André Manuel Alves Costa

Machine Learning for Earthquake Damage Detection: A Comparative Analysis of Algorithm Performance

Machine Learning for Earthquake Damage Detection: A Comparative Analysis of Algorithm Performance.

By:
André Manuel Alves Costa

Supervised by
Feliz Ribeiro, PhD

Co-Supervised by

Vítor Silva, PhD

Dissertation presented to University Fernando Pessoa as part of the requirements for obtaining the degree of Master in Computer Engineering – Mobile Computing

Biographical Note

André Manuel Alves Costa was born in Vila Nova de Gaia, on 15th January 1997. In 2021 he completed a bachelor's degree in Computer Engineering at University Fernando Pessoa. He enrolled in the same year in the Master in Computer Engineering – Mobile Computing, at the same University. André worked as a junior researcher for the ASSIMILATE project from October 2021.

Acknowledgments

Throughout my master's journey and research on the ASSIMILATE project, I could improve my hard and soft skills, and expand my knowledge in computer science, particularly in machine learning. This experience not only enriched my academic knowledge but also played a pivotal role in shaping my personal and professional aspirations.

I would like to extend my deepest gratitude to my professors and supervisors, Feliz Gouveia and Vitor Silva, for their unwavering commitment, guidance, and shared knowledge throughout the construction of this dissertation and my entire master's program. Their support has been instrumental. In addition, I extend my appreciation to Petros Kalakonas for assisting in the provision of essential data used in this research.

I'd like to offer my heartfelt thanks to Cátia for her exceptional support, patience, and motivating presence in this challenging path. And thank my family and friends for always believing in me.

Lastly, I wish to acknowledge the inspiring colleagues and professors I had the privilege of encountering during my academic tenure. Their contributions have significantly enriched my educational experience and have greatly influenced the successful completion of this dissertation.

The support of the Portuguese Foundation of Science and Technology, through research grant PTDC/ECI-EGC/7244/2020, project ASSIMILATE, is greatly acknowledged.

Abstract

This work aims to analyse and improve the results of machine learning algorithms for estimating damage in buildings following an earthquake, thus enabling rapid post-earthquake assessment to prevent further physical, economic and social damage. Using real datasets, nine algorithms were tested and compared: Ridge Regressor, Lasso Regressor, Support Vector Regressor, Decision Tree, Random Forest, Gradient Boost, Extreme Gradient Boost, Artificial Neural Networks and Multi-layer Perceptron. The key findings of the research resulted in the demonstration of the importance of dataset practicality, while encompassing heterogeneity of buildings, and highlights the positive impact of data transformation on algorithm performance when compared to previous research papers lacking such transformations. Furthermore, it was concluded that the Artificial Neural Network algorithm consistently outperforms others, justifying its academic and practical preference despite the longer training times and reaffirming its significance in earthquake damage prediction. It was possible to assess that other algorithms such as Gradient Boost, Extreme Gradient Boost and Random Forest are acceptable, practical, understandable and reliable alternatives. These findings contribute to the advance of earthquake engineering and highlight the potential of Machine Learning in post-earthquake risk mitigation.

Keywords: machine learning algorithms, seismic damage detection, earthquake engineering.

Sumário

Esta dissertação tem por objetivo analisar a eficiência da deteção de danos sísmicos em edifícios, através do teste e comparação de nove algoritmos de Aprendizagem Automática (*Ridge Regressor, Lasso Regressor*, *Support Vetor Regressor*, *Decision Tree*, *Random Forest, Gradient Boost, Extreme Gradient Boost, Artificial Neural Networks* e *Multi-layer Perceptron*), permitindo uma rápida avaliação pós-sísmica para prevenir adicionais danos físicos, económicos e sociais. Os principais resultados da pesquisa resultaram na demonstração da importância da praticidade do conjunto de dados utilizados, abrangendo também a heterogeneidade dos edifícios, e destaca o impacto positivo da transformação de dados no desempenho do algoritmo quando comparado com trabalhos de investigação atuais que não contemplam tais transformações. Além disso, concluiu-se que o algoritmo *Artificial Neural Network* supera consistentemente os outros algoritmos, justificando assim o motivo da sua preferência ao nível acadêmico e profissional, apesar dos prolongados tempos de treinamento requeridos, reafirmando assim a sua importância na previsão de danos causados por treino. Contudo, foi possível determinar que outros algoritmos como *Gradient Boost, Extreme Gradient Boost e Random Forest* revelaram ser alternativas aceitáveis, práticas, compreensíveis e confiáveis. Estes resultados contribuem para o avanço da engenharia sísmica e sublinham o potencial da Aprendizagem Automática na mitigação do risco pós-sismo.

