

## Article

# Rapid Visual Screening Feature Importance for Seismic Vulnerability Ranking via Machine Learning and SHAP Values

Ioannis Karampinis <sup>1</sup>, Lazaros Iliadis <sup>1,\*</sup> and Athanasios Karabinis <sup>2</sup>

<sup>1</sup> Lab of Mathematics and Informatics (ISCE), Department of Civil Engineering, Democritus University of Thrace, 67100 Xanthi, Greece; ikarampi@civil.duth.gr

<sup>2</sup> Lab of Reinforced Concrete and Seismic Design, Department of Civil Engineering, Democritus University of Thrace, 67100 Xanthi, Greece; akarabin@civil.duth.gr

\* Correspondence: liliadis@civil.duth.gr

**Abstract:** Structures inevitably suffer damage after an earthquake, with severity ranging from minimal damage of nonstructural elements to partial or even total collapse, possibly with loss of human lives. Thus, it is essential for engineers to understand the crucial factors that drive a structure towards suffering higher degrees of damage in order for preventative measures to be taken. In the present study, we focus on three well-known damage thresholds: the Collapse Limit State, Ultimate Limit State, and Serviceability Limit State. We analyze the features obtained via Rapid Visual Screening to determine whether or not a given structure crosses these thresholds. To this end, we use machine learning to perform binary classification for each damage threshold, and use explainability to quantify the effect of each parameter via SHAP values (SHapley Additive exPlanations). The quantitative results that we obtain demonstrate the potential applicability of ML methods for recalibrating the computation of structural vulnerability indices using data from recent earthquakes.

**Keywords:** rapid visual screening; explainable AI; feature importance; SHAP



**Citation:** Karampinis, I.; Iliadis, L.; Karabinis, A. Rapid Visual Screening Feature Importance for Seismic Vulnerability Ranking via Machine Learning and SHAP Values. *Appl. Sci.* **2024**, *14*, 2609. <https://doi.org/10.3390/app14062609>

Academic Editors: Giuseppe Lacidogna, Bijan Samali and Amir M. Yousefi

Received: 12 February 2024

Revised: 22 February 2024

Accepted: 19 March 2024

Published: 20 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

During the last decades, due to the large amount of existing building stock, engineering focus has shifted from analyzing and designing new structures to maintaining preexisting buildings to modern standards of safety and serviceability [1]. As is well known, the results of an earthquake can be catastrophic to society in terms of loss of human lives and require large monetary reparations, with examples including the Turkey (Izmit) 1999, Athens 1999, Pakistan 2005, and Turkey 2023 earthquakes.

Governments and authorities can take preemptive measures to mitigate these effects; however, due to obvious limitations in resources and manpower, it is not possible to do so for all existing buildings, especially in large urban areas. Thus, most countries have introduced multi-stage procedures to assess and evaluate the total potential consequences and losses from an earthquake, and thereby identify the most critical structures where allocation of further resources should be prioritized.

As a first step in these methods a Rapid Visual Screening Procedure (RVSP) [2] is usually performed, wherein experts quickly inspect buildings and identify key structural characteristics that affect the overall seismic behaviour. For example, this could include whether or not the structure has short columns or soft storeys, the presence of neighboring buildings that could result in pounding effects, irregularities in the horizontal or vertical plan of the building, and others [3,4]. Subsequently, these obtained characteristics are weighted to compute a seismic vulnerability index which is used to rank the structures according to their expected degree of damage [5]. Finally, the most vulnerable structures that have been identified from the aforementioned steps are subjected to more accurate analytical methods such as step-by-step dynamic analysis. These methods take into account other structural characteristics, such as the design of structural reinforcement and quality

of concrete, and yield an accurate assessment of the seismic vulnerability of the structures under consideration. In turn, this allows for the identification of any potentially required preemptive measures to be applied. However, they are prohibitively costly and time consuming to apply to every structure in the population.

In USA, the Federal Emergency Management Agency (FEMA) first introduced such an RVSP [2] in 1988, which has since been modified to include more structural features that affect the overall seismic performance [6]. Countries with high seismic activity, such as Japan, Italy, Canada, India, and Greece, have derived similar pre-earthquake assessments adapted to the characteristics of their respective building stocks. The success of RVSP in screening candidate structures for further analysis heavily depends on accurate calibration of the weights of the structural characteristics. Thus, past researchers have used data from major recorded earthquakes in conjunction with engineering expertise for this task [7,8]. Similarly, for masonry buildings, both index-based [9] and physics-based [10] structural vulnerability assessment studies have been conducted. The effect of the structural parameters of this type of building on their structural vulnerability has been studied as well [11,12].

On the other hand, recent years have seen an increase in the use of Machine Learning (ML) methods for the task of predicting the degree of damage of reinforced concrete structures. Classification techniques have been previously employed to classify structures into predicted damage categories. Harichian et al. [13] employed Support Vector Machines, which they calibrated on dataset of earthquakes in four different countries. Sajan et al. [14] employed a variety of models, including Decision Trees, Random Forests, XGBoost, and Logistic Regression. Similarly, regression methods have been employed for this task. Among others, Luo and Paal [15] and Kazemi et al. [16] used ML methods to predict the interstorey drift, which can be used as a damage index.

Even though machine learning methods are powerful, they often lack the desired interpretability. The path that a Decision Tree follows to reach its predictions can be readily visualized; however, the same does not hold for more complex ML models. Thus, explainability techniques and models have been employed in ML [17] in order to analyze how these models weigh their input parameters when making a decision, thereby increasing the reliability of their predictions. Among others, Mangalathu et al. [18] recently employed Shapley additive explanations (SHAP) [19] to quantify the effect of each input parameter on damage predictions of bridges in California. Sajan et al. [14] performed multiclass classification to predict the damage category of structures and binary classification to predict whether the damage was recoverable or reconstruction was needed. Subsequently, they employed SHAP values to identify 19 of the top 20 most important features for both tasks. However, the features they employed significantly deviated from those in the RVS procedure, and lack many of the features employed in the present study.

The features employed in the present study have been used previously [20,21]; however, there is no consensus on the magnitude of the effect that each feature has on the vulnerability ranking, with different researchers and different seismic codes employing different values. In this paper, we implement explainable machine learning techniques and SHAP values to analyze features' contribution to the relative classification of structures in the respective damage categories. To the best of our knowledge, the novel contribution of the introduced approach is that it does not attempt to directly predict the damage category. Instead, it considers the well known thresholds of the Serviceability Limit State (SLS), Ultimate Limit State (ULS), and Collapse Limit State (CLS) to distinguish structures that not only surpass the ULS threshold but suffer partial or total collapse which could potentially lead to loss of human life. Moreover, machine learning is used to develop binary classification models capable of distinguishing between adjacent damage categories.

