be computationally demanding, particularly when working with large datasets, and interpreting random forests can be more complex than single DTs or linear models because the predictions are averaged across numerous DTs. Despite these drawbacks, RFs have earned popularity and extensive use due to their simplicity, adaptability, and impressive performance in a broad array of applications. An adaptation of RF algorithm is the Extremely Randomized Trees (ERTs) which introduced by Geurts et al. [117] in 2006. This method aims to further reduce the correlation between the trees in the forest to decrease overfitting and enhance the stability of the predictions. In ERTs, the feature selection process at each split in the DTs is randomized also, instead of using a greedy optimization algorithm like traditional DTs. Alongside random feature selection, ERTs also employ random thresholds for each feature at every split [117]. By incorporating randomness into the feature selection process, ERTs reduce the correlation between the trees in the forest and make the final predictions more resilient to minor changes of the data. This can lead to enhanced generalization performance, particularly when handling noisy data [117].

3.3.2. Boosted Trees

Boosted trees come in several variations, each with their distinct advantages and disadvantages. Adaboost (Adaptive Boosting) [118,119] is among the earliest boosting algorithms. Its fundamental principle is to assign higher weights to challenging to predict instances and lower weights to easily predicted instances. The algorithm trains a weak learner on the weighted training data and adjusts the weights based on the weak learner's performance. This process iterates, with each iteration training a weak learner and updating the weights. The final prediction is derived by combining all weak learners' predictions through weighted majority voting.

Gradient Boosting (GBoost) [116] initiates by fitting a simple decision tree to the training data, followed by fitting another decision tree to the residuals (the discrepancy between the actual target values and the first tree's predictions), and so forth. By integrating multiple trees' predictions, the algorithm strives to rectify previous trees' errors and incrementally enhance the model's overall performance. The central concept of Gradient Boosting is to utilize gradient information to direct model improvement. The gradient information stems from the loss function, which measures the difference between the actual target values and the predictions. The algorithm fits the subsequent tree to the loss function's negative gradient concerning the prediction to minimize the loss. Equation (7) explains the general idea behind Gradient Boosting.

$$F(x) = F_{n-1}(x) + \beta_n T_n(x) \quad (7)$$

where $F(x)$ is the final prediction of the model, $F_{n-1}(x)$ is the prediction of the previous iteration, $T_n(x)$ is the n -th tree model in the ensemble, β_n is the weight assigned to the n -th tree model.

The weights β_n are optimized to minimize the cost function \mathcal{J} (Equation (8)), which measures the difference between the true values and the model's predictions.

$$\mathcal{J}(\beta_n) = \sum_{i=1}^m \mathcal{L}(y_i, F_{n-1}(x_i) + \beta_n T_n(x_i)) \quad (8)$$

where $F_{n-1}(x)$ is the prediction of the previous iteration, m is the number of samples, and \mathcal{L} is the loss function.

The learning rate, also referred to as shrinkage or step size, governs the increase of adjustment in each iteration. A smaller learning rate results in slower convergence but generally produces more precise models.

Gradient Boosting presents several benefits over other ML methods, including its ability to handle complex non-linear relationships, work with both categorical and continuous variables, and perform effectively with noisy and incomplete data. However, gradient boosting can be resource-intensive and may overfit the training data if there are excessive trees or if the learning rate is excessively high. Furthermore, interpreting gradient boosting models can be more difficult compared to linear models or individual DTs.

With time, numerous variations of Gradient Boosting have arisen. Some of the most prominent versions of Gradient Boosting include the following: Microsoft's Light Gradient Boosting or LightGBM [120] has gained recognition due to its speed and effectiveness. Created for large-scale, high-dimensional datasets, LightGBM optimizes memory usage and training time. Its gradient-based one-side sampling (GOSS) method, which randomly samples a subset of instances for training each tree, is a key feature that minimizes model bias and improves performance. CatBoost [121,122] employs multiple overfitting prevention techniques such as regularization and early stopping to prevent the model from fitting data noise. XGBoost (eXtreme Gradient Boosting) [123], a gradient boosting implementation, can manage missing values and outliers by partitioning data into smaller segments and fitting trees to the subsets. XGBoost also uses regularization techniques to prevent overfitting and leverages parallel processing for rapid and efficient performance on large datasets. In contrast, Natural Gradient Boosting (NGBost) [124] prioritizes model interpretability through a

probabilistic framework for decision trees, enabling uncertainty estimation and generating meaningful feature importances that are not easily achieved in conventional gradient boosting.

3.4. Feedforward Neural Networks

ANNs are information processing structures initially inspired by the functioning of the human brain and biological neurons [125]. Specifically, the Multilayer Perceptron (MLP) [126–128], also known as Feedforward Neural Networks, is a type of ANN that serves as a potent learning model for addressing both classification and regression challenges. The MLP consists of artificial neurons called perceptrons or units and organized in layers. Its architecture is composed of successive fully connected layers, with the input layer containing the predictors or independent variables and the final output holding the target variable. Layers and units placed between the input and output layers are referred to as hidden layers and units. Within a hidden layer, each neuron obtains activations from every neuron in the preceding layer and sends its activation to all units in the following layer. More specifically, each neuron's value is calculated by applying a non-linear function, known as the activation function, to the previous layer units linear combination. The coefficients of the aforementioned linear combination are divided into weights (W_{ij}^ℓ) and biases (b_i^ℓ), collectively so-called as the trainable parameters of layer ℓ . The process of mapping input features to the output target, called Forward Propagation of a neural network and illustrated by Equation (9). The value a_i^ℓ of the i^{th} neuron in the ℓ^{th} layer depends on the j^{th} neuron in the $(\ell - 1)^{\text{th}}$ layer, while the vector α_j^0 represent the input attributes. Initially, random values and zeros assigned to weights and biases, respectively, at the beginning of training. During training, the process of estimating the cost function (\mathcal{G}) partial derivatives with respect to trainable parameters (W_{ij}^ℓ, b_i^ℓ) is known as back-propagation (Equations (10)) [129,130]. It begins by determining the error at the output layer through a comparison between the predicted target values $\hat{\mathbf{y}}$ and the actual target values \mathbf{y} . The error is then propagated backwards through the network. This procedure is iterated for every layer until the error reaches the input layer. To minimize the losses between the predicted and actual target values in the training set, these parameters are updated in each step according to the gradient descent optimization algorithm. This algorithm operates by taking incremental steps in the direction of the negative gradient of the cost function concerning the trainable parameters. The update rules for weights and biases are provided by Equations (11). Furthermore, the complexity of the MLP is influenced by the number of hidden layers and units, as well as the regularization parameter (λ), which functions similarly to the explanation in subsection 3.1, resulting in Equation (12) for the regularized cost function.

$$a_i^\ell = \sigma \left(\underbrace{\sum_j W_{ij}^\ell a_j^{\ell-1} + b_i^\ell}_{z_i^\ell} \right) \quad (9)$$

$$\frac{\partial \mathcal{G}}{\partial W_{ij}^\ell} = \left(\frac{\partial \mathcal{G}}{\partial a_i^\ell} \odot \frac{\partial a_i^\ell}{\partial z_i^\ell} \right) a_j^{\ell-1}, \quad \frac{\partial \mathcal{G}}{\partial b_i^\ell} = \frac{\partial \mathcal{G}}{\partial a_i^\ell} \odot \frac{\partial a_i^\ell}{\partial z_i^\ell} \quad (10)$$

where \odot denotes element-wise multiplication.

$$W_{ij}^\ell \leftarrow W_{ij}^\ell - \alpha \frac{\partial \mathcal{G}}{\partial W_{ij}^\ell}, \quad b_i^\ell \leftarrow b_i^\ell - \alpha \frac{\partial \mathcal{G}}{\partial b_i^\ell} \quad (11)$$

where α is the learning rate, and \leftarrow denotes assignment.

$$\mathcal{G}_{\text{reg}}(W_{ij}, b_i) = \mathcal{G}(W_{ij}, b_i) + \lambda \sum_{\ell=1}^L \sum_{i=1}^n \sum_{j=1}^m (W_{ij}^\ell)^2 \quad (12)$$

where L is the number of layers, n is the number neurons of layer ℓ , m is the number neurons of layer $\ell - 1$, and λ is the regularization parameter.

3.5. Hyperparameter Tuning and K-fold Cross Validation

The trainable parameters of every parametric ML model could be calibrated after minimization of the cost function on the training set, however this is not the case about the hyperparameters which should be set before the training. Consequently, each variation of hyperparameters' values configured a different instance of a ML method and by comparing the performance of multiple fitted models their optimized values are determined. Fine-tuned hyperparameters could significantly reduce the bias, variance effects and results in a model with high generalization ability. Hyperparameter tuning is frequently performed using the Grid Search method [131]; however, the Randomized Search Cross-Validation (CV) method [132] can be more efficient when dealing with a large number of hyperparameters and their possible values, as it reduces computational expense. This method selects randomly a subset of the possible hyperparameters values combinations and utilizing K-fold CV [133] to assess their performance. According to K-fold procedure, the dataset would be divided in K sections and every ML algorithm should be fitted K times in different every time training set constituted by K-1 parts and evaluated using the remaining Kth part. Afterward, for all of the examined models, the mean cross-validation predicting capacities, which have emerged from the K-fold cross-validation procedure, will be compared, enabling a conclusion to be reached concerning the best performing model for the given problem. All of the above processes are implemented using scikit-learn [134] which is an open source and freely accessible ML library built in Python.

3.6. Interpretation Methods

While interpreting a single decision tree can be relatively straightforward by analyzing its structure, including the impurity decrease at each node, gradient boosting models encompass numerous regression trees, making understanding them through individual tree examination more challenging. Moreover, the complexity of ANNs exacerbates interpretability difficulties of high-level ML methods. To address this issue, model-agnostic interpretation approaches have been developed recently [63]. This kind of techniques could be used to explain the predictions of any previous described ML method, without consider the internal structure of the model. In this section, several methods that have been developed to distill and interpret advanced and complex ML models, such as gradient boosting models or ANNs, are described.

3.6.1. Global Interpretation Methods

Contrary to local methods, which emphasize in particular instances, global interpretation approaches elucidate the mean performance of ML models. By employing expected values contingent on data distributions, these approaches facilitate the identification of prevailing trends. An illustrative example is the Partial Dependence Plot (PDP) [116,135], a type of feature effect plot that presents the projected outcome whilst marginalizing extraneous features. In essence, the PDP can be seen as the anticipated target response based on the input features of interest. Due to human perceptual constraints, the set of input features of interest should be limited, typically encompassing one or two features. For a single selected feature, the partial dependence function, at a specific feature value, indicates the average prediction when all data points are assigned that particular value. Additionally, Accumulated Local Effects (ALE) [136] plots extend the concept of PDP by accounting for potential feature interactions and correlations. Also, the Permutation feature importance, introduced by Breiman [115], is another global method that assesses the impact of individual features by measuring the change in model performance when the feature values are randomly shuffled, thereby breaking the relationship between the feature and the target.

3.6.2. Local Interpretation Methods

In contrast, local interpretation methods such as Local Interpretable Model-agnostic Explanations (LIME) [137] and SHapley Additive exPlanations (SHAP) [138] focus on explaining individual

predictions. More specifically, these methods clarify each feature's contribution to an individual prediction, which can be positive, negative, or neutral. LIME creates an artificial dataset in the vicinity of the examined data point, using the predictions of the complex model as ground truth. In the next step, fit an interpretable model (e.g., linear or decision tree) to the artificial dataset, to approximate the complex model locally with a simpler one. This process offers insights into the model's behaviour for a specific data instance. SHAP is an approach for interpreting ML models that provides a unified and scalable framework for explaining any ML model's output. SHAP values, rooted in cooperative game theory [139], offer a unified measure of feature importance that represent the contribution of each feature to a specific prediction. This technique assess a feature's influence on a prediction by taking into account all possible combinations of feature values and aggregating the resulting predictions. SHAP values quantify a feature's contribution to the prediction while considering feature interactions, and the sum of contributions for all features in a prediction is equal to the difference between the prediction and the average prediction. Choosing between SHAP and LIME depends on the context, as no consensus exists on which is universally better.