Palavras-chave: algoritmos de aprendizagem automática, deteção de danos sísmicos, engenharia sísmica, regressão, redes neuronais.

Table Contents

List of Tables

List of Figures

Acronyms

AI - Artificial Intelligence

ANN - Artificial Neural Network

DTR - Decision Tree Regressor

GBR - Gradient Boosting Regressor

IM – Intensity Measure

LR - Lasso Regressor

MAE - Mean Absolute Error

ML - Machine Learning

MLP - Multi Layer Perceptron

MSE – Mean Squared Error

OLS – Ordinary Least Squares

PGA – Peak Ground Acceleration

RFR - Random Forest Regressor

RMSE – Root Mean Squared Error

RR - Ridge Regressor

S-10V – Dataset with 10 input variables

S-10V-NL/L – Dataset with 10 variables corrected for linearity

S-19V – Dataset with 19 input variables

SA - Spectral acceleration

SD – Standard deviation

SVR - Support Vector Regressor

XGBR - Extreme Gradient Boosting Regressor

1. Introduction

Earthquakes are an extremely important natural phenomenon that must be studied as their impact can be devastating, physically as well as economically and socially. Throughout history many severe seismic events have been recorded worldwide. In Portugal, for example, there was the Lisbon earthquake in 1755 and the 1909 earthquake in Benavente and Salvaterra de Magos. The study of earthquakes is of the utmost importance because it is a permanent necessity, as we have recently seen in 2023 in Turkey and Morocco. Thus, many academics and researchers have focused their attention on predicting and detecting negative impacts on urban infrastructure, recurring to traditional, statistical and analytical methods, and more recently resorting to Machine Learning (ML) algorithms, being the latter the scope of this research.

It is extremely important to act quickly and efficiently after the occurrence of potentially destructive natural phenomena, more specifically in the case of earthquakes. In fact, it has been observed that, even a few days after the occurrence, buildings affected by earthquakes can collapse. In this context, it becomes crucial to have the means to carry out a rapid assessment of buildings at risk of colliding with other buildings. In order to develop fast and reliable means of detection, it is possible to resort to ML algorithms.

Therefore, this work aims to contribute to the study of ML algorithms for the assessment of damage caused by earthquakes in buildings immediately after their occurrence. The solution to be developed will allow assessing the damage in a more efficient way, as soon as possible, to avoid possible additional post-earthquake catastrophes.

1.1. Objectives

The objectives of this dissertation are:

1. The study and development of a state-of-the-art ML based regression problem to predict seismic damage in buildings.

2. The creation of a ML model to accurately predict seismic damage and usage of various algorithms to compare.
3. The evaluation of the selected algorithms based on their results and suitability for predicting building damage.

The dataset available for conducting the second objective concerns seven different building classes representing the most common constructions in the Balkan region, determined by key structural attributes like construction material, lateral load resistance, number of storeys, and seismic code. Please note, that numerical modelling is employed to simulate these structures, further enabling the extraction of specific parameters, such as the maximum displacement. The input variables are the intensity measures (IM) - quantitative parameters used to assess the severity of ground shaking during an earthquake or seismic event. They are: Peak Ground Acceleration (PGA, maximum recorded acceleration at a specific location during an earthquake is usually obtained by seismographs or accelerometers), and Spectral acceleration (SA, obtained from a frequency-dependent function that characterizes a structure's maximum acceleration response at various frequencies) in multiple ranges. The output variable considered is the Maximum Displacement, representing the peak movement of a building component during seismic events, conventionally indicating damage beyond a specific threshold (per building type class). Predicting the Maximum Displacement is thus the goal of this work.