The benefit of this modeling research effort in comparison with the previously established literature is twofold. On the one hand, the obtained binary classifiers have significantly improved accuracy compared to previous models. This higher accuracy enhances the reliability of the extracted feature importance coefficients, which is the main focus of the present study. On the other hand, the binary classification approach allows

us to examine each of the damage thresholds separately. This allows us to answer the following questions: What are the deciding factors that lead a structure which would have otherwise suffered minimal to no damage to cross the serviceability limit threshold? If a structure does cross the serviceability threshold, what factors prevent it from crossing the ultimate limit state threshold as well? Finally, if it does cross the ULS threshold, what factors prevent it from ultimately collapsing?

## 2. Materials and Methods

### 2.1. Dataset Description

The dataset used in the present study is a sample consisting of 457 structures obtained after the 1999 Athens Earthquake via Rapid Visual Screening (RVS) [20]. The selected structures suffered damage across the spectrum, ranging from very low or minimal damage to structures that partially or completely collapsed during the earthquake. The dataset was drawn from different geographical region; thus, the local conditions varied across the sample. In [20], the authors took steps to mitigate the effect on the study of potential biases due to local effects. When sampling from a specific building block, they sampled structures across the entire damage spectrum. This mitigated the effect of the location of the structure on its seismic damage, as structures in the same building block had the same local conditions. The degree of damage was labeled using four categories:

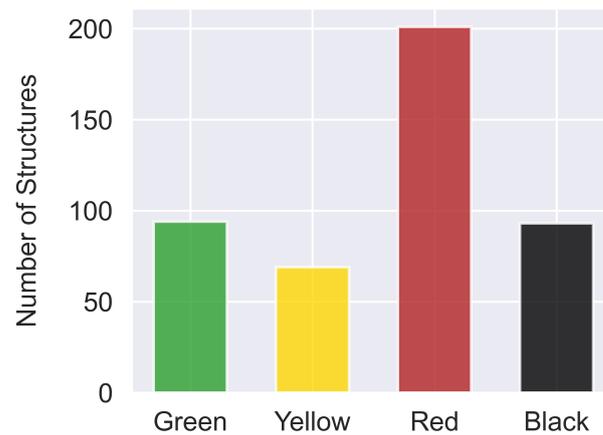
- Black: Structures that suffered total or partial collapse during the earthquake, potentially leading to loss of human life.
- Red: Structures with significant damage to their structural members.
- Yellow: Structures with moderate damage to the structural members, potentially including extended damage to nonstructural elements.
- Green: Structures that suffered very little or no damage.

An example of the application of the RVS procedure can be seen in Figure 1, courtesy of [20].



**Figure 1.** Application of Rapid Visual Screening in a specific area. Samples of structures across the damage spectrum were drawn to mitigate local effects. Image courtesy of [20].

The distribution of structures across the above damage categories is shown in Figure 2.



**Figure 2.** Distribution of structures across the damage spectrum.

For each structure, a set of attributes were documented, specifically:

1. Free ground level (Pilotis), soft storeys and/or short columns: In general, this attribute pertains to structures wherein a storey has significantly less structural rigidity than the rest. For example, this can manifest on the ground floor (pilotis) when it has greater height than the typical structure storey, or when the wall fillings do not cover the whole height of a storey, effectively reducing the active height of the adjacent columns.
2. Wall fillings regularity: This indicates whether the infill walls are of sufficient thickness and with few openings. The presence of such wall fillings is beneficial to the structure's overall seismic response, as during an earthquake they act as diagonal struts that support the surrounding frames.
3. Absence of design seismic codes: In Greece, this pertains to pre-1960 structures which were not designed following a dedicated seismic code.
4. Poor condition: Very high or non-uniform ground sinking, concrete with aggregate segregation or erosion, or corrosion in the reinforcement bars are examples of maintenance-related factors that can reduce the seismic capacity of a building.
5. Previous damage: This pertains to structures which had suffered previous earthquake damages that was not adequately repaired. Although this is distinct feature from "poor condition", it causes a similar reduction in the nominal seismic capacity of the building.
6. Significant height: This describes structures with five or more storeys.
7. Irregularity in height: This describes structures with a discontinuity in the vertical path of the loads.
8. Irregularity in plan: This pertains to structures with floor plans that significantly deviate from a rectangular shape, e.g., floor plans with highly acute angles in their outer walls or with E, Z, or H-shapes. Irregularity in height, plan, or both can cause excess seismic overload on the building.
9. Torsion: This affects structures with high horizontal eccentricity, which are subjected to torsion during the earthquake.
10. Pounding: If adjacent buildings do not have a sufficient gap between them, and especially if they have different heights, then the floor slabs of one building can ram into the columns of the other.
11. Heavy nonstructural elements: These elements can potentially create eccentricities if they are displaced during an earthquake, leading to additional torsion. This is because even though these are nonstructural elements, they can often contribute to the total mass and horizontal stiffness of the structure.
12. Foundation Soil: The Greek Code for Seismic Resistant Structures–EAK 200 [3] classifies soils into categories A, B, C, D, and X. Class A refers to rock or semi-rock formations

extending in wide area and large depth. Class B refers to strongly weathered rocks or soils mechanically equivalent to granular materials. Classes C and D refer to granular materials and soft clay, respectively, while class X refers to loose fine-grained silt [3]. In [20], as well as in the present study, soils in EAK category A are classified as S1, while those in category B are classified as S2; soils in EAK categories C, D, and X were not encountered.

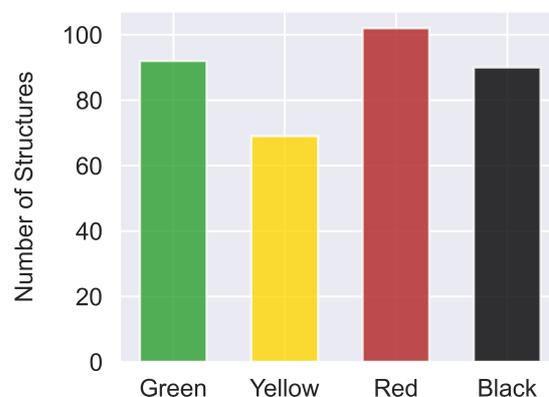
13. The design Seismic Code: This feature describes the seismic code(s) that the structures adhered to at the time of their design. Specifically, structures that were built before 1984 are classified as RC1, buildings constructed between 1985 and 1994 are labeled RC2, and buildings constructed after 1995 are labeled RC3, as the Greek state introduced updated seismic codes at these milestones.