4. Results and Discussion

4.1. Hyperparameter Tuning and ML Models Comparison

An extensive examination of 10,000 distinct variations for the majority of ML methods outlined in Section 3 was conducted, utilizing the Randomized Search CV procedure. Each variation, also referred as an instance of an ML method, was configured with different values and combinations of hyperparameters. A total of 15 ML methods were assessed to determine the most efficient method for predicting the cumulative seismic damage under MS-AS sequences. The best performing instances for each of the examined ML methods are presented in Figure 3, which showcases their performance in terms of the determination coefficient R^2 . This coefficient was calculated for both the training and CV sets during the K-fold process, as well as for the final test set, which was reserved for unbiased evaluation and comprised 15% of the overall dataset. An absolute improvement of 10% in R^2 values was observed when comparing linear models to the most recently developed and advanced ML methods (ensemble, MLP). Our results highlight the high efficiency of the majority of ensemble methods, including both boosted and randomized trees, for predicting the cumulative seismic damage. The MLP model exhibits a slightly more effective bias-variance balance, as it achieves more similar performance across training, CV, and test sets in comparison with the other advanced models. The most efficient model showcased an instance of the LightGBM method which demonstrate the better CV and test performance, while the the poorest performance demonstrated by KNN, with $R^2 \approx 0.4$.

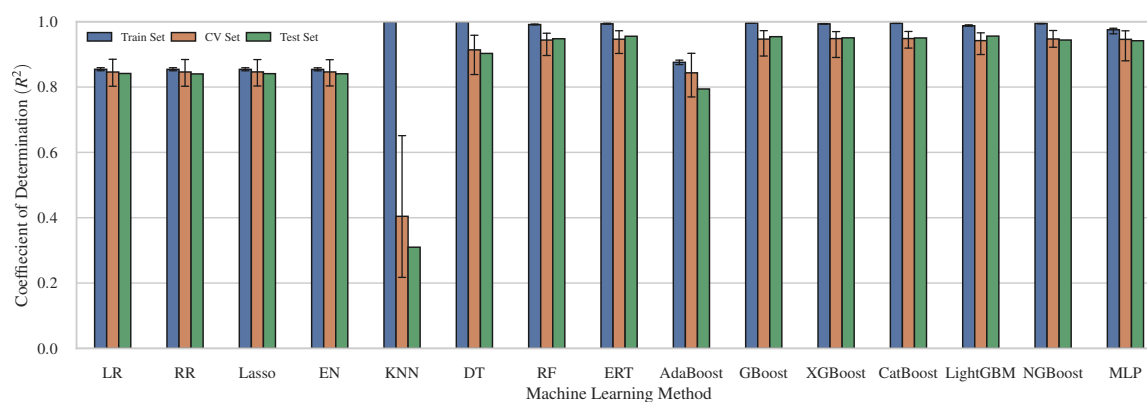


Figure 3. Side-by-Side Bar Plot Comparing the Performance of Different ML Methods on Training (10-fold), CV (10-fold), and Test Sets.

In the case of linear models (Section 3.1), regularization factors ranging from 0 to 1 were employed as hyperparameters. For DT, the splitting criterion options included Squared Error, Friedman MSE,

Absolute Error, and Poisson, while the maximum tree depth ranged from 0 to 100, both serving as hyperparameters. In the context of random forests, the number of trees varied from 0 to 200, with the splitting criterion, maximum tree depth, and maximum number of features for selection during the splitting procedure all functioning as hyperparameters. In contrast, the splitting criterion is not an individual hyperparameter for boosted trees, and consequently, exploration regarding this aspect is not conducted for this ML method family. However, the learning rate, a hyperparameter which controls the magnitude of updates during the training process of boosted trees, is included.

In Figure 4, the evolution of R^2 against the number of trees is compared for different types of boosted trees. The colored bands represent the influence of other hyperparameters on the performance of each method, with a 95% confidence interval. The AdaBoost method exhibits the poorest performance and lacks the ability to control overfitting as the number of trees increases. In contrast, LightGBM and XGBoost demonstrate the most optimized and robust performance, with higher R^2 values and lower variability as the number of trees increases. Additionally, CatBoost, GBoost, and NGBost, in descending order, present intermediate R^2 values, with GBoost exhibiting more robust behavior in response to changes in the number of trees. In summary, the majority of boosted trees display instances with high prediction performance, exceeding 0.9. The comparison between the two examined methods of randomized trees, is illustrated in Figure 5. The figure presents the evolution of the mean determination coefficient versus the number of base learner trees, with the hatched band depicting the uncertainty introduced by the other hyperparameters with the same as in case of boosted confidence interval.

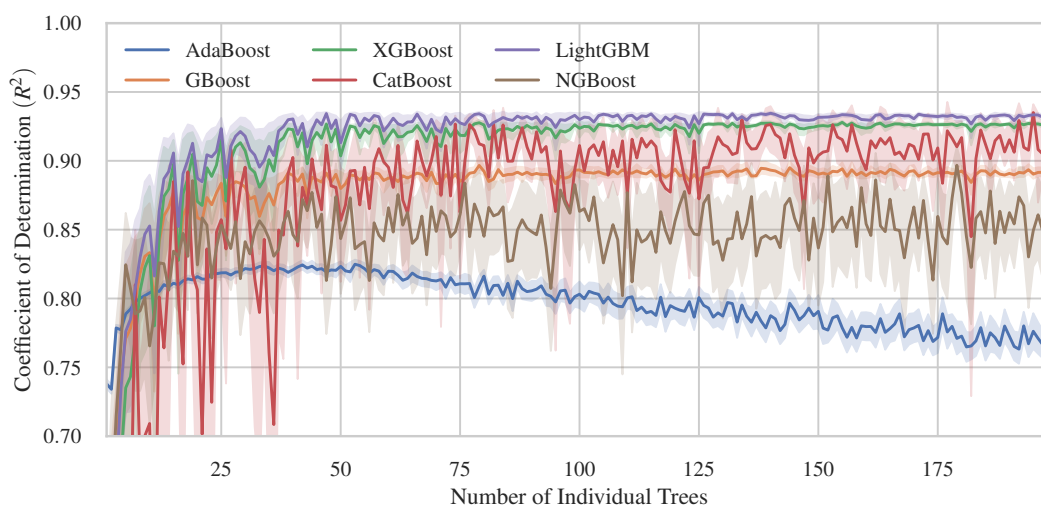


Figure 4. The evolution of R^2 versus the number of base learner trees for Boosted Trees.

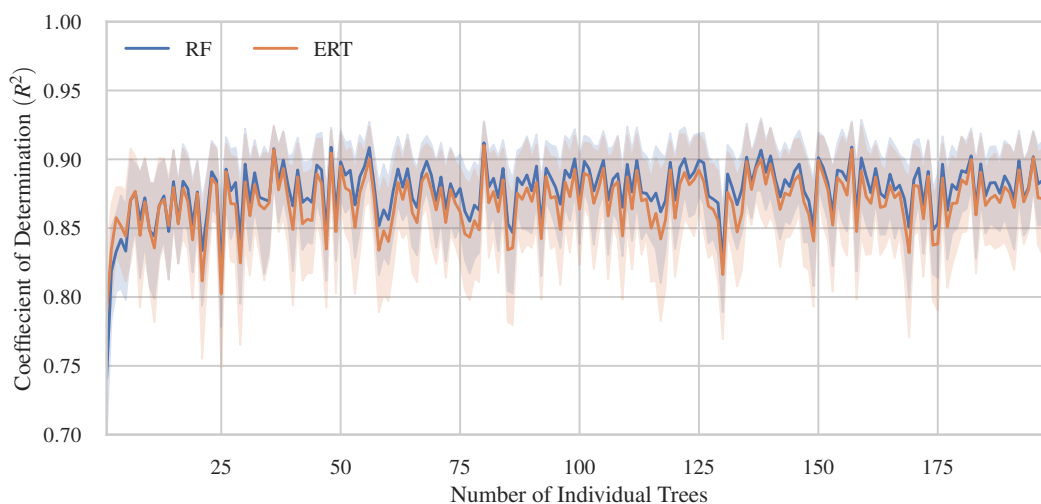


Figure 5. The evolution of R^2 versus the number of base learner trees for Random Forest Algorithms.

In the case of Multi-Layer Perceptrons (MLPs), the regularization parameter ranged from 0 to 1. MLPs consisted of one, two, or three hidden layers, each containing between 1 and 200 neurons. The activation function of the hidden neurons varied among identity (no activation function), hyperbolic tangent, and ReLU (Rectified Linear Unit); however, no activation function was consistently applied to the output neuron because the examined problem was being addressed as regression. The sigmoid function was not investigated, as it is more appropriate for use in the output neuron of an MLP classifier.

In Figure 6, the mean R^2 score, calculated in the CV sets using the K-fold validation procedure with K equal to 10, is depicted as being plotted against the total number of hidden neurons composing the MLP. The MLPs utilizing the ReLU activation function appear to exhibit better performance, followed by those with the Tanh activation function. The MLPs with no activation function essentially represent linear mapping, and their R^2 scores are similar to the results from linear models (Figure 3). Additionally, increasing the number of hidden units beyond 100 seems to have little impact on the method's performance, as the R^2 score plateaus around a constant value for each activation function after this point.

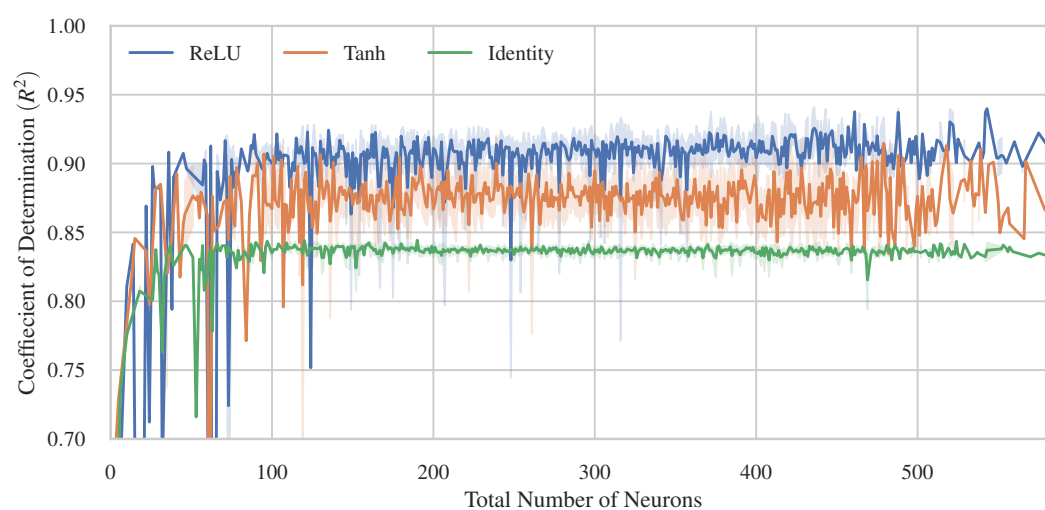


Figure 6. The evolution of R^2 versus the total number of neurons, for each activation function.

4.2. Interpretation of the Best ML Model

Generally, input features have varying levels of contribution to predicting the target outcome, and many of them may not be significant. The primary goal in model interpretation is to identify the key features and grasp how they influence the prediction of the target outcome. In this section, the interpretability of the devised ML model is explored, beginning with a discussion of feature importances.