1.2. Problem

The main problems of predicting damage to buildings after a seismic event using machine learning are:

- The availability of high-quality training data: the accuracy of a ML model is heavily dependent on the quality and completeness of the training data that is provided. In the case of predicting damage to buildings after a seismic event, this data is difficult to obtain and is likely to be incomplete or inaccurate in many cases. This can affect the ability of the model to accurately learn the relationships between the factors involved and to make accurate predictions.

- The complexity of the relationships between the factors involved: the relationships between the characteristics of the buildings, the seismic event, and the extent of the damage are complex and may not be fully understood. This can make it difficult for the model to accurately learn and predict the damage to a building, even with a high-quality dataset. Predicting damage to buildings after a seismic event requires the use of advanced ML algorithms and techniques.
- The potential for uncertainty and error: despite the use of advanced algorithms and techniques, there is always a potential for uncertainty and error in the predictions made by the model. This can affect the reliability and usefulness of the predictions, and it may be necessary to incorporate additional sources of information or uncertainty measures to improve the accuracy of the model.

Nevertheless, it is essential to recognise that, before the widespread use and application of ML, earthquake research used to predict earthquakes, analyse seismic hazards and assess damage, among other things, using more traditional and statistical methods. These methods will not be discussed in this work, as there is a comprehensive review in the paper of Calvi et al. (2006). As stated by Xie et al. (2020), when compared to conventional methods, ML offers advantages in the handling complex problems, computation efficiency, treatment of uncertainties, and decision-making facilitation. More precisely, Xu et al. (2020) summarised that with ML algorithms there is:

- Greater flexibility in combining IMs: unlike traditional methods, which were restricted to comparing low-dimensional combinations of IMs, machine learning allows for the flexible combination of IMs as it can accept vectors of varying dimensions;
- Elimination of regression function assumption: meaning that the implicit definition of regression functions is not required, which lessens the introduction of judgmental assumptions into predictions;
- Clear criterion: the criterion for comparison is inevitably the capacity to interpret seismic damage;
- Enhanced adaptability and scalability: allows evaluating the performance of new IMs directly, without repeating nonlinear time history analysis tasks, by just adding them to the input vector and tracking changes in prediction accuracy.

Indeed, this assertion is not purely theoretical, empirical studies have demonstrated the enhanced efficacy of algorithmic models when juxtaposed with conventional methodologies, as exemplified by the research of Kalakonas & Silva (2022).

1.3. Motivation

In August 2021, I decided to apply for a scientific research grant. The public notice for the scientific research grant mentioned a very interesting field in computer engineering, ML combined with a crucial subject matter of civil engineering, as earthquakes. The urge to study this, regards to my deeply interest in contributing to the current state of the art, with this research being focused on predicting building damage.

This work arises from the growing importance of machine learning in our society, where we come across applications in various areas daily. The relevance of this study is clear, as it explores and compares different machine learning techniques and algorithms. Furthermore, the importance of the work extends to the literature in general, as it contributes to fill an existing gap in studies that combine machine learning with earthquake engineering, more particularly there are few studies that address the comparison of various algorithms in this specific area. This dissertation is not only motivated to make academic contributions, but also to have significant practical implications. Finally, the motivation and relevance entails applying machine learning to a crucial issue in society, namely the rapid and effective response after an earthquake, and also to a recurring and devastating event in physical, economic and social terms as observed in recent earthquakes (e.g., Turkey and Syria on February 6th, 2023; Morocco on September 8th, 2023; Afghanistan on October 7th, 2023;).

2. Overview of Machine Learning algorithms

Before delving further into the subsequent section of this dissertation, it is imperative to establish a foundational comprehension of machine learning algorithms.