Note that most of the above features are binary, i.e., the dataset provides a Yes/No statement about whether or not the structure displayed the relevant feature. We transformed these to Boolean values, i.e., {Yes, No}  $\rightarrow$  {0, 1}. The design seismic code was transformed to an integer value, i.e., {RC1, RC2, RC3}  $\rightarrow$  {1, 2, 3}. Finally, in 452 out of the 457 total documents, the authors of [20] noted the exact number of storeys instead of whether or not this was  $\geq 5$ . As this was deemed more informative, we opted to disregard these structures (1.09% of the sample) and use this feature instead.

## 2.2. Data Preprocessing

The core of the designed and employed modeling effort lies in the development of a Machine Learning (ML) model for binary classification  $f : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \{-1, +1\}$  that, given a pair of structures  $(s_i, s_j)$  with corresponding feature vectors  $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n$ , is capable of predicting whether  $s_j$  should rank higher than  $s_i$  or vice versa [22].

However, it can be readily observed from Figure 2 that the “Red” label heavily dominates the sampled dataset. This so-called “class imbalance problem” has significant adverse effects on any machine learning algorithm [23–26]. It leads the model to be skewed towards the majority class, creating bias and rendering the algorithm unable to adapt to the features of the minority classes [23,24]. This imbalance can be treated by undersampling the majority class, and there are numerous methods in the literature in order to do so [27–29]. These methods include randomly selecting a subset of the samples in the majority class [30,31], or using model-based methods such as NearMiss, Tomek Links, or Edited Nearest Neighbours [27–29]. NearMiss-2 was found to perform the best, and is used in the sequelae. We undersampled the majority class by a factor of 50% in order to achieve a relative class balance, which, as mentioned, is crucial to the performance of machine learning algorithms. The distribution of structures across the above damage categories after undersampling is shown in Figure 3.



**Figure 3.** Distribution of structures across the damage spectrum after undersampling.

Next, in order to represent the pair  $(\mathbf{x}_i, \mathbf{x}_j)$  using a single feature vector  $\mathbf{x}^{new}$  as input for the machine learning model, we considered the pairwise transformation  $T : \mathbb{R}^n \times \mathbb{R}^n \rightarrow$

$\mathbb{R}^n$  with  $T(x_i, x_j) = x_j - x_i$ . Other pairwise transformations can be employed, e.g.,  $T_2 : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{2n}$ , with  $T_2(x_i, x_j) = [x_i; x_j]$ , i.e., appending  $x_j$  to  $x_i$  [32]. However, the transformation employed in the present study has the advantage of a more natural interpretation, which is the goal of this study. For a example, a value of 2 storeys in the transformed dataset indicates that structure  $s_j$  has two more storeys than  $s_i$ . Similarly, a transformed value of  $-1$  for the “pounding” attribute indicates that  $s_i$  suffered from pounding while  $s_j$  did not.

A similar transformation was applied to the labels of the damage categories. To this end, the labels were first ranked in ascending order, i.e., Green, Yellow, Red, Black  $\rightarrow \{1, 2, 3, 4\}$ . Then, for a pair of structures  $(s_i, s_j)$  with  $(y_i, y_j) \in \{1, 2, 3, 4\}^2$  and  $y_i \neq y_j$ , the transformed target variable was  $y^{new} = \text{sign}(y_j - y_i)$ , where  $\text{sign}$  denotes the sign function. Thus, for example, a transformed variable of  $-1$  indicates that  $s_j$  suffered more severe damage than  $s_i$ . As the focus of this research is to gauge the contribution of the involved parameters to the extent of a structure’s relative damage, pairs with  $y_i = y_j$  were not included in the transformed dataset.

Thus, the final transformed dataset had inputs  $X^{new}$  and outputs  $y^{new}$  obtained via the above transformations described.

### 2.3. Machine Learning Algorithm

In order to analyze the importance of each feature for the relative classification of each pair of structures, we considered three different pairings of structures. Specifically, we considered the subset consisting of the (Green, Yellow), (Yellow, Red), and (Red, Black) structures. We did this because each of the labels has a very distinct definition: the Black and Red structures correspond to the Collapse state and Ultimate Limit State (ULS), respectively, while Yellow corresponds to the Serviceability Limit State (SLS). Thus, by using this pairing our models learn to distinguish adjacent damage states and the features that lead to this increase in damage. For each of these pairs, we performed the pairwise transformations presented above. The number of structures in each pair and each transformed dataset is shown in Table 1.

**Table 1.** Number of structures for each label pair and corresponding samples in the transformed dataset.

Pair	Damage Threshold	Number of Structures	Samples in the Transformed Dataset
(Green, Yellow)	Serviceability Limit State	(92, 69)	6348
(Yellow, Red)	Ultimate Limit State	(69, 102)	7038
(Red, Black)	Collapse Limit State	(102, 90)	9180

We constructed a binary classifier for each of the above pairs, as described in Section 2.2. The subsequent analysis of the importance of the features of these classifiers helps to determine the deciding factors that lead a structure to being in the Red rather than the Yellow category, i.e., crossing the ULS and suffer heavy damage instead of only crossing the SLS and suffering moderate damage. There are many classifiers available in the literature to perform this task. In [22], the authors worked on the same dataset and analyzed a variety of models. The best performing one was found to be the Gradient Boosting (GB) Classifier [33], which is what we employing in the sequelae. GB is a powerful method that learns a classifier incrementally, starting from a base model; specifically, it learns a function

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i h_i(\mathbf{x}; \theta_i), \tag{1}$$

where  $h_i$  represents the individual “weak” models (Decision Trees [34]) that the algorithm learns at each iteration,  $\theta_i$  represents their parameters,  $N$  is the user-defined number of such models, and  $\alpha_i$  represents the learned weights that produce the final linear combination. The steps of the method are shown in Algorithm 1 [35]. The algorithm was implemented

in Python programming language (v. 3.11.5) using the scikit-learn machine learning library (v. 1.3.0) [36].