4.2.1. Features Importances

Many past studies in earthquake engineering have been conducted to identify the most critical seismic parameters causing structural damage after an individual earthquake [34,69–77]. However, only a few [10,140,141] consider the accumulation of damage due to successive earthquakes and employ classical statistics to achieve this.

As the hyperparameter tuning and comparison of the examined ML methods are concluded, 8 of them are demonstrated to have high efficiency with $R^2 > 0.9$. The best ML models are instances of the ensemble and MLP methods and more specifically the more efficient is shown to be the LightGBM model with 195 trees, a maximum depth of 4, a maximum of 17 features (all) and learning rate equal to 0.1589. The Permutation technique, a global model-agnostic interpretation method, is implemented to extract the feature importances for the best models. The relative feature importances of the most efficient models, are presented in Figure 7. As observed, the majority of models agree that the most crucial predictor of the final damage is the pre-existing damage $DI_{G,PA,1st}$, followed by I_{FVF} , SI_H and $\frac{PGA}{PGV}$.

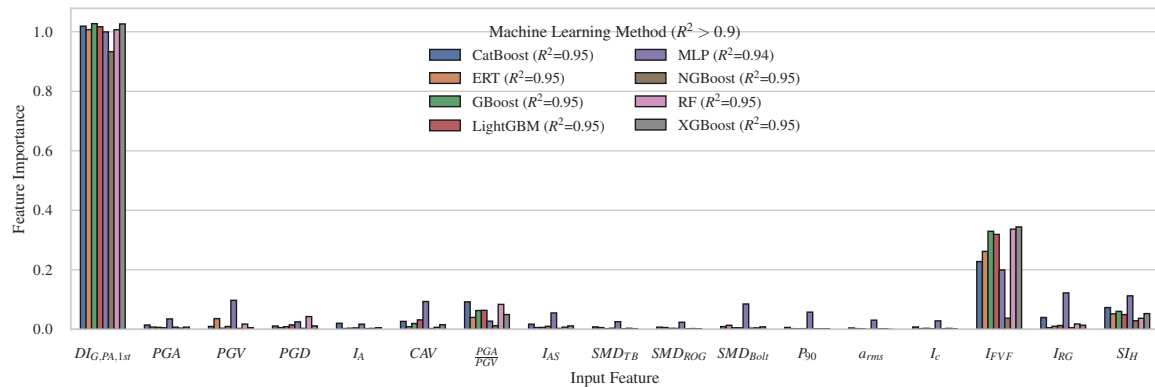


Figure 7. Feature Importances for the ML methods with $R^2 > 0.9$, according to Permutation method.

Individual DTs inherently perform feature selection by choosing suitable split points based on impurity reduction, which can be utilized to assess the significance of each feature. By measuring the decrease in impurity contributed by each feature throughout the tree, we can estimate its relative importance in the decision-making process. Since the most suitable ML method for our problem turned out to be LightGBM, a tree-based ensemble method, the notion of significance can be applied to our model by averaging the impurity-based feature importance across all trees. In Figure 8, the importances of input features, computed using three model-agnostic methods (LIME, SHAP, and Permutation) and the impurity-based method, which is only applicable to decision tree-oriented ML models, are presented. The y-axis of each subplot displays the features in descending order of importance according to the respective interpretation method. Both LIME and SHAP, as previously described, calculate the contributions of each feature for every data point individually. To obtain comprehensive importances, the mean absolute attribution and value for each method are calculated, respectively, over the entire dataset. As depicted in Figure 8, all interpretation methods identify the initial damage $DI_{G,PA,1st}$ as the most significant feature. The majority of interpretation methods indicate the second most important feature and the most significant among intensity measures (IMs) to be I_{FVF} , except for the impurity-based explanation, which identifies S_{IH} . LIME and SHAP rank S_{IH} as the third most important IM (third input feature), while Permutation suggests $\frac{PGA}{PGV}$ and Impurity proposes I_{AS} . The examined methods also, do not agree on the third most important IM: LIME identifies it as CAV, SHAP as $\frac{PGA}{PGV}$, Permutation as S_{IH} , and Impurity as PGD.

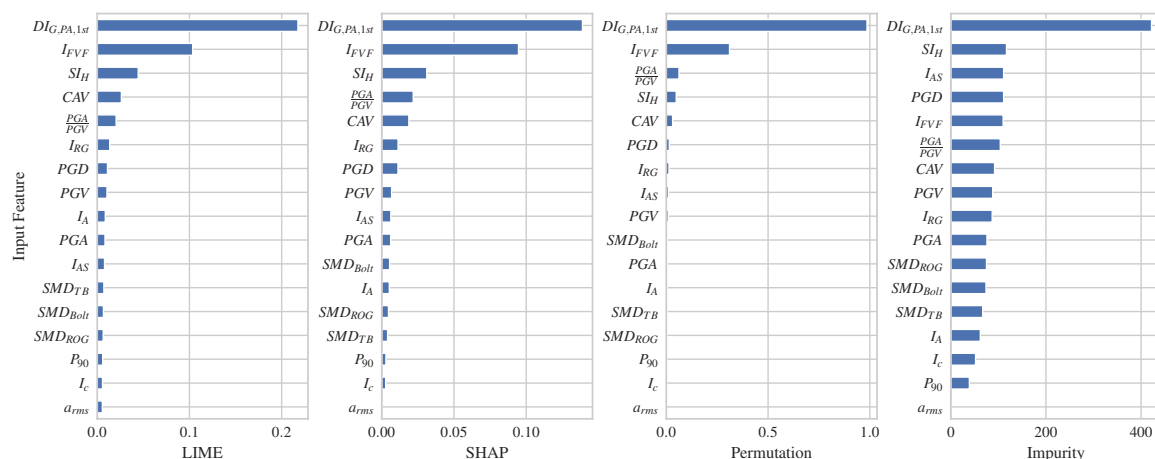


Figure 8. Feature Importances for the LightGBM best model, using different interpretation methods.

However, feature importances does not provide insights into whether a positive or negative change in an input variable leads to a corresponding positive or negative influence on the output variable, or the contrary.

4.2.2. Local Explanation Methods (LIME, SHAP)

On the other hand, local interpretation methods offer numerous advantages over other methods for explaining model predictions, including being model-agnostic, enabling explanation of individual predictions, and providing a unified approach for interpreting both linear and non-linear models. By considering each feature's contribution and its interactions with other features, deliver accurate explanations of the model's prediction. The sum of SHAP values for a given instance is equal to the difference between the prediction and the expected prediction value over the entire dataset, ensuring the model's overall behaviour is considered. For a model $f(x)$ that maps input x to a prediction, the SHAP value of feature i for a specific sample x is defined in Equation (13).

$$\phi_i(x) = \sum_{S \subseteq T} \frac{|S|!(|T| - |S|)!}{|T|!} [f(x_S \cup x_i) - f(x_S)] \quad (13)$$

where T is the set of all feature indices and x_S is the input vector with only the features in S present. The formula represents the average difference that adding feature i to the input makes over all possible combinations of the remaining features.

LIME assesses the impact of each input variable on an individual example basis. The distribution of LIME attributions across the entire dataset is depicted in Figure 9a. On the y-axis, the input features are arranged in descending order based on their mean absolute LIME per feature, as discussed in the previous section. Features such as $DI_{G,PA,1st}$, I_{FVF} , and SI_H appear to have a positive impact, while the $\frac{PGA}{PGV}$ ratio suggests a negative impact on the final damage outcome. The distribution of SHAP values for each predictor throughout the overall dataset is depicted in Figure 9b. The x-axis values display the SHAP values for each contributing variable, signifying their impact on the final damage in terms of $DI_{G,PA}$, while the y-axis arranges the predictors according to their importance. The colors in the figure represents the values of the input features. For instance, increased values of $DI_{G,PA,1st}$, I_{FVF} , SI_H , and PGV lead to a higher estimated final damage, $DI_{G,PA}$. On the other hand, lower values of $\frac{PGA}{PGV}$ and PGA result in higher SHAP values, which negatively influence the final damage. To examine the contribution variation of pre-existing damage and consequent seismic shock severity characteristics across different levels of the final damages, the dataset and the corresponding LIME attributions and SHAP values were grouped according to the following damage states: Minor ($0 < DI_{G,PA} \leq 0.3$), Moderate ($0.3 < DI_{G,PA} \leq 0.6$), Severe ($0.6 < DI_{G,PA} \leq 0.8$), Collapse ($0.8 < DI_{G,PA}$). The mean absolute values for each of the above-mentioned interpretation methods were calculated per damage state. The contributions of all the IMs are summarized and compared to those of the initial damage. In Figure 10, the results of the above process are displayed, with values normalized to unity. This illustration highlights the comparison between the contributions of seismic parameters and those of initial damage across each damage state. Quite different results emerge for each interpretation method, but both agree that the contribution of IMs is larger than that of $DI_{G,PA,1st}$ for minor damages. According to LIME, both IMs and established damage contribute equally for the three higher damage states. In contrast, SHAP estimates a larger impact for IMs, which is maximized for moderate damage and decreases as the damage level increases.

4.2.3. Global Explanation Methods (PDP and ALE)

Both PDPs and ALE, as global model-agnostic interpretation methods, can describe the general trends of our ML model with respect to input variables and depict the relationship between the cumulative seismic damage and a group of relevant predictors.

Figure 11 summarizes both PDPs and ALE results for the majority of the damage predictors. In each subplot, the y-axis depicts the expected value of the dependent variable, which in our case is the final damage ($DI_{G,PA}$), while the x-axis represents the value of the examined damage predictor. The positive impact of the initial damage on the final damage across its entire range of values can be observed. There is a sharp increase to the point of unity initial damage, and beyond that, the trend levels off horizontally. I_{FVF} has a positive impact for values between 50 and 150 $\text{cm} \cdot \text{s}^{-0.75}$, and zero

impact for smaller and larger values. The corresponding final damage ranges from 0.25 up to 0.56 by PDP and 0.25 to 0.65 by ALE. As observed, a larger cumulative damage range affected by I_{FVF} is identified by ALE. The SI_H has a positive impact for values between 80 and 180 cm, which appear to be responsible for a greater increase in the final damage. Also, final damage values between 0.3 and 0.45 seem to be influenced by this parameter. Subsequently, values of $\frac{PGA}{PGV}$ smaller than 15 s^{-1} appear to have a negative impact on the final damages and larger values present zero impact. CAV has an increasing outcome in the interval of 500–1500 $\frac{\text{cm}}{\text{s}}$, increasing the damage from 0.3 to 0.4, and zero effect outside of this range. I_{RG} appears to have a predominantly negative influence on damage in the range of [0.2, 0.4]. Other seismic parameters (PGD , PGV , I_{AS} , PGA , SMD_{Bolt} , I_A) seem to have less influence on shaping the final damage, affecting its values between 0.3 and 0.4. All the other input parameters appear to have very small or zero impact on the prediction of final damage, predicting values around 0.33, which is the mean output of our total sample. As a general observation, ALE indicates wider ranges of the dependent variable affected in comparison with PDP for every input feature.