Firstly, algorithms should be chosen based on the problem, and information available. Statistical learning problems can be allocated into one of two primary categories: supervised or unsupervised, that according to data characteristics and objectives of study can be categorised into classification or regression in case of supervised learning, or clustering and dimensionality reduction in case of unsupervised learning. (Hastie et al., 2021; Xie et al., 2020).

A more detailed explanation will be presented next:

- Supervised Learning: consists in building a model/function to forecast or estimate an output based on a collection of labelled data with one or more inputs. Meaning that, supervised data uses previous known outcomes to train a model that provides the most accurate approximation of the connection between input and output. After the model has been trained, it is possible to use it to predict outcomes from fresh, i.e., unlabelled, data.

    - Regression: in regression problems, the response variable is continuous, and the aim is to predict its value based on one or more predictor variables. To evaluate regression models' quality the metrics to be used are **(i)** Mean Absolute Error (MAE), **(ii)** Mean Squared Error (MSE), **(iii)** Root Mean Squared Error (RMSE), **(iv)** R-squared (R^2) and **(v)** Standard Deviation (SD), that will be explained in detailed in chapter 4.5. Evaluation metrics. Apart from R^2, the smaller the metrics the better is the predictive accuracy (i.e., ability to provide precise predictions that closely aligns to the observed real-world data).

    - Classification: in classification problems, the answer variable is categorical, and the objective is to determine to which category it belongs based on one or

more predictor factors. In classification models, metrics such as *Accuracy*[1], *Precision*, *Recall*, F1 Score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) are commonly employed to assess the model's performance in categorizing data into distinct classes.

- Unsupervised Learning: Statistical learning in which patterns and correlations in data are discovered without a predefined output variable to predict. In simple terms, a collection of unlabelled data is employed to learn/infer about the data's underlying structure.

Furthermore, when dealing with machine learning models they can also be assessed based on their efficiency, which refers to their ability to perform tasks with minimal resource consumption (processing power, memory used and data storage, processing time, quantity of data required, and other social and monetary resources needed), and explainability, which involves the model's ability to provide clear and understandable explanations for its decisions.

In conclusion, according to the dataset presented in this thesis (which will be elaborated upon in Chapter 4. Methodology), the appropriate choice for the predictive task, considering the intended output and comparison, should be supervised learning. Given that the target output variable in the dataset, maximum displacement, exhibits a continuous range of values, this constitutes a regression problem.

Below is a brief description of all the chosen regression algorithms (Ridge Regressor, Lasso Regressor, Support Vector Regressor, Decision Tree, Random Forest, Gradient Boost, Extreme Gradient Boost, Artificial Neural Networks and Multi-layer Perceptron), based on the books of Aurélien Géron (2022) and Hastie et al. (2021).

[1] It is imperative to distinguish the *Accuracy Metric* (proportion of correct predictions out of the total predictions made) for classification models, from the overall accuracy concept in forecasting.

2.1. Ridge Regressor

Ridge regressor (RR) is a statistical technique used to estimate the coefficients of a linear regression model. It is a type of regularized regression that adds a penalty parameter to the ordinary least squares (OLS) regression cost function. The penalty parameter is proportional to the square of the magnitude of the coefficients, which means that it shrinks the coefficients towards zero, and is usually chosen through cross-validation. The benefit of using RR is that it can prevent overfitting[2] of the model, especially when dealing with high-dimensional datasets or datasets with multicollinearity, where the predictors are highly correlated. The penalty term reduces the variance of the coefficient estimates, leading to a better generalization performance of the model. Overall, Ridge regressor is a useful tool for building predictive models that can handle complex datasets while avoiding overfitting.