---

**Algorithm 1** Gradient Boosting Learning Process [35]

---

Initialize  $f_0(\mathbf{x})$

**for**  $i = 1, 2, \dots, N$  **do**:

    Compute  $w_j(\mathbf{x}_j) = \frac{\partial \mathcal{L}(y_j, F(\mathbf{x}_j))}{\partial F(\mathbf{x}_j)} \Big|_{F(\mathbf{x})=F_{i-1}(\mathbf{x})}$ ,  $j = 1, 2, \dots, M$

    Compute  $\theta_i = \arg \min_{\theta, \mu} \sum_{j=1}^M [-w_j(\mathbf{x}_j) - \mu h_i(\mathbf{x}; \theta_i)]^2$

    Compute  $\alpha_i = \arg \min_{\alpha} \sum_{j=1}^M \mathcal{L}(y_j, f_{i-1} + \alpha h_i(\mathbf{x}; \theta_i))$

    Update  $f_i(\mathbf{x}) = f_{i-1}(\mathbf{x}) + \lambda \alpha_i h_i(\mathbf{x}; \theta_i)$

**end for**

---

In the above algorithm,  $\mathcal{L}$  is the loss function that measures the error between the predictions and the true values,  $M$  is the number of samples the model is trained on and  $\lambda > 0$  is the so-called “learning rate”, which modifies the contribution of each individual tree [37].

#### 2.4. Hyperparameter Tuning

As is evident from (1) and Algorithm 1, Gradient Boosting learns a number of parameters during its training, e.g., weights  $\alpha_i$ . However, there are a number of so-called hyperparameters, i.e., parameters set by the user before training begins, such as the number  $N$  of individual Decision Trees and the maximum allowed depth of each tree. The configuration of these hyperparameters can reduce overfitting [38,39] and has a direct impact on the overall accuracy of the model [40].

Thus, the importance of appropriately tuning of these hyperparameters to achieve optimal results becomes clear. This has led to a variety of methods to address this process, with reviews of the existing algorithms provided by Yu and Zhu [41] and by Yang and Shami [40]. In this paper we opt for Bayesian optimization, as it does not search the hyperparameter space blindly, instead using each iteration’s results in the next one, which can lead to faster convergence to the optimal solution [42]. The implementation was carried out using the dedicated Python library scikit-optimize [43] (v. 0.10.1).

#### 2.5. SHAP

A common measure used to gauge the strength of each feature’s effect on the outcome, which is the focus of the present study, is the so-called SHapley Additive exPlanation (SHAP) [19]. This is the equivalent in the machine learning literature to the Shapley values in cooperative game theory introduced by Lloyd Shapley in 1951 [44]. SHAP values provide interpretability by constructing a simpler explainable model in the local neighborhood of each point in the dataset. Thus, given a learned Machine Learning model  $f$ , a local approximation  $g$  can be formulated as follows [19]:

$$g(\mathbf{u}) = \phi_0 + \sum_{i=1}^n \phi_i u_i, \quad (2)$$

where  $n$  is the number of features,  $\mathbf{u} \in \mathbb{R}^n$  is a binary vector whose value in the  $i^{\text{th}}$  position denotes whether or not the corresponding feature was used in the prediction, and  $\phi_i$  denotes the SHAP value of that feature, i.e., the strength of its contribution to the model’s output.

The values of the  $\phi_i$ s, following the notation of Lundberg et al. [45], are computed as follows: let  $N = 1, 2, \dots, n$  be the set of features used and let  $S \subseteq N$  be a subset of  $N$ ; then, we have [45,46]

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [f(S \cup \{i\}) - f(S)]. \quad (3)$$

Intuitively, this corresponds to the weighted average over all feature combinations (coalitions) of the difference in the model prediction with and without the inclusion of the  $i^{\text{th}}$  feature.

As has been mentioned, the above  $\phi_i$  values pertain to a specific point. For example, considering the pair (Red, Black), there were 102 Red structures in the undersampled dataset and 90 Black ones, yielding  $90 \times 102 = 9180$  pairs, i.e., samples in the transformed space, as shown in Table 1.

Thus, we have a matrix  $\Phi \in \mathbb{R}^{9180 \times 13}$  in which each value  $\phi_{ij}$  is the SHAP value of the  $j^{\text{th}}$  feature calculated at the  $i^{\text{th}}$  sample. Thus, in order to obtain an aggregated value for the whole dataset, we used a normalized norm of each column in the matrix. We compared the results obtained using the  $\mathcal{L}_1$  norm (sum of absolute values), which is the most commonly used in the literature, and the well known Euclidean norm  $\mathcal{L}_2$ , which increases the contribution of larger values while simultaneously reducing the effect of smaller noisy components. Thus, for each feature  $j = 1, 2, \dots, 13$  we considered the alternatives as obtained by Equation (4):

$$\bar{\phi}_{ij} = \begin{cases} \frac{1}{m} \sum_{i=1}^m |\phi_{ij}|, & \text{using } \mathcal{L}_1 \\ \frac{1}{m} \sqrt{\sum_{i=1}^m \phi_{ij}^2}, & \text{using } \mathcal{L}_2 \end{cases} \quad (4)$$

where  $m$  is the number of samples in the transformed space for each pair, as shown in Table 1. The computation of the SHAP values was carried out using the dedicated Python library by Lundberg et al. [47].

Thus, our overall proposed methodology comprises the following steps. For each damage threshold: (1) obtain the transformed inputs  $X^{new}$  and outputs  $y^{new}$  as described in Section 2.2; (2) train the corresponding binary classification ML model using data for the particular damage threshold; and (3) obtain the feature importance metrics of the trained ML model using SHAP values via Equation (4).

### 3. Results

As previously stated, the main focus of this study is to analyze the importance of each feature in deciding whether a structure will cross each of the respective damage thresholds. As explained in Section 2.5, this is carried out using SHAP values, which offer just such a quantification. However, the reliability of any feature importance analysis is directly related to the performance of the model under consideration. If a model has poor performance, then the way that it arrives at its predictions will not be very informative. On the other hand, the higher a model's performance, the closer its predictions are to the truth. Thus, the extracted feature importance values are closely coupled with the underlying physical phenomenon, and can be considered highly reliable.

To this end, the rest of this section is structured as follows. In the first part, Section 3.1, we present the results of the hyperparameter tuning and the classification performance metrics. Tuning the hyperparameters allows us to find the model with the highest accuracy and the most reliable feature importance values. Subsequently, we present the accuracy metrics obtained using the optimal values of the involved hyperparameters. This demonstrates the high accuracy obtained by the models, especially in the most critical damage categories, which enhances the reliability of the extracted feature importance values. Finally,

in Section 3.2, we present the main results of this research based on the feature importance values obtained from these models.

### 3.1. Binary Classifiers and Hyperparameter Tuning

As mentioned in Section 2.3, we constructed a binary classifier for each pair of labels considered here, namely, (Green, Yellow), (Yellow, Red) and (Red, Black). Each of these classifiers was tuned separately, and we optimized the following hyperparameters:

- **max\_depth**: This is the maximum allowed depth of each individual Decision Tree; too large or too small values can lead to overfitting or underfitting, respectively [48].
- **n\_estimators**: This is the number of individual Decision Trees used in Gradient Boosting.
- **min\_samples\_leaf**: This is the minimum number of samples that must remain in an end node (leaf) of each individual tree.
- **learning\_rate**: This controls the contribution of each individual tree, as shown in Algorithm 1. If the value is too large, the algorithm might overfit; however, a lower learning rate has the trade-off that more trees are required to reach the desired accuracy.