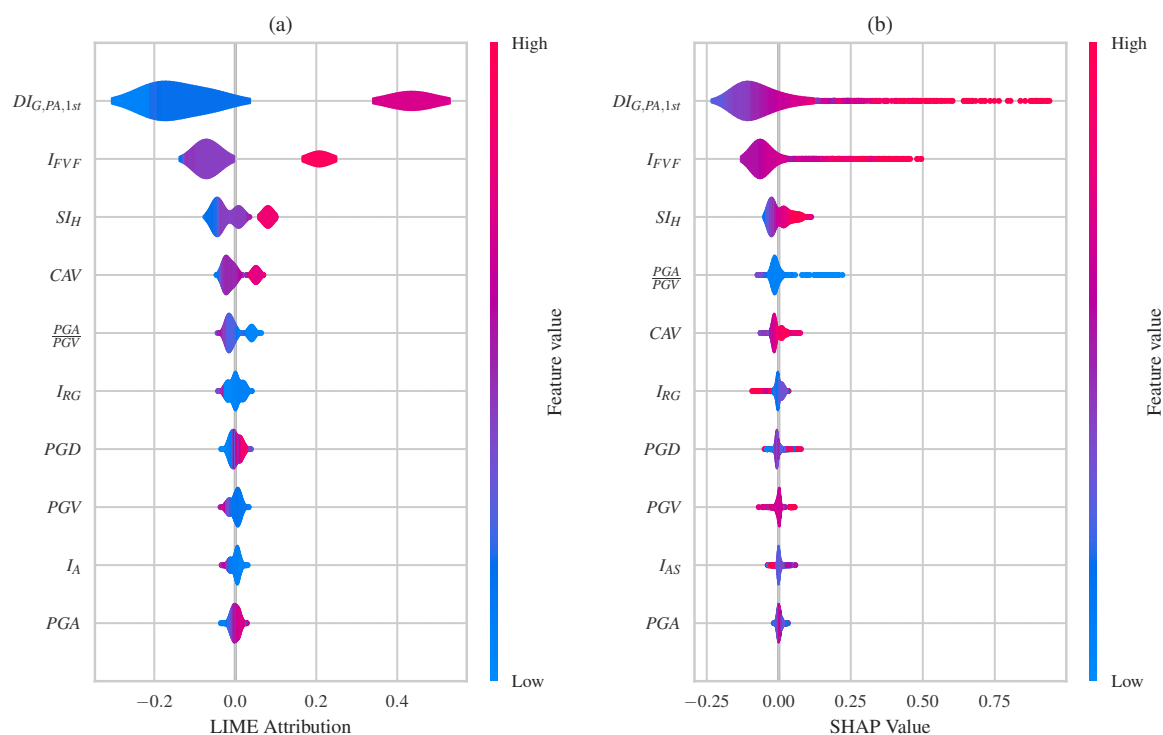


Figure 9. Violin plots represent the distribution of a) LIME attributions b) SHAP values.

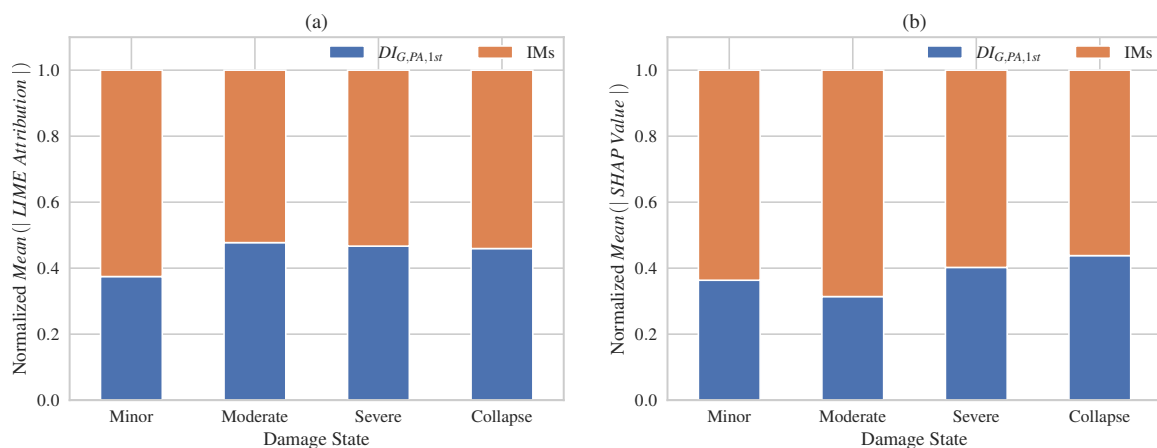


Figure 10. Stacked bar plot of grouped a) LIME Attributions b) SHAP values for IMs and $DI_{G,PA}$, normalized to unity.

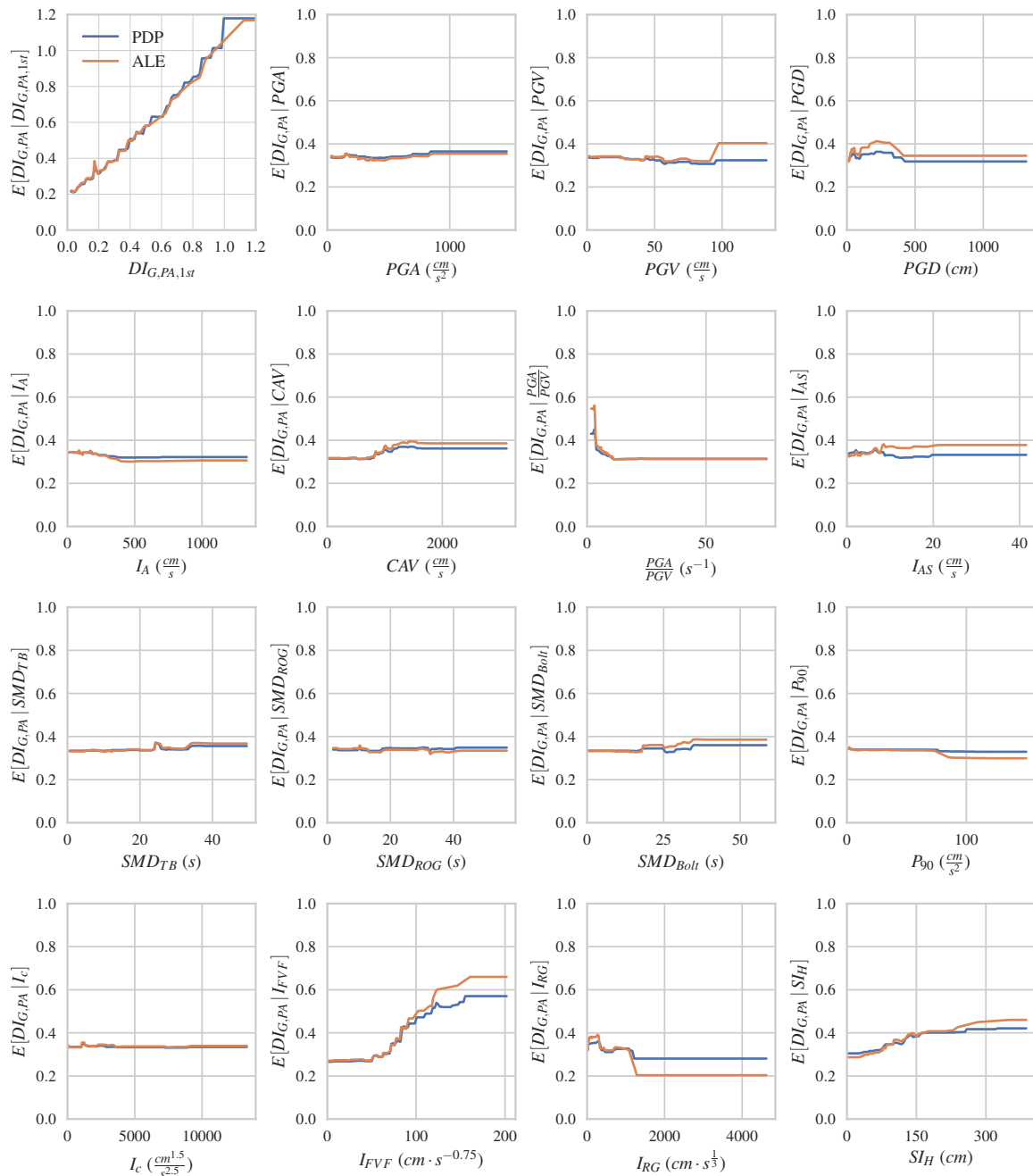


Figure 11. The expected value of $DI_{G,PA}$ in respect with each examined input feature, according to PDP and ALE methods.

In summary, this section explored the interpretability of the devised ML model by discussing feature importances, local explanation methods (LIME and SHAP), and global explanation methods (PDP and ALE). The results showed that initial damage, I_{FVF} , and SI_H are among the most crucial predictors of the final damage. Additionally, the local explanation methods provided insights into the positive or negative influence of each input variable on the cumulative seismic damage, while the global explanation methods described the general trends of the ML model with respect to damage predictors.

5. Conclusions

In this study, interpretable ML models to estimate the cumulative damage of an eight-story RC frame subjected to earthquake sequences were presented and analysed. Through the application of local and global explanation methods, a more profound understanding of the impact of individual

features and their interactions in the context of ultimate seismic damage was achieved. Local explanation techniques, LIME and SHAP, delivered in-depth insights into how each feature influences the prediction on an individual level, while global explanation approaches, PDP and ALE, facilitated comprehension of the general trends of the ML model concerning input variables. The utilization of these interpretation methodologies contributes to the creation of more transparent and interpretable models, which is of paramount importance for the implementation of ML methods in earthquake engineering problems. Our research investigated the accumulation of damage during a sequence of earthquakes, identifying crucial predictors and understanding their impact on the final damage ($DI_{G,PA}$). The input variables for the regression problem were divided into two distinct physical classes: pre-existing damage from the initial seismic event and the characteristics of the subsequent ground motion expressed using the Park and Ang damage index ($DI_{G,PA,1st}$) and 16 Intensity Measures (IMs), respectively. The main outcomes are:

- The most efficient model for predicting final seismic damage under MS-AS sequences was an instance of the LightGBM method with R^2 greater than 0.95, while the method with the poorest performance was KNN, with an R^2 value of approximately 0.4.
- Among the examined boosted trees, LightGBM and XGBoost demonstrated the most optimized and robust performance even against small changes in their hyperparameters. Moreover, they present great resistance to overfitting as the number of trees increases.
- In the case of Multi-Layer Perceptrons (MLPs), the ReLU activation function appeared to yield better performance, followed by the Tanh activation function. Also the MLP model present slightly better bias-variance balance than the other advanced ML models.
- All the interpretation methods identified the initial damage $DI_{G,PA,1st}$ as the most significant feature followed by the IMs of the subsequent seismic shock. However, the ranking of the IMs importance is varying between the adopted approaches. The majority of interpretation methods indicate the I_{FVF} as the most important IM, except for the impurity-based explanation, which identified SI_H . As the second most important IM, LIME and SHAP ranked SI_H , although Permutation ranked $\frac{PGA}{PGV}$, and Impurity ranked I_{AS} .
- In case of examining the effect of all the IMs in total, both LIME and SHAP local explanation methods show that the contribution of the subsequent ground motion is larger than that of initial damage $DI_{G,PA,1st}$. In general, the effect of the initial damage tends increase as the final increases. However, they differ in their estimation of contributions for higher damage states.
- The analysis of PDPs and ALE reveals key insights into the effects of damage predictors on the final damage. The pre-existing damage demonstrates a positive influence across the entire range of cumulative damage. Additionally, I_{FVF} and SI_H present a notable positive impact on moderate final damages. In contrast, $\frac{PGA}{PGV}$ values smaller than 15 s^{-1} seems to have a negative impact on moderate final damages, while CAV and I_{RC} demonstrate more complex effects in narrower range of the final damage.