2.2. Lasso Regressor

Lasso regressor (LR) is also a statistical technique used to estimate the coefficients of a linear regression model. It is another type of regularized regression that adds a penalty term to the OLS regression cost function, being also usually chosen through cross-validation. The penalty term in LR is the absolute value of the magnitude of the coefficients, which can lead to some of the coefficients being exactly equal to zero. Like RR, the LR can prevent overfitting of the model and handle high-dimensional datasets or datasets with multicollinearity. However, the LR has the additional benefit of feature selection by shrinking the coefficients of less important predictors to exactly zero. Lasso regression is a powerful tool for building predictive models that can handle complex datasets while also providing feature selection capabilities.

2.3. Support Vector Regressor

Support vector regressor (SVR) is a technique used in machine learning to predict the values of a continuous response variable. The method involves breaking the response variable down

[2] Overfitting occurs when a model becomes excessively tailored to the training data, resulting in difficulties when applying it to unfamiliar data while testing. In this case, the model excels on the training data but exhibits poorer performance on the testing data due to its focus on learning specific intricacies and even noise within the training data rather than capturing the underlying patterns (Ying, 2019).

into sectors or intervals and constructing a separate linear regression model for each sector. By doing so, the model can better capture the unique patterns and trends present within each sector, leading to more accurate and reliable predictions. These models are then combined to produce a single, piecewise-linear function that can accurately predict the response variable for any given set of predictor variables.

One of the main benefits of SVR is its flexibility in modelling the relationship between predictor variables and the response variable. Unlike traditional linear regression models, SVR can handle nonlinear and complex relationships between variables, making it a valuable tool in many predictive modelling scenarios.

However, it's important to note that SVR can be computationally demanding, especially when working with large datasets or high-dimensional predictor variables. Therefore, researchers should carefully consider the trade-offs between model complexity and computational efficiency when deciding whether to use this technique.

Overall, support vector regression is a powerful technique for predicting continuous response variables, particularly in cases where traditional linear regression methods may not be sufficient. By breaking the response variable down into sectors and constructing separate models for each sector, sector vector regression is able to capture the unique patterns and trends within the data, resulting in a more accurate and reliable prediction mode.

2.4. Decision Tree Regressor

A decision tree regressor (DTR) is a type of model used in regression analysis. It works by recursively partitioning the input space into regions that correspond to different levels of the output variable. At each internal node of the tree, a decision is made based on one of the input variables, splitting the data into two subsets that are as homogeneous as possible with respect to the output variable. To make a prediction for a new observation, the decision tree starts at the root node and follows the path that corresponds to the combination of input variable values until it reaches a leaf node, which contains the predicted value for the output variable. The predicted value is typically the average of the training samples that belong to the same leaf node.

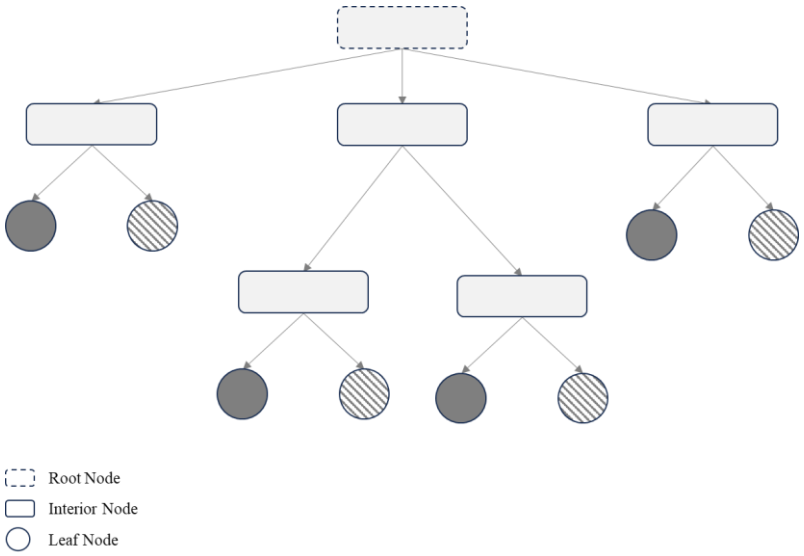

Figure 1. Diagram of a (simple) Decision Tree algorithm[3]

One advantage of the DTR is its ability to capture complex interactions and nonlinear relationships between the input and output variables. However, if the tree is too large or the stopping criterion is not well tuned, it can overfit the training data and lead to poor generalization performance on new data. To address this issue, various regularization techniques can be used, such as pruning, bagging, boosting, or random forests, which combine multiple decision trees to improve the generalization performance and reduce the variance of the prediction.