Table 2 presents the tuning range of each hyperparameter as well as the optimal value for each of the three classifiers considered here.

**Table 2.** Hyperparameter tuning.

Hyperparameter	Tuning Range	Optimal Value per Pair		
		(Green, Yellow)	(Yellow, Red)	(Red, Black)
max_depth	[3, 11]	3	5	3
n_estimators	[50, 300]	297	50	293
min_samples_leaf	[1, 10]	9	8	10
learning_rate	[0.05, 0.25]	0.086887	0.120314	0.182278

Having obtained the optimal hyperparameter configuration, we trained and tested our three models using five-fold cross-validation [49]. In this framework, the dataset is split into five parts and each part is iteratively used as test set, while the remaining parts are used for training. This ensures that the model’s predictions are always on unseen data and reduces the sensitivity/variability of the obtained performance metrics. The performance was measured using the well known classification metrics of Precision, Recall, F1-score, Accuracy, and Area Under the Curve (AUC) [50], with the results shown in Table 3. The results clearly show that the classifiers achieved high performance, especially for the most critical pairs, i.e., (Red, Black) and (Yellow, Red). The accuracy with which the model was able to distinguish between these two categories increases the reliability of the feature importance analysis, which is the main focus of the study.

**Table 3.** Classification metrics for the binary classifier of each pair cross-validated on the whole dataset.

	(Green, Yellow)		(Yellow, Red)		(Red, Black)	
	−1	+1	−1	+1	−1	+1
Precision	0.69585	0.76301	0.90943	0.86933	0.93992	0.90379
Recall	0.73221	0.72928	0.84995	0.92183	0.88501	0.95024
F1-score	0.71357	0.74576	0.87869	0.89481	0.91164	0.92644
Accuracy	0.73062		0.88732		0.91972	
AUC	0.81451		0.95232		0.98128	

### 3.2. Feature Importance

This subsection presents our main results analyzing of the importance of the RVS features for the relative classification of structures, which we performed using the SHAP values, as explained in Section 2.5. Note that there is some inherent variability in the

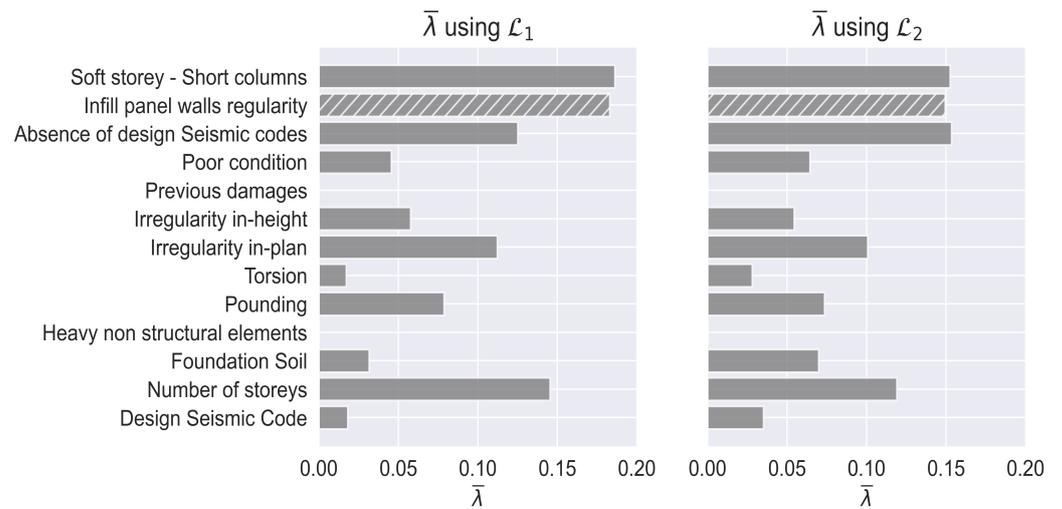
computations of the  $\phi_i$ , and consequently in  $\bar{\phi}$  from (3) and (4). This can stem from how the algorithm splits the dataset between training and testing at each iteration or from the computation of the SHAP values themselves. To mitigate the sensitivity of the results to these factors, we performed 100 runs of our proposed methodology and averaged the obtained feature importances. This heavily reduces the variability of the computations and increases the reliability of the extracted feature importance values. Thus, we constructed a matrix  $\Theta \in \mathbb{R}^{100 \times 13}$ , where  $\theta_{ij}$  is the value  $\bar{\phi}_j$  from (4) for the  $j^{\text{th}}$  feature at the  $i^{\text{th}}$  iteration. From this, we calculated the average value per column/feature, i.e., we defined

$$\lambda_j = \frac{1}{100} \sum_{i=1}^{100} \theta_{ij}. \tag{5}$$

Finally, in order to normalize these coefficients, we divided them with their sum, i.e.,

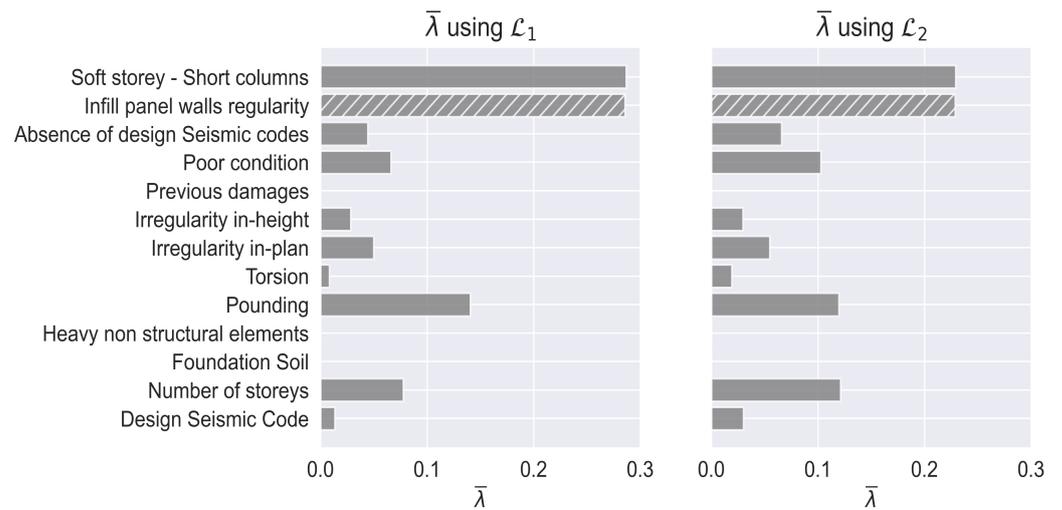
$$\bar{\lambda}_j = \frac{\lambda_j}{\sum_{j=1}^{13} \lambda_j}. \tag{6}$$

With this normalization, we now have  $0 \leq \bar{\lambda}_i \leq 1$  and  $\sum_{i=1}^{13} \bar{\lambda}_i = 1$ ; therefore, these coefficients can be interpreted as the percentage of the contribution of the corresponding features to the overall predictions of the model. We carried out the above using both of the alternatives used in (4). The results are shown in Figure 4.