Author Contributions: Conceptualization, P.C.L. and I.E.K.; methodology, P.C.L.; Python and Octave Programming, P.C.L.; validation, P.C.L., I.E.K. and K.D.; formal analysis, P.C.L.; investigation, P.C.L.; resources, P.C.L.; data curation, P.C.L.; writing—original draft preparation, P.C.L.; writing—review and editing, P.C.L., I.E.K. and K.D.; visualization, P.C.L.; supervision, L.I. and L.K.V.; project administration, I.E.K. and K.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The first author would like to express heartfelt gratitude to his parents, Cha-ralampos Lazaridis and Kiriaki Andreadou, for their significant support throughout his studies.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ML	Machine learning
RC	Reinforced Concrete
IM	Intensity Measure
NN	Neural Network
ANN	Artificial Neural Network
MLP	Multi-Layer Perceptron
LIME	Local Interpretable Model-agnostic Explanations
SHAP	SHapley Additive exPlanations
ALE	Accumulated Local Effects
PDP	Partial Dependence Plot
$a_g(t)$	ground acceleration signal
$v_g(t)$	ground velocity acceleration signal
$d_g(t)$	ground displacement signal
H_d	Husid Diagram
PSV	Pseudo-velocity spectrum
PGA	Peak Ground Acceleration
PGV	Peak Ground Velocity
PGD	Peak Ground Displacement
I_A	Arias intensity
CAV	Cumulative Absolute Velocity
I_{AS}	Seismic intensity after Araya and Saragoni
SMD_{TB}	Strong motion duration after Trifunac and Brady
SMD_{ROG}	Strong motion duration after Reinoso, Ordaz and Guerrero
SMD_{Bolt}	Strong motion duration after Bolt
a_{rms}	Root-mean-squared of ground acceleration signal
I_c	Characteristic Intensity
I_{FVF}	Potential damage measure after Fajfar, Vidic and Fischinger
I_{RG}	Intensity measure after Riddel and Garcia
S_{IH}	Spectral intensity after Housner
$DI_{G,PA,1st}$	The overall Park and Ang damage index after the first seismic shock (input feature)
$DI_{G,PA}$	The overall Park and Ang damage index after the second seismic shock (target)
CV	Cross-Validation
AdaBoost	Adaptive Boosting
DT	Decision tree
ERT	Extremely Randomized Trees
GBoost	Gradient boosting
KNN	K nearest neighbors
LightGBM	Light Gradient Boosting Machine
LR	Linear Regression
Lasso	Lasso Regression
RR	Ridge Regression
EN	Elastic Net
RF	Random forest
NGBoost	Natural Gradient Boosting
XGBoost	eXtreme Gradient Boosting

CatBoost Categorical Boosting

Appendix A

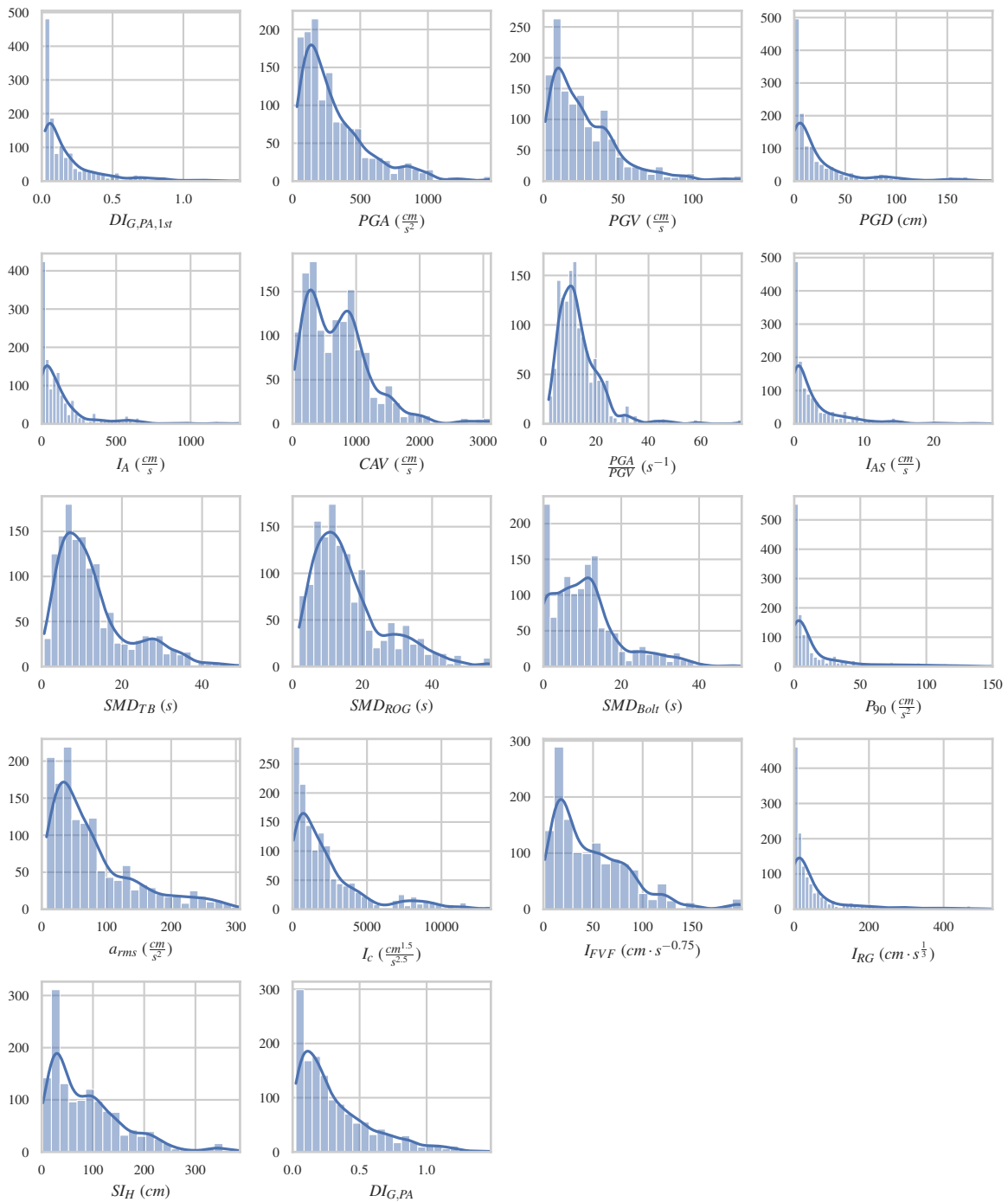


Figure A1. Histograms of Input Features ($DI_{G,PA,1st}$ and IMs) and Target ($DI_{G,PA}$).

Table A1. Seismic Metadata for the Real Sequences.

Region	1st Shock		2nd Shock		Station Code / Name	Component	PGA _{1st} (g)	PGA _{2nd} (g)					
	Date	M	Date	M									
Ancona	1972-06-14	4.2	1972-06-21	4.0	ANP	N-S	0.220	0.410					
Friuli	1976-09-11	5.8	1976-09-15	6.1	BUI	N-S	0.233	0.110					
						E-W	0.108	0.093					
					GMN	N-S	0.328	0.324					
						E-W	0.299	0.644					
Montenegro	1979-04-15	6.9	1979-04-15	5.8	PETO	E-W	0.304	0.089					
					1979-05-24	6.2	BAR	N-S	0.371	0.201			
			HRZ	E-W	0.360	0.267							
				N-S	0.215	0.066							
				E-W	0.254	0.076							
				N-S	0.282	0.033							
Imperial Valley	1979-10-15	6.5	1979-10-15	5.0	Holtville Post Office	E-W	0.236	0.030					
					315	N-S	0.221	0.254					
Mammoth Lakes	1980-05-25	6.1	1980-05-25	5.7	Convict Creek	90	0.419	0.371					
Irpinia	1980-11-23	6.9	1980-11-24	5.0	BGI	N-S	0.129	0.031					
					E-W	0.189	0.033						
					STR	N-S	0.224	0.018					
						E-W	0.320	0.032					
Gulf of Corinth	1981-02-24	6.6	1981-02-25	6.3	KORA	Trans	0.296	0.121					
					Logn	0.240	0.121						
Coalinga	1983-07-22	5.8	1983-07-25	5.2	Elm (Old CHP)	90	0.519	0.677					
						0	0.341	0.481					
Kalamata	1986-09-13	5.9	1986-09-15	4.8	KAL1	Trans	0.269	0.140					
						Logn	0.232	0.237					
					KALA	Trans	0.296	0.152					
						Logn	0.216	0.334					
Spitak	1988-12-07	6.7	1988-12-07	5.9	GUK	N-S	0.181	0.144					
						E-W	0.182	0.099					
Georgia	1989-01-08	4.0	1989-01-08	4.1	NAB	E-W	0.206	0.217					
						1991-05-03	5.6	1991-05-03	5.2	SAMB	N-S	0.354	0.208
											E-W	0.504	0.122
Erzican	1992-03-13	6.6	1992-03-15	5.9	AI 178 ERC MET	N-S	0.411	0.032					
						E-W	0.487	0.039					
Ilia	1993-03-26	4.7	1993-03-26	4.9	PYR1	Logn	0.109	0.100					
Northridge	1994-01-17	6.7	1994-01-17	5.9	Moorpark - Fire Station	90	0.193	0.139					
						180	0.291	0.184					
					1994-01-17	5.2	Pacoima Kagel Canyon	360	0.432	0.053			
					1994-03-20	5.3	Rinaldi Receiving Station	228	0.874	0.529			
					Sepulveda Hospital	270	0.752	0.102					
Sylmar - Olive Med	90	0.605	0.181										
Umbria Marche	1997-09-26	5.7	1997-09-26	6.0	CLF	N-S	0.276	0.197					
						E-W	0.256	0.227					
					NCR	N-S	0.395	0.502					
Kalamata	1997-10-13	6.5	1997-11-18	6.4	KRN1	Trans	0.119	0.071					
						Logn	0.118	0.092					
Bovec	1998-04-12	5.7	1998-08-31	4.3	FAGG	N-S	0.024	0.023					

Continued on next page

Region	1st Shock		2nd Shock		Station Code / Name	Component	PGA _{1st} (g)	PGA _{2nd} (g)					
	Date	M	Date	M									
Azores Islands	1998-07-09	6.2	1998-07-11	4.7	HOR	E-W	0.023	0.026					
						N-S	0.405	0.082					
						E-W	0.369	0.092					
Izmit	1999-08-17	7.6	1999-11-12	7.3	ARC	N-S	0.210	0.007					
						E-W	0.132	0.007					
					ATK	N-S	0.102	0.016					
						E-W	0.167	0.016					
					DHM	N-S	0.090	0.017					
						E-W	0.084	0.017					
					FAT	N-S	0.181	0.034					
						E-W	0.161	0.024					
					KMP	N-S	0.102	0.014					
						E-W	0.127	0.017					
ZYT	N-S	0.119	0.021										
Athens	1999-09-07	5.9	1999-09-07	4.3	SPLB	Trans	0.324	0.059					
						Logn	0.341	0.071					
Chi-Chi	1999-09-20	7.6	1999-09-20	6.2	TCU071	N-S	0.651	0.382					
						E-W	0.528	0.193					
					TCU129	N-S	0.624	0.398					
						E-W	1.005	0.947					
					1999-09-25	6.3	TCU078	N-S	0.307	0.387			
								E-W	0.447	0.266			
TCU079	N-S	0.424	0.626										
E-W	0.592	0.776											
Duzce	1999-11-12	7.3	1999-11-12	4.7	AI 010 BOL	E-W	0.820	0.060					
Bingöl	2003-05-01	6.3	2003-05-01	3.5	AI 049 BNG	N-S	0.519	0.147					
						E-W	0.291	0.068					
L Aquila	2009-04-06	6.1	2009-04-07	5.5	AQK	N-S	0.353	0.081					
						E-W	0.330	0.090					
					AQV	N-S	0.545	0.146					
						E-W	0.657	0.129					
					AVZ	N-S	0.069	0.021					
					2009-04-09	5.4	AQA	N-S	0.442	0.057			
Darfield	2010-09-03	7.0	2011-02-21	6.2	Botanical Gardens	S01W	0.190	0.452					
						N89W	0.155	0.552					
					Cashmere High School	S80E	0.251	0.349					
						Cathedral College	N26W	0.194	0.384				
					Christchurch Hospital	N64E	0.233	0.478					
						N01W	0.209	0.346					
					S89W	0.152	0.363						
					Emilia	2012-05-20	6.1	2012-05-29	6.0	MRN	N-S	0.263	0.294
											E-W	0.262	0.222
2012-06-03	5.1	2012-06-12	4.9	T0827	N-S	0.490	0.585						
					E-W	0.263	0.234						
Central Italy	2016-08-24	6.0	2016-08-24	5.4	AQK	E-W	0.050	0.010					
						2016-08-26	4.8	AMT	N-S	0.375	0.336		
									E-W	0.867	0.325		
	2016-10-26	5.4	2016-10-26	5.9	CMI	N-S	0.341	0.308					
						E-W	0.720	0.651					
CNE	E-W	0.556	0.537										
2016-10-30	6.5	CIT	N-S	0.052	0.213								