2.5. Random Forest Regressor

A Random forest regressor (RFR) is a popular and powerful machine learning decision tree algorithm. It works by constructing a multitude of decision trees at training time and outputting the class or mean prediction of the individual trees. Each tree in a random forest is constructed using a random subset of the original dataset, which is called a bootstrap sample. Additionally, each split in the tree is made using a random subset of the features. These two

[3] The referred diagram is purely illustrative and does not directly represent the application discussed in the present work. Its purpose is to elucidate the operational principles of the algorithm under consideration in a simplified manner.

randomization techniques ensure that each tree in the forest is diverse and reduces the risk of overfitting to the training data.

At prediction time, the random forest algorithm (in general) aggregates the predictions of all the trees in the forest, either by majority voting (in classification tasks) or averaging (in regression tasks), to output the final prediction.

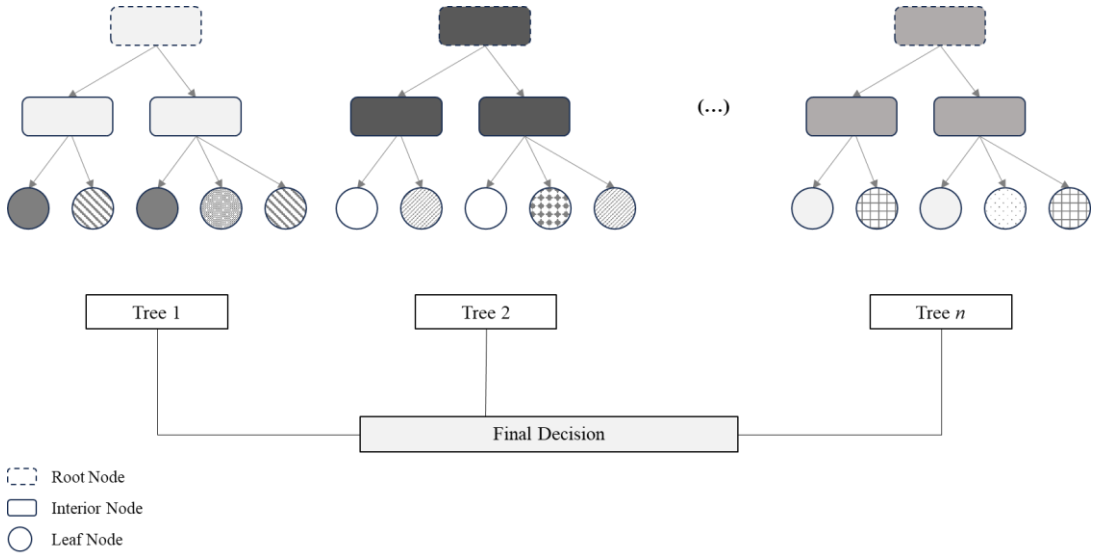

Figure 2. Diagram of a Random Forest algorithm[4]

One of the advantages of random forest is its ability to handle a large number of features, even when the number of samples is relatively small. It is also less prone to overfitting than other decision tree algorithms and is relatively easy to interpret and tune. In general, RFR is a highly effective and versatile algorithm.

[4] The referred diagram is purely illustrative and does not directly represent the application discussed in the present work. Its purpose is to elucidate the operational principles of the algorithm under consideration in a simplified manner.