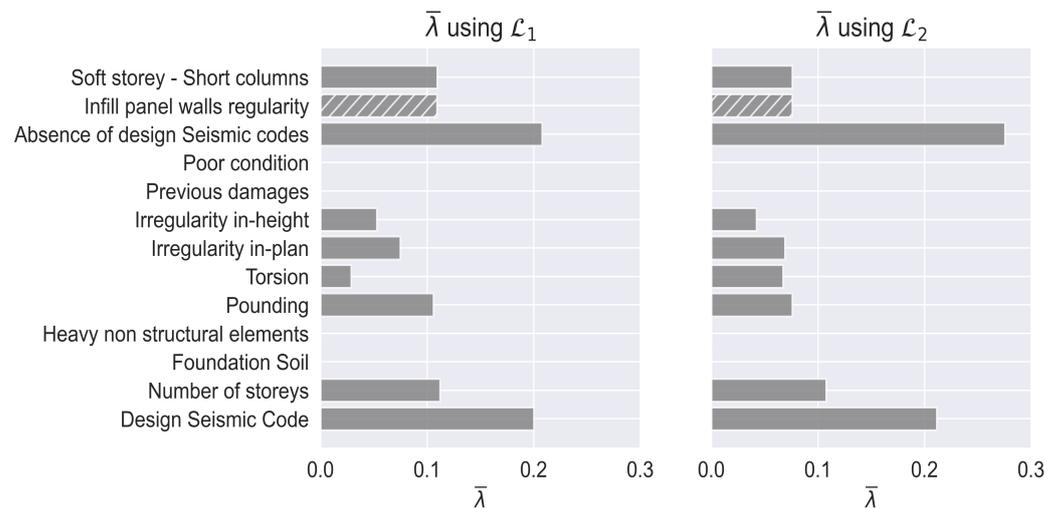


(a) (Red, Black) pair.

Figure 4. Cont.



(b) (Yellow, Red) pair.



(c) (Green, Yellow) pair.

**Figure 4.** Feature importance coefficients  $\bar{\lambda}_i$  as defined in (6) for the three damage label pairs: (a) (Red, Black), (b) (Yellow, Red), and (c) (Green, Yellow).

This figure presents the comparative results of the contribution of each feature to the model predictions expressed as a percentage of the total. As previously discussed, these correspond to the mean contributions for all pairs of structures, which, given that we are averaging over thousands of pairs, are representative of the overall parameter effect on seismic behaviour across the various limit states of all the structures in the dataset. The left subfigures in Figure 4a–c pertain to  $\mathcal{L}_1$ , i.e., the absolute values of these features, while the right subfigures pertain to  $\mathcal{L}_2$ , i.e., their squares. The results demonstrate a basic hierarchy of the structural properties that influence the seismic vulnerability of the studied structures and contribute to the observed degrees of damage. In general, the results are in agreement with the existing structural mechanics literature and the seismic behaviour of reinforced concrete structures. We analyze and discuss each of Figure 4a–c separately.

- Distinction between Red (ULS) and Black (Collapse): As can be seen from the left part of Figure 4a, the most crucial factor overall for the Collapse Limit State is the presence of soft storeys and/or short columns, with a weight of approximately 18%.

The presence of regular infill panel walls, however, has an almost equal in magnitude, but a positive effect, which is why the corresponding bar in the figure is hatched. This is an important feature that helped prevent structures that crossed the ULS to cross the CLS as well. Finally, the absence of design seismic codes, the number of storeys in the structure, and the presence of an irregular plan all play import roles for this damage threshold.

The right part of this feature displays an important distinction, as the absence of design seismic codes is now the dominant feature, even if only slightly. This can be explained in the following way. The absence of design seismic codes feature is indeed a crucial factor, as is well known in the literature, and the model assigns high SHAP values to it. However, not many structures were affected by this feature. Of the 452 structures in our dataset, only 26 lacked a design seismic code. Of these, 20 (77%) crossed the ULS, and 19 of those (95%) crossed the CLS as well. Thus, by taking the squares of the SHAP values, as per the right figure of Figure 4a, we assign more weight to these extreme SHAP values even though they pertained to only a limited number of cases. It is important to note that there is not a noteworthy distinction in the other factors, such as soft storeys/short columns, regularity of the infill panel walls, or structure height, between the left and right subfigures of Figure 4a, as the corresponding SHAP values are more balanced.

- Distinction between Yellow (SLS) and Red (ULS): As can be seen from Figure 4b, the most important features by far are the presence of soft storeys and/or short columns as well as the presence of regular infill wall panels. Soft storeys/short columns had a detrimental effect, accounting for approximately 30% of the total. On the other hand, regular infill wall panels had a beneficial effect with approximately equal magnitude. This is in agreement with the established engineering literature, as bricks walls help to reduce storey drift, and consequently decrease the overall degree of damage. The absence of design seismic codes did not play an important role in this case, as most structures that displayed this feature crossed the CLS as well, as mentioned above. Pounding, on the other hand, had a contribution of approximately 15%. The height of the structure and potential preexisting poor condition accounted for 7–8% each. Out of the thirteen total features, these five combined to account for approximately 85% of the total in the model's predictions. Finally, we note that in this case the SHAP values are balanced, as the left and right subfigures, using  $\mathcal{L}_1$  and  $\mathcal{L}_2$ , respectively, show minimal differences.
- Distinction between Green (minimal to no damage) and Yellow (SLS): Finally, the results for the distinction between structures that crossed the SLS (Yellow) and those that suffered minimal to no damage are shown in Figure 4c. It can be seen that the most important factors here are the existence and type of design seismic codes, each of which account for approximately 20% of the total. This is in agreement with the post-1985 Greek seismic codes, which enforce lower damage degrees for the same earthquake design. Regular infill panel walls, soft storeys and/or short columns, and the presence of adjacent structures that could lead to pounding were relevant here, although the magnitude of their effect was only approximately 10%.