Continued on next page

Region	1st Shock		2nd Shock		Station Code / Name	Component	PGA _{1st} (g)	PGA _{2nd} (g)
	Date	M	Date	M				
	2016-10-26	5.9	2016-10-30	6.5	CLO	E-W	0.092	0.325
						N-S	0.193	0.582
						E-W	0.183	0.427
						CNE	0.380	0.294
						MMO	0.168	0.188
	2016-10-30	6.5	2016-10-31	4.2	T1213	E-W	0.170	0.189
						NOR	0.215	0.311
						N-S	0.867	0.185
	2017-01-18	5.5	2017-01-18	5.4	PCB	E-W	0.794	0.212
						N-S	0.586	0.561
Dodecanese Islands	2019-08-08	4.8	2020-10-30	7.0	GMLD	E-W	0.408	0.388
						N-S	0.450	0.899
						E-W	0.673	0.763

References

- Atwater, B.F. Evidence for great Holocene earthquakes along the outer coast of Washington State. *Science* **1987**, *236*, 942–944.
- Plafker, G.; Savage, J.C. Mechanism of the Chilean earthquakes of May 21 and 22, 1960. *Geological Society of America Bulletin* **1970**, *81*, 1001–1030. doi:10.1130/0016-7606(1970)81[1001:MOTCEO]2.0.CO;2.
- Chiaraluce, L.; Di Stefano, R.; Tinti, E.; Scognamiglio, L.; Michele, M.; Casarotti, E.; Cattaneo, M.; De Gori, P.; Chiarabba, C.; Monachesi, G.; others. The 2016 central Italy seismic sequence: A first look at the mainshocks, aftershocks, and source models. *Seismological Research Letters* **2017**, *88*, 757–771. doi:10.1785/0220160221.
- Gatti, M. Peak horizontal vibrations from GPS response spectra in the epicentral areas of the 2016 earthquake in central Italy. *Geomatics, Natural Hazards and Risk* **2018**, *9*, 403–415. doi:10.1080/19475705.2018.1445665.
- Brandenberg, S.J.; Wang, P.; Nweke, C.C.; Hudson, K.; Mazzoni, S.; Bozorgnia, Y.; Hudnut, K.W.; Davis, C.A.; Ahdi, S.K.; Zareian, F.; others. Preliminary report on engineering and geological effects of the July 2019 Ridgecrest earthquake sequence. Technical report, Geotechnical Extreme Event Reconnaissance Association, 2019. doi:10.18118/G6H66K.
- Naddaf, M. Turkey-Syria earthquake: What scientists know. *Nature* **2023**. doi:10.1038/d41586-023-00364-y.
- İlki, A.; Demir, C.; Goksu, C.; Sarı, B. A brief outline of February 6, 2023 Earthquakes (M7.8-M7.7) in Türkiye with a focus on performance/failure of structures; Workshop on Innovative Seismic Protection of Structural Elements and Structures with Novel Materials, Democritus University of Thrace (DUTH): Civil Engineering Department (DUTH), Xanthi, Greece, Febr 24, 2023. Available online: <https://drive.google.com/file/d/1acQRyrNnHlba87xAdTI3iUHBphFEAbIx/view> (accessed on March 15, 2023).
- Abdelnaby, A. Multiple earthquake effects on degrading reinforced concrete structures. PhD thesis, University of Illinois at Urbana-Champaign, Urbana and Champaign, Illinois, United States, 2012.
- Hatzivassiliou, M.; Hatzigeorgiou, G.D. Seismic sequence effects on three-dimensional reinforced concrete buildings. *Soil Dynamics and Earthquake Engineering* **2015**, *72*, 77–88. doi:10.1016/j.soildyn.2015.02.005.
- Kavvadias, I.E.; Rovithis, P.Z.; Vasiliadis, L.K.; Elenas, A. Effect of the aftershock intensity characteristics on the seismic response of RC frame buildings; 16th European Conference on Earthquake Engineering, , June 18–21, 2018.
- Trevlopoulos, K.; Guéguen, P. Period elongation-based framework for operative assessment of the variation of seismic vulnerability of reinforced concrete buildings during aftershock sequences. *Soil Dynamics and Earthquake Engineering* **2016**, *84*, 224–237. doi:10.1016/j.soildyn.2016.02.009.
- Shokrabadi, M.; Burton, H.V.; Stewart, J.P. Impact of sequential ground motion pairing on mainshock-aftershock structural response and collapse performance assessment. *Journal of Structural Engineering* **2018**, *144*, 04018177. doi:10.1061/(ASCE)ST.1943-541X.0002170.

13. Furtado, A.; Rodrigues, H.; Varum, H.; Arêde, A. Mainshock-aftershock damage assessment of infilled RC structures. *Engineering Structures* **2018**, *175*, 645–660. doi:10.1016/j.engstruct.2018.08.063.
14. Di Sarno, L.; Pugliese, F. Seismic fragility of existing RC buildings with corroded bars under earthquake sequences. *Soil Dynamics and Earthquake Engineering* **2020**, *134*, 106169. doi:10.1016/j.soildyn.2020.106169.
15. Iervolino, I.; Chioccarelli, E.; Suzuki, A. Seismic damage accumulation in multiple mainshock–aftershock sequences. *Earthquake Engineering & Structural Dynamics* **2020**, *49*, 1007–1027. doi:10.1002/eqe.3275.
16. Rajabi, E.; Ghodrati Amiri, G. Behavior factor prediction equations for reinforced concrete frames under critical mainshock-aftershock sequences using artificial neural networks. *Sustainable and Resilient Infrastructure* **2021**, pp. 1–16. doi:10.1080/23789689.2021.1970301.
17. Soureshjani, O.K.; Massumi, A. Seismic behavior of RC moment resisting structures with concrete shear wall under mainshock–aftershock seismic sequences. *Bulletin of Earthquake Engineering* **2022**, *20*, 1087–1114. doi:10.1007/s10518-021-01291-x.
18. Khansefid, A. An investigation of the structural nonlinearity effects on the building seismic risk assessment under mainshock–aftershock sequences in Tehran metro city. *Advances in Structural Engineering* **2021**, p. 13694332211038446. doi:10.1177/13694332211038446.
19. Hu, J.; Wen, W.; Zhai, C.; Pei, S.; Ji, D. Seismic resilience assessment of buildings considering the effects of mainshock and multiple aftershocks. *Journal of Building Engineering* **2023**, p. 106110. doi:10.1016/j.jobe.2023.106110.
20. Askouni, P.K. The Effect of Sequential Excitations on Asymmetrical Reinforced Concrete Low-Rise Framed Structures. *Symmetry* **2023**, *15*. doi:10.3390/sym15050968.
21. Zhao, J.; Ivan, J.N.; DeWolf, J.T. Structural damage detection using artificial neural networks. *Journal of Infrastructure Systems* **1998**, *4*, 93–101. doi:10.1061/(ASCE)1076-0342(1998)4:3(93).
22. Stavroulakis, G.E.; Antes, H. Nondestructive elastostatic identification of unilateral cracks through BEM and neural networks. *Computational Mechanics* **1997**, *20*, 439–451. doi:10.1007/s004660050264.
23. De Lautour, O.R.; Omenzetter, P. Prediction of seismic-induced structural damage using artificial neural networks. *Engineering Structures* **2009**, *31*, 600–606. doi:10.1016/j.engstruct.2008.11.010.
24. Lagaros, N.D.; Papadrakakis, M. Neural network based prediction schemes of the non-linear seismic response of 3D buildings. *Advances in Engineering Software* **2012**, *44*, 92–115. doi:10.1016/j.advengsoft.2011.05.033.
25. Sánchez Silva, M.; García, L. Earthquake damage assessment based on fuzzy logic and neural networks. *Earthquake Spectra* **2001**, *17*, 89–112. doi:10.1193/1.1586168.
26. Alvanitopoulos, P.; Andreadis, I.; Elenas, A. Neuro–fuzzy techniques for the classification of earthquake damages in buildings. *Measurement* **2010**, *43*, 797–809. doi:10.1016/j.measurement.2010.02.011.
27. Vrochidou, E.; Alvanitopoulos, P.F.; Andreadis, I.; Elenas, A. Intelligent systems for structural damage assessment. *Journal of Intelligent Systems* **2018**, *29*, 378–392. doi:10.1515/jisys-2017-0193.
28. Mangalathu, S.; Sun, H.; Nweke, C.C.; Yi, Z.; Burton, H.V. Classifying earthquake damage to buildings using machine learning. *Earthquake Spectra* **2020**, *36*, 183–208. doi:10.1177/8755293019878137.
29. Wang, S.; Cheng, X.; Li, Y.; Song, X.; Guo, R.; Zhang, H.; Liang, Z. Rapid visual simulation of the progressive collapse of regular reinforced concrete frame structures based on machine learning and physics engine. *Engineering Structures* **2023**, *286*, 116129. doi:10.1016/j.engstruct.2023.116129.
30. Wen, W.; Zhang, C.; Zhai, C. Rapid seismic response prediction of RC frames based on deep learning and limited building information. *Engineering Structures* **2022**, *267*, 114638. doi:10.1016/j.engstruct.2022.114638.
31. Zhang, R.; Liu, Y.; Sun, H. Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. *Engineering Structures* **2020**, *215*, 110704. doi:10.1016/j.engstruct.2020.110704.
32. Muradova, A.D.; Stavroulakis, G.E. Physics-informed neural networks for elastic plate problems with bending and Winkler-type contact effects. *Journal of the Serbian Society for Computational Mechanics* **2021**, *15*, 45–54. doi:10.24874/jsscm.2021.15.02.05.
33. Katsikis, D.; Muradova, A.D.; Stavroulakis, G.E. A Gentle Introduction to Physics-Informed Neural Networks, with Applications in Static Rod and Beam Problems. *Journal of Advances in Applied & Computational Mathematics* **2022**, *9*, 103–128. doi:10.15377/2409-5761.2022.09.8.
34. Morfidis, K.; Kostinakis, K. Seismic parameters' combinations for the optimum prediction of the damage state of R/C buildings using neural networks. *Advances in Engineering Software* **2017**, *106*, 1–16. doi:10.1016/j.advengsoft.2017.01.001.