#### 4. Summary and Conclusions

In this research, we have employed a novel machine learning methodology to approach one of the problems commonly found in countries with high seismic activity, namely, that of the preseismic structural assessment. Specifically, we performed an analysis of how the features obtained in the Rapid Visual Screening procedure affect the seismic vulnerability of structures. We specifically focused on three well-known damage thresholds: the Serviceability Limit State, the Ultimate Limit State, and the Collapse Limit State, to further emphasize structures that, in addition to crossing the ULS, suffered total or partial collapse. We employed a pairwise approach to perform our analysis, creating pairs from all structures belonging to adjacent damage categories, as shown in Table 1. We then

used a Gradient Boosting Machine to create a binary classification model that learned to distinguish structures for each of the above damage thresholds. As shown in Table 2, we tuned some of the model's hyperparameters to increase its performance. This led to the model having high accuracy, especially in the higher damage categories.

As can be seen from Table 3, the model learned to distinguish the CLS threshold with almost 92% accuracy; similarly, for the ULS threshold it displayed an accuracy close to 89%. While the model's performance dropped to 73% for the SLS, this is the least impactful of the three damage thresholds in engineering practice. Finally, we used SHAP values to quantify the effect of each of the features in our models' predictions. The previously mentioned high accuracy of our models, especially in the higher damage categories, enhances the reliability of the subsequently extracted SHAP values.

In addition, the present study highlights the participation of various factors that contribute to the overall structural vulnerability index as calculated via the RSVP. Qualitatively, our results broadly agree with the previously established engineering literature. For the CLS threshold, soft storeys/short columns, the height of the structure, absence of design seismic codes, and irregularities in height and plan were the most impactful detrimental factors. Regular infill wall panels were shown to have a very positive effect. For the ULS threshold, the absence of a design seismic code did not have a significant influence, as the vast majority of structures with this feature that crossed the ULS crossed the CLS as well. Finally, the implementation of modern design seismic codes played a crucial role in preventing structures from crossing the SLS threshold.

The quantitative results obtained via the application of ML methods and SHAP values demonstrates the potential applicability of this approach for recalibrating the computation of structural vulnerability indices using data from recent earthquakes. The method implemented in the present paper pertains to reinforced concrete structures with a particular set of input features; however, it could be implemented in an identical manner using a different set of input features, for example, in countries where other parameters are deemed more important. It could also be employed in different structural types altogether, for example, in masonry buildings commonly found in traditional communities.

**Author Contributions:** Conceptualization, I.K. and A.K.; methodology, I.K., L.I. and A.K.; software, I.K., L.I. and A.K.; validation, I.K., L.I. and A.K.; formal analysis, I.K., L.I. and A.K.; investigation, I.K., L.I. and A.K.; resources, A.K.; data curation, I.K., L.I. and A.K.; writing—original draft preparation, I.K. and A.K.; writing—review and editing, L.I.; visualization, I.K.; supervision L.I. and A.K. 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 dataset employed in this study can be made available upon reasonable request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

RVSP	Rapid Visual Screening Procedure
ML	Machine Learning
SLS	Serviceability Limit State
ULS	Ultimate Limit State
CLS	Collapse Limit State
SHAP	SHapley Additive exPlanations

## References

1. Palermo, V.; Tsonis, G.; Sousa, M.L. *Building Stock Inventory to Assess Seismic Vulnerability Across Europe*; Publications Office of the European Union: Luxembourg, 2018.
2. Federal Emergency Management Agency (US). *Rapid Visual Screening of Buildings for Potential Seismic Hazards: A Handbook*; Government Printing Office: Washington, DC, USA, 2017.
3. Greek Code for Seismic Resistant Structures–EAK. 2000. Available online: [https://iisee.kenken.go.jp/worldlist/23\\_Greece/23\\_Greece\\_Code.pdf](https://iisee.kenken.go.jp/worldlist/23_Greece/23_Greece_Code.pdf) (accessed on 3 January 2024).
4. Lizundia, B.; Durphy, S.; Griffin, M.; Holmes, W.; Hortacsu, A.; Kehoe, B.; Porter, K.; Welliver, B. Update of FEMA P-154: Rapid visual screening for potential seismic hazards. In *Improving the Seismic Performance of Existing Buildings and Other Structures*; American Society of Civil Engineers: Reston, VA, USA, 2015; pp. 775–786.
5. Vulpe, A.; Carausu, A.; Vulpe, G.E. Earthquake induced damage quantification and damage state evaluation by fragility and vulnerability models. In Proceedings of the SMiRT 16, Washington, DC, USA, 12–17 August 2001.
6. NEHRP Handbook for the Seismic Evaluation of Existing Buildings. Available online: [https://www.preventionweb.net/files/7543\\_SHARPISDRFLOOR120081209171548.pdf](https://www.preventionweb.net/files/7543_SHARPISDRFLOOR120081209171548.pdf) (accessed on 3 January 2024).
7. Rossetto, T.; Elnashai, A. Derivation of vulnerability functions for European-type RC structures based on observational data. *Eng. Struct.* **2003**, *25*, 1241–1263. [[CrossRef](#)]
8. Eleftheriadou, A.; Karabinis, A. Damage probability matrices derived from earthquake statistical data. In Proceedings of the 14th World Conference on Earthquake Engineering, Beijing, China, 12–17 October 2008; pp. 07–0201.
9. Chieffo, N.; Formisano, A.; Louren, P.B. Seismic vulnerability procedures for historical masonry structural aggregates: Analysis of the historical centre of Castelpoto (South Italy). *Structures* **2023**, *48*, 852–866. [[CrossRef](#)]
10. Chieffo, N.; Fasan, M.; Romanelli, F.; Formisano, A.; Mochi, G. Physics-based ground motion simulations for the prediction of the seismic vulnerability of masonry building compounds in Mirandola (Italy). *Buildings* **2021**, *11*, 667. [[CrossRef](#)]
11. Scala, S.A.; Gaudio, C.D.; Verderame, G.M. Influence of construction age on seismic vulnerability of masonry buildings damaged after 2009 L'Aquila earthquake. *Soil Dyn. Earthq. Eng.* **2022**, *157*, 107199. [[CrossRef](#)]
12. Scala, S.A.; Gaudio, C.D.; Verderame, G.M. Towards a multi-parametric fragility model for Italian masonry buildings based on the informative level. *Structures* **2024**, *59*, 105613. [[CrossRef](#)]
13. Harirchian, E.; Kumari, V.; Jadhav, K.; Das, R.R.; Rasolzade, S.; Lahmer, T. A Machine Learning Framework for Assessing Seismic Hazard Safety of Reinforced Concrete Buildings. *Appl. Sci.* **2020**, *10*, 7153. [[CrossRef](#)]
14. Sajan, K.; Bhusal, A.; Gautam, D.; Rupakhety, R. Earthquake damage and rehabilitation intervention prediction using machine learning. *Eng. Fail. Anal.* **2023**, *144*, 106949.
15. Luo, H.; Paal, S.G. A locally weighted machine learning model for generalized prediction of drift capacity in seismic vulnerability assessments. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 935–950. [[CrossRef](#)]
16. Kazemi, F.; Asgarkhani, N.; Jankowski, R. Machine learning-based seismic response and performance assessment of reinforced concrete buildings. *Arch. Civ. Mech. Eng.* **2023**, *23*, 94. [[CrossRef](#)]
17. Burkart, N.; Huber, M.F. A survey on the explainability of supervised machine learning. *J. Artif. Intell. Res.* **2021**, *70*, 245–317. [[CrossRef](#)]
18. Mangalathu, S.; Karthikeyan, K.; Feng, D.-C.; Jeon, J.-S. Machine-learning interpretability techniques for seismic performance assessment of infrastructure systems. *Eng. Struct.* **2022**, *250*, 112883. [[CrossRef](#)]
19. Futagami, K.; Fukazawa, Y.; Kapoor, N.; Kito, T. Pairwise acquisition prediction with shap value interpretation. *J. Financ. Data Sci.* **2021**, *7*, 22–44. [[CrossRef](#)]
20. Karabinis, A. Calibration of Rapid Visual Screening in Reinforced Concrete Structures Based on Data after a Near Field Earthquake (7.9.1999 Athens-Greece). 2004. Available online: [https://oasp.gr/sites/default/files/program\\_documents/261%20-%20Teliki%20ekthesi.pdf](https://oasp.gr/sites/default/files/program_documents/261%20-%20Teliki%20ekthesi.pdf) (accessed on 20 March 2024).
21. Ruggieri, S.; Cardellicchio, A.; Leggieri, V.; Uva, G. Machine-learning based vulnerability analysis of existing buildings. *Autom. Constr.* **2021**, *132*, 103936. [[CrossRef](#)]
22. Karampinis, I.; Iliadis, L. A Machine Learning Approach for Seismic Vulnerability Ranking. In Proceedings of the International Conference on Engineering Applications of Neural Networks, León, Spain, 14–17 June 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 3–16.
23. Elrahman, S.M.A.; Abraham, A. A review of class imbalance problem. *J. Netw. Innov. Comput.* **2013**, *1*, 332–340.
24. Longadge, R.; Dongre, S. Class imbalance problem in data mining review. *arXiv* **2013**, arXiv:1305.1707.
25. Maheshwari, S.; Jain, R.; Jadon, R. A review on class imbalance problem: Analysis and potential solutions. *Int. J. Comput. Issues (IJCSI)* **2017**, *14*, 43–51.
26. Satyasree, K.; Murthy, J. An exhaustive literature review on class imbalance problem. *Int. J. Emerg. Trends Technol. Comput. Sci.* **2013**, *2*, 109–118.
27. Bansal, A.; Jain, A. Analysis of focussed under-sampling techniques with machine learning classifiers. In Proceedings of the 2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA), Las Vegas, NE, USA, 22–25 May 2022; IEEE: Piscataway, NJ, USA, 2021; pp. 91–96.

28. Mohammed, R.; Rawashdeh, J.; Abdullah, M. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In Proceedings of the 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 243–248.
29. Newaz, A.; Hassan, S.; Haq, F.S. An empirical analysis of the efficacy of different sampling techniques for imbalanced classification. *arXiv* **2022**, arXiv:2208.11852
30. Hasanin, T.; Khoshgoftaar, T. The effects of random undersampling with simulated class imbalance for big data. In Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 6–9 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 70–79.
31. Liu, B.; Tsoumakas, G. Dealing with class imbalance in classifier chains via random undersampling. *Knowl.-Based Syst.* **2020**, *192*, 105292. [[CrossRef](#)]
32. Liu, Y.; Li, X.; Kong, A.W.K.; Goh, C.K. Learning from small data: A pairwise approach for ordinal regression. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
33. Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurobotics* **2013**, *7*, 21. [[CrossRef](#)]
34. Kingsford, C.; Salzberg, S.L. What are decision trees? *Nat. Biotechnol.* **2008**, *26*, 1011–1013. [[CrossRef](#)] [[PubMed](#)]
35. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
37. Zhang, Y.; Haghani, A. A gradient boosting method to improve travel time prediction. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 308–324. [[CrossRef](#)]
38. Hawkins, D.M. The problem of overfitting. *J. Chem. Comput. Sci.* **2004**, *44*, 1–12.
39. Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 437–478.
40. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [[CrossRef](#)]
41. Yu, T.; Zhu, H. Hyper-parameter optimization: A review of algorithms and applications. *arXiv* **2020**, arXiv:2003.05689.
42. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NE, USA, 3–6 December 2012; Volume 25.
43. Head, T.; MechCoder; Louppe, G.; Shcherbatyi, I.; fcharras; Vinícius, Z.; cmmalone; Schröder, C.; nel215; Campos, N.; et al. scikit-optimize/scikit-optimize: V0. 5.2. Version v0. 2018. Available online: <https://zenodo.org/records/1207017> (accessed on 20 March 2024).
44. Shapley, L.S. *Notes on the n-Person Game: The Value of an n-Person Game*; RAND Corporation: Santa Monica, CA, USA, 1951; Volume 7.
45. Lundberg, S.M.; Erion, G.G.; Lee, S.-I. Consistent individualized feature attribution for tree ensembles. *arXiv* **2018**, arXiv:1802.03888.
46. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
47. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.-I. From local explanations to global understanding with explainable ai for trees. *Nat. Mach.* **2020**, *2*, 2522–5839. [[CrossRef](#)]
48. Zharmagambetov, A.; Hada, S.S.; Carreira-Perpiñán, M.Á.; Gabidolla, M. An experimental comparison of old and new decision tree algorithms. *arXiv* **2019**, arXiv:1911.03054.
49. Browne, M. Cross-validation methods. *J. Math. Psychol.* **2000**, *44*, 108–132. [[CrossRef](#)] [[PubMed](#)]
50. Ferrer, L. Analysis and comparison of classification metrics. *arXiv* **2022**, arXiv:2209.05355.

**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.