35. Hwang, S.H.; Mangalathu, S.; Shin, J.; Jeon, J.S. Machine learning-based approaches for seismic demand and collapse of ductile reinforced concrete building frames. *Journal of Building Engineering* **2021**, *34*, 101905. doi:10.1016/j.jobe.2020.101905.
36. Kalakonas, P.; Silva, V. Earthquake scenarios for building portfolios using artificial neural networks: Part I—ground motion modelling. *Bulletin of Earthquake Engineering* **2022**, pp. 1–22. doi:10.1007/s10518-022-01598-3.
37. Zhang, H.; Cheng, X.; Li, Y.; He, D.; Du, X. Rapid seismic damage state assessment of RC frames using machine learning methods. *Journal of Building Engineering* **2022**, p. 105797. doi:10.1016/j.jobe.2022.105797.
38. Yuan, X.; Chen, G.; Jiao, P.; Li, L.; Han, J.; Zhang, H. A neural network-based multivariate seismic classifier for simultaneous post-earthquake fragility estimation and damage classification. *Engineering Structures* **2022**, *255*, 113918. doi:10.1016/j.engstruct.2022.113918.
39. Kazemi, P.; Ghisi, A.; Mariani, S. Classification of the Structural Behavior of Tall Buildings with a Diagrid Structure: A Machine Learning-Based Approach. *Algorithms* **2022**, *15*, 349. doi:10.3390/a151100349.
40. Karbassi, A.; Mohebi, B.; Rezaee, S.; Lestuzzi, P. Damage prediction for regular reinforced concrete buildings using the decision tree algorithm. *Computers & Structures* **2014**, *130*, 46–56. doi:10.1016/j.compstruc.2013.10.006.
41. Ghiasi, R.; Torkzadeh, P.; Noori, M. A machine-learning approach for structural damage detection using least square support vector machine based on a new combinational kernel function. *Structural Health Monitoring* **2016**, *15*, 302–316. doi:10.1177/1475921716639587.
42. Thaler, D.; Stoffel, M.; Markert, B.; Bamer, F. Machine-learning-enhanced tail end prediction of structural response statistics in earthquake engineering. *Earthquake Engineering & Structural Dynamics* **2021**. doi:10.1002/eqe.3432.
43. Chen, S.Z.; Feng, D.C.; Wang, W.J.; Tacioglu, E. Probabilistic Machine-Learning Methods for Performance Prediction of Structure and Infrastructures through Natural Gradient Boosting. *Journal of Structural Engineering* **2022**, *148*, 04022096. doi:10.1061/(ASCE)ST.1943-541X.0003401.
44. Jia, D.W.; Wu, Z.Y. Structural probabilistic seismic risk analysis and damage prediction based on artificial neural network. *Structures*, **2022**, Vol. 41, pp. 982–996. doi:10.1016/j.istruc.2022.05.056.
45. Kiani, J.; Camp, C.; Pezeshk, S. On the application of machine learning techniques to derive seismic fragility curves. *Computers & Structures* **2019**, *218*, 108–122. doi:10.1016/j.compstruc.2019.03.004.
46. Kazemi, F.; Asgarkhani, N.; Jankowski, R. Machine learning-based seismic fragility and seismic vulnerability assessment of reinforced concrete structures. *Soil Dynamics and Earthquake Engineering* **2023**, *166*, 107761. doi:10.1016/j.soildyn.2023.107761.
47. Dabiri, H.; Faramarzi, A.; Dall'Asta, A.; Tondi, E.; Micozzi, F. A machine learning-based analysis for predicting fragility curve parameters of buildings. *Journal of Building Engineering* **2022**, p. 105367. doi:10.1016/j.jobe.2022.105367.
48. Yang, J.; Zhuang, H.; Zhang, G.; Tang, B.; Xu, C. Seismic performance and fragility of two-story and three-span underground structures using a random forest model and a new damage description method. *Tunnelling and Underground Space Technology* **2023**, *135*, 104980. doi:10.1016/j.tust.2022.104980.
49. De-Miguel-Rodríguez, J.; Morales-Esteban, A.; Requena-García-Cruz, M.; Zapico-Blanco, B.; Segovia-Verjel, M.; Romero-Sánchez, E.; Carvalho-Estêvão, J.M. Fast seismic assessment of built urban areas with the accuracy of mechanical methods using a feedforward neural network. *Sustainability* **2022**, *14*, 5274. doi:10.3390/su14095274.
50. Morfidis, K.; Stefanidou, S.; Markogiannaki, O. A Rapid Seismic Damage Assessment (RASDA) Tool for RC Buildings Based on an Artificial Intelligence Algorithm. *Applied Sciences* **2023**, *13*. doi:10.3390/app13085100.
51. Lazaridis, P.C.; Kavvadias, I.E.; Demertzis, K.; Iliadis, L.; Papaleonidas, A.; Vasiliadis, L.K.; Elenas, A. Structural Damage Prediction Under Seismic Sequence Using Neural Networks; 8th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPdyn 2021), ECCOMAS: Athens, Greece, June 28–30, 2021. doi:10.7712/120121.8750.18752.
52. Lazaridis, P.C.; Kavvadias, I.E.; Demertzis, K.; Iliadis, L.; Vasiliadis, L.K. Structural Damage Prediction of a Reinforced Concrete Frame under Single and Multiple Seismic Events Using Machine Learning Algorithms. *Applied Sciences* **2022**, *12*, 3845. doi:10.3390/app12083845.

53. Sun, H.; Burton, H.V.; Huang, H. Machine Learning Applications for Building Structural Design and Performance Assessment: State-of-the-Art Review. *Journal of Building Engineering* **2020**, p. 101816. doi:10.1016/j.jobe.2020.101816.
54. Xie, Y.; Ebad Sichani, M.; Padgett, J.E.; DesRoches, R. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra* **2020**, p. 8755293020919419. doi:10.1177/8755293020919419.
55. Harirchian, E.; Hosseini, S.E.A.; Jadhav, K.; Kumari, V.; Rasulzade, S.; Işık, E.; Wasif, M.; Lahmer, T. A Review on Application of Soft Computing Techniques for the Rapid Visual Safety Evaluation and Damage Classification of Existing Buildings. *Journal of Building Engineering* **2021**, p. 102536. doi:10.1016/j.jobe.2021.102536.
56. Sony, S.; Dunphy, K.; Sadhu, A.; Capretz, M. A systematic review of convolutional neural network-based structural condition assessment techniques. *Engineering Structures* **2021**, *226*, 111347. doi:10.1016/j.engstruct.2020.111347.
57. Tapeh, A.T.G.; Naser, M. Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices. *Archives of Computational Methods in Engineering* **2022**, pp. 1–45. doi:10.1007/s11831-022-09793-w.
58. Thai, H.T. Machine learning for structural engineering: A state-of-the-art review. *Structures* **2022**, *38*, 448–491. doi:10.1016/j.istruc.2022.02.003.
59. You, Z. Intelligent construction: Unlocking opportunities for the digital transformation of China's construction industry. *Engineering, Construction and Architectural Management* **2022**. doi:10.1108/ECAM-08-2022-0706.
60. Málaga-Chuquitaype, C. Machine learning in structural design: An opinionated review. *Frontiers in Built Environment* **2022**, *8*, 6. doi:10.3389/fbuil.2022.815717.
61. Demertzis, K.; Demertzis, S.; Iliadis, L. A Selective Survey Review of Computational Intelligence Applications in the Primary Subdomains of Civil Engineering Specializations. *Applied Sciences* **2023**, *13*. doi:10.3390/app13063380.
62. Wang, C.; Song, L.H.; Yuan, Z.; Fan, J.S. State-of-the-Art AI-Based Computational Analysis in Civil Engineering. *Journal of Industrial Information Integration* **2023**, p. 100470. doi:10.1016/j.jii.2023.100470.
63. Molnar, C. *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*, 2nd ed.; Independently published, 2023. Available online: <https://christophm.github.io/interpretable-ml-book> (accessed on March 15, 2023).
64. Mangalathu, S.; Karthikeyan, K.; Feng, D.C.; Jeon, J.S. Machine-learning interpretability techniques for seismic performance assessment of infrastructure systems. *Engineering Structures* **2022**, *250*, 112883. doi:10.1016/j.engstruct.2021.112883.
65. Wakjira, T.G.; Ibrahim, M.; Ebead, U.; Alam, M.S. Explainable machine learning model and reliability analysis for flexural capacity prediction of RC beams strengthened in flexure with FRCC. *Engineering Structures* **2022**, *255*, 113903. doi:10.1016/j.engstruct.2022.113903.
66. Junda, E.; Málaga-Chuquitaype, C.; Chawgien, K. Interpretable machine learning models for the estimation of seismic drifts in CLT buildings. *Journal of Building Engineering* **2023**, p. 106365. doi:10.1016/j.jobe.2023.106365.
67. Demertzis, K.; Kostinakis, K.; Morfidis, K.; Iliadis, L. An interpretable machine learning method for the prediction of R/C buildings' seismic response. *Journal of Building Engineering* **2022**, p. 105493. doi:10.1016/j.jobe.2022.105493.
68. Park, Y.J.; Ang, A.H.S. Mechanistic seismic damage model for reinforced concrete. *Journal of Structural Engineering* **1985**, *111*, 722–739. doi:10.1061/(ASCE)0733-9445(1985)111:4(722).
69. Elenas, A. Interdependency between seismic acceleration parameters and the behaviour of structures. *Soil Dynamics and Earthquake Engineering* **1997**, *16*, 317–322. doi:10.1016/S0267-7261(97)00005-5.
70. Elenas, A.; Liolios, A.; Vasiliadis, L. Correlation factors between seismic acceleration parameters and damage indicators of reinforced concrete structures. *Balkema Publishers, Structural Dynamics: EURO DYN'99* **1999**, *2*, 1095–1100.
71. Elenas, A. Correlation between seismic acceleration parameters and overall structural damage indices of buildings. *Soil Dynamics and Earthquake Engineering* **2000**, *20*, 93–100. doi:10.1016/S0267-7261(00)00041-5.

72. Vrochidou, E.; Alvanitopoulos, P.; Andreadis, I.; Elenas, A. Correlation between seismic intensity parameters of HHT-based synthetic seismic accelerograms and damage indices of buildings; IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer: Halkidiki, Greece, Sep 27–30, 2012; pp. 425–434. doi:10.1007/978-3-642-33409-2_44.
73. Fontara, I.; Athanatopoulou, A.; Avramidis, I. Correlation between advanced, structure-specific ground motion intensity measures and damage indices; 15th World Conference on Earthquake Engineering, Curran Associates, Inc.: Lisbon, Portugal, Sep 24–28, 2012; pp. 24–28.
74. Van Cao, V.; Ronagh, H.R. Correlation between seismic parameters of far-fault motions and damage indices of low-rise reinforced concrete frames. *Soil Dynamics and Earthquake Engineering* **2014**, *66*, 102–112. doi:10.1016/j.soildyn.2014.06.020.
75. Kostinakis, K.; Athanatopoulou, A.; Morfidis, K. Correlation between ground motion intensity measures and seismic damage of 3D R/C buildings. *Engineering Structures* **2015**, *82*, 151–167. doi:10.1016/j.engstruct.2014.10.035.
76. Lazaridis, P.C.; Kavvadias, I.E.; Vasiliadis, L.K. Correlation between Seismic Parameters and Damage Indices of Reinforced Concrete Structures; 4th Panhellenic Conference on Earthquake Engineering and Engineering Seismology, Hellenic Association for Earthquake Engineering (HAEE / ETAM) and Technical Chamber of Greece: Athens, Greece, Sep 5-7, 2019.
77. Palanci, M.; Senel, S.M. Correlation of earthquake intensity measures and spectral displacement demands in building type structures. *Soil Dynamics and Earthquake Engineering* **2019**, *121*, 306–326. doi:10.1016/j.soildyn.2019.03.023.
78. Kramer, S.L. *Geotechnical earthquake engineering*; Prentice Hall: Upper Saddle River, New Jersey 07458, 1996.
79. Arias, A. A measure of earthquake intensity. Seismic Design for Nuclear Power Plants. *Massachusetts Institute of Technology, Cambridge, MA, USA* **1970**.
80. Araya, R.; Saragoni, G.R. Earthquake accelerogram destructiveness potential factor; 8th World Conference on Earthquake Engineering, Earthquake Engineering Research Institute (EERI): San Francisco, California, USA, July, 1985; Vol. 11, pp. 835–843.
81. Trifunac, M.D.; Brady, A.G. A study on the duration of strong earthquake ground motion. *Bulletin of the Seismological Society of America* **1975**, *65*, 581–626.
82. Reinoso, E.; Ordaz, M.; Guerrero, R. Influence of strong ground-motion duration in seismic design of structures; 12th World Conference on Earthquake Engineering, New Zealand Society for Earthquake Engineering, Upper Hutt, N.Z., 2000: Auckland, New Zealand, Jan 30–Febr 4, 2000; Vol. 1151.
83. Bolt, B.A. Duration of strong ground motion; 5th World Conference on Earthquake Engineering, International Association for Earthquake Engineering (IAEE): Rome, Italy, June 25–29, 1973; Vol. 292, pp. 25–29.
84. Fajfar, P.; Vidic, T.; Fischinger, M. A measure of earthquake motion capacity to damage medium-period structures. *Soil Dynamics and Earthquake Engineering* **1990**, *9*, 236–242. doi:10.1016/S0267-7261(05)80002-8.
85. Riddell, R.; Garcia, J.E. Hysteretic energy spectrum and damage control. *Earthquake Engineering & Structural Dynamics* **2001**, *30*, 1791–1816. doi:10.1002/eqe.93.
86. Housner, G.W. Spectrum intensities of strong-motion earthquakes; Symposium on Earthquake and Blast Effects on Structures, Earthquake Engineering Research Institute: Los Angeles, California, USA, June 25–29, 1952.
87. Van Rossum, G. Python reference manual, 1995.
88. Harris, C.R.; Millman, K.J.; Van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; Kern, R.; Picus, M.; Hoyer, S.; Van Kerkwijk, M.H.; Brett, M.; Haldane, A.; Fernández del Río, J.; Wiebe, M.; Peterson, P.; Gérard Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Hameer Abbasi, C.; Gohlke, C.; Oliphant, T.E. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. doi:10.1038/s41586-020-2649-2.
89. Papazafeiropoulos, G.; Plevris, V. OpenSeismoMatlab: A new open-source software for strong ground motion data processing. *Heliyon* **2018**, *4*, e00784. doi:10.1016/j.heliyon.2018.e00784.
90. Husid, R. Características de terremotos. Análisis general. *Revista IDIEM* **1969**, *8*, ág–21.
91. Kunnath, S.K.; Reinhorn, A.M.; Lobo, R. IDARC Version 3.0: A program for the inelastic damage analysis of reinforced concrete structures. Technical report, US National Center for Earthquake Engineering Research (NCEER), University at Buffalo 212 Ketter Hall Buffalo, NY 14260-4300, 1992.

92. Karabini, M.A.; Karabinis, A.J.; Karayannis, C.G. Seismic characteristics of a II-shaped 4-story RC structure with open ground floor. *Earthquakes and Structures* **2022**, *22*, 345. doi:10.12989/eas.2022.22.4.345.
93. Park, Y.J.; Ang, A.H.; Wen, Y.K. Damage-limiting aseismic design of buildings. *Earthquake Spectra* **1987**, *3*, 1–26. doi:10.1193/1.1585416.
94. Luzi, L.; Lanzano, G.; Felicetta, C.; D'Amico, M.; Russo, E.; Sgobba, S.; Pacor, F.; ORFEUS Working Group 5. Engineering Strong Motion Database (ESM) (Version 2.0). *Istituto Nazionale di Geofisica e Vulcanologia (INGV), Rome, Italy* **2020**. doi:10.13127/ESM.2.
95. Goulet, C.A.; Kishida, T.; Ancheta, T.D.; Cramer, C.H.; Darragh, R.B.; Silva, W.J.; Hashash, Y.M.; Harmon, J.; Stewart, J.P.; Wooddell, K.E.; others. PEER NGA-East Database. Technical report, Pacific Earthquake Engineering Research Center Berkeley, CA, 2014.
96. Hatzigeorgiou, G.D.; Liolios, A.A. Nonlinear behaviour of RC frames under repeated strong ground motions. *Soil Dynamics and Earthquake Engineering* **2010**, *30*, 1010–1025. doi:10.1016/j.soildyn.2010.04.013.
97. CEN. Eurocode 8: Design of structures for earthquake resistance-part 1: General rules, seismic actions and rules for buildings. *Brussels: European Committee for Standardization* **2005**.
98. Chalioris, C.E. Analytical model for the torsional behaviour of reinforced concrete beams retrofitted with FRP materials. *Engineering Structures* **2007**, *29*, 3263–3276. doi:10.1016/j.engstruct.2007.09.009.
99. Bantilas, K.E.; Kavvadias, I.E.; Vasiliadis, L.K. Capacity spectrum method based on inelastic spectra for high viscous damped buildings. *Earthquakes and Structures* **2017**, *13*, 337–351. doi:10.12989/eas.2017.13.4.337.
100. Anagnostou, E.; Rousakis, T.C.; Karabinis, A.I. Seismic retrofitting of damaged RC columns with lap-spliced bars using FRP sheets. *Composites Part B: Engineering* **2019**, *166*, 598–612. doi:10.1016/j.compositesb.2019.02.018.
101. Thomoglou, A.K.; Rousakis, T.C.; Achillopoulou, D.V.; Karabinis, A.I. Ultimate shear strength prediction model for unreinforced masonry retrofitted externally with textile reinforced mortar. *Earthquakes and Structures* **2020**, *19*, 411–425. doi:10.12989/eas.2020.19.6.411.
102. Rousakis, T.; Anagnostou, E.; Fanaradelli, T. Advanced Composite Retrofit of RC Columns and Frames with Prior Damages—Pseudodynamic Finite Element Analyses and Design Approaches. *Fibers* **2021**, *9*, 56. doi:10.3390/fib9090056.
103. Karabini, M.; Rousakis, T.; Goliass, E.; Karayannis, C. Seismic Tests of Full Scale Reinforced Concrete T Joints with Light External Continuous Composite Rope Strengthening—Joint Deterioration and Failure Assessment. *Materials* **2023**, *16*, 2718. doi:10.3390/ma16072718.
104. Valles, R.; Reinhorn, A.M.; Kunnath, S.K.; Li, C.; Madan, A. IDARC2D version 4.0: A computer program for the inelastic damage analysis of buildings. Technical report, US National Center for Earthquake Engineering Research (NCEER), University at Buffalo 212 Ketter Hall Buffalo, NY 14260-4300, 1996.
105. Park, Y.J.; Reinhorn, A.M.; Kunnath, S.K. *IDARC: Inelastic Damage Analysis of Reinforced Concrete Frame-Shear-Wall Structures*; National Center for Earthquake Engineering Research Buffalo: University at Buffalo 212 Ketter Hall Buffalo, NY 14260-4300, 1987.
106. Eaton, J.W. GNU Octave and reproducible research. *Journal of Process Control* **2012**, *22*, 1433–1438. doi:10.1016/j.jprocont.2012.04.006.
107. Eaton, J.W.; Bateman, D.; Hauberg, S.; Wehbring, R. *GNU Octave version 6.1.0 manual: A high-level interactive language for numerical computations*, 2020.
108. Galton, F. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland* **1886**, *15*, 246–263. doi:10.2307/2841583.
109. Stanton, J.M. Galton, Pearson, and the peas: A brief history of linear regression for statistics instructors. *Journal of Statistics Education* **2001**, *9*. doi:10.1080/10691898.2001.11910537.
110. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **1996**, *58*, 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.
111. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67. doi:10.1080/00401706.1970.10488634.
112. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: Series B (statistical methodology)* **2005**, *67*, 301–320. doi:10.1111/j.1467-9868.2005.00503.x.
113. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* **1992**, *46*, 175–185. doi:10.1080/00031305.1992.10475879.

114. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and regression trees*; Routledge: United Kingdom, 2017. doi:10.1201/9781315139470.
115. Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32. doi:10.1023/{A}:1010933404324.
116. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Annals of statistics* **2001**, pp. 1189–1232.
117. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Machine learning* **2006**, *63*, 3–42. doi:10.1007/s10994-006-6226-1.
118. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* **1997**, *55*, 119–139. doi:10.1006/jcss.1997.1504.
119. Drucker, H. Improving regressors using boosting techniques; 14th International Conference on Machine Learning (ICML), Association for Computing Machinery: Nashville, TN, USA, July 8–12, 1997; Vol. 97, pp. 107–115.
120. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree; Advances in Neural Information Processing Systems (NIPS), Curran Associates, Inc.: Long Beach, California, USA, Dec 4–9, 2017; Vol. 30.
121. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Gradient boosting with categorical features support; Workshop on ML Systems at NIPS, Thirty-first Conference on Neural Information Processing Systems: Long Beach, California, USA, Dec 8, 2017.
122. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features; Advances in Neural Information Processing Systems (NeurIPS), Curran Associates, Inc.: Montréal, Quebec, Canada, Dec 3–8, 2018; Vol. 31.
123. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System; 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery: San Francisco, California, USA, Aug 13–17, 2016; pp. 785–794. doi:10.1145/2939672.2939785.
124. Duan, T.; Anand, A.; Ding, D.Y.; Thai, K.K.; Basu, S.; Ng, A.; Schuler, A. NGBoost: Natural Gradient Boosting for Probabilistic Prediction; 37th International Conference on Machine Learning (ICML), PMLR: Virtual, Jul 13–18, 2020; Vol. 119, *Proceedings of Machine Learning Research*, pp. 2690–2700.
125. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* **1943**, *5*, 115–133.
126. Rojas, R. *Neural Networks: A Systematic Introduction*; Springer Science & Business Media: Berlin, Heidelberg, 1996. doi:10.1007/978-3-642-61068-4.
127. Minsky, M.; Papert, S.A. *Perceptrons: An introduction to computational geometry*; MIT Press: Cambridge, MA, USA, 2017.
128. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016. Available online: <http://www.deeplearningbook.org> (accessed on March 15, 2023).
129. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536.
130. LeCun, Y.A.; Bottou, L.; Orr, G.B.; Müller, K.R. Efficient backprop; Springer, 2012; pp. 9–48. doi:10.1007/978-3-642-35289-8_3.
131. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization; Advances in Neural Information Processing Systems (NIPS), Curran Associates, Inc.: Granada, Spain, EU, Dec 12–15, 2011; Vol. 24.
132. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **2012**, *13*.
133. Picard, R.R.; Cook, R.D. Cross-validation of regression models. *Journal of the American Statistical Association* **1984**, *79*, 575–583. doi:10.1080/01621459.1984.10478083.
134. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
135. Hastie, T.; Friedman, J.H.; Tibshirani, R. *The elements of statistical learning: Data mining, inference, and prediction*; Vol. 2, Springer: New York, NY, USA, 2009. doi:10.1007/978-0-387-21606-5.

136. Apley, D.W.; Zhu, J. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **2020**, *82*, 1059–1086. doi:10.1111/rssb.12377.
137. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier; 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery: San Francisco, California, USA, Aug 13–17, 2016; KDD '16, p. 1135–1144. doi:10.1145/2939672.2939778.
138. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions; 31st Conference on Neural Information Processing Systems (NIPS 2017), Curran Associates, Inc.: Long Beach, CA, USA, Dec 4–9, 2017; pp. 4765–4774.
139. Shapley, L.S. A value for n-person games. *Annals of Mathematical Studies* **1953**, *28*, 307–317.
140. Zhou, Z.; Yu, X.; Lu, D. Identifying Optimal Intensity Measures for Predicting Damage Potential of Mainshock–Aftershock Sequences. *Applied Sciences* **2020**, *10*, 6795. doi:10.3390/app10196795.
141. Amiri, S.; Di Sarno, L.; Garakaninezhad, A. Correlation between non-spectral and cumulative-based ground motion intensity measures and demands of structures under mainshock-aftershock seismic sequences considering the effects of incident angles. *Structures* **2022**, *46*, 1209–1223. doi:10.1016/j.istruc.2022.10.076.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.