2.6. Gradient Boost Regressor

A Gradient Boost Regressor (GBR) is typically a decision tree that splits the data recursively based on informative features. It works by combining several simple models to create a more accurate final model. To start, the algorithm creates an initial model that makes a guess based on the average outcome for all the data points. Then, it calculates the difference between the actual outcomes and the initial predictions. After making initial predictions, the algorithm builds a new model that predicts the differences between the actual outcomes and the initial predictions. The algorithm then enhances the influence of this model on the final predictions by assigning it a weight, a process called "boosting". The algorithm repeats these steps several times, creating new models to predict the residuals of the previous step and adding them to the ensemble. The final prediction is made by combining the predictions of all the models, with each model's weight reflecting its ability to reduce the errors of the previous models.

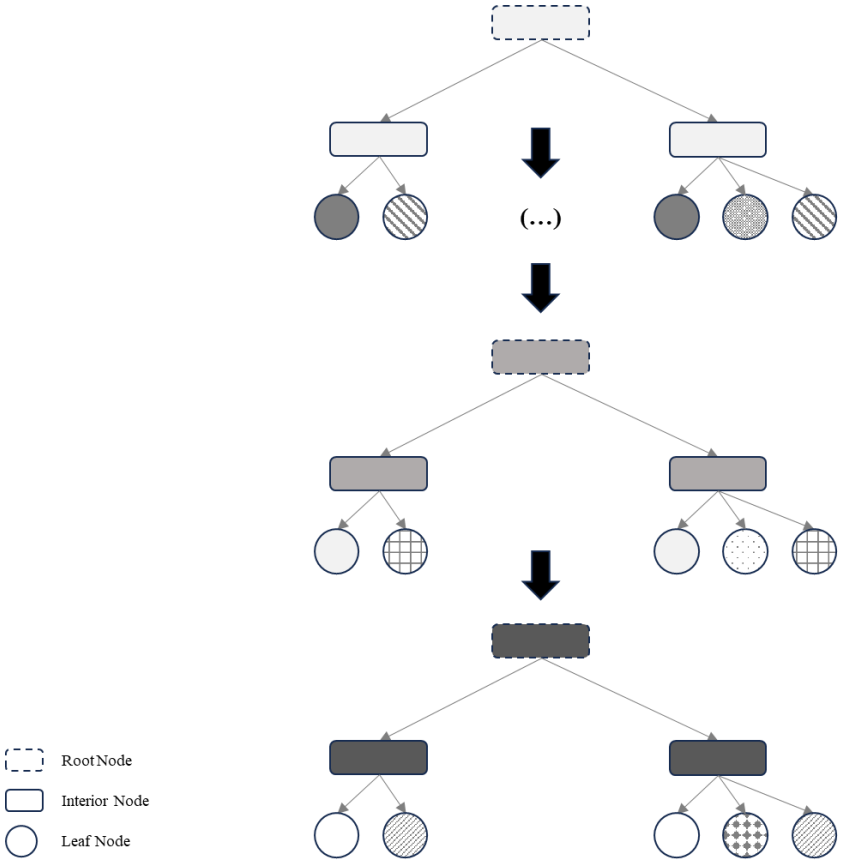

Figure 3. Diagram of a Gradient Boosting algorithm[5]

[5] The referred diagram is purely illustrative and does not directly represent the application discussed in the present work. Its purpose is to elucidate the operational principles of the algorithm under consideration in a simplified manner.

Gradient boosting can manage diverse data types and missing values, but it may overfit if the models are too intricate or if there is noise in the data, so it is necessary to tune the hyperparameters and monitor model performance during training.

2.7. Extreme Gradient Boosting Regressor

Extreme gradient boosting regressor (XGBR) is a decision tree-based machine learning algorithm that is designed to be highly scalable and efficient. It is based on the idea of gradient boosting, where each subsequent model is trained on the residual errors of the previous models. XGBR builds decision trees in a sequential manner, where each tree is designed to correct the errors of the previous tree.

One key feature of XGBR is its use of a regularized objective function that includes both a loss term and a penalty term. The loss term measures the difference between the predicted and actual values, while the penalty term helps to control the complexity of the model and prevent overfitting. XGBR supports several different types of penalty functions, including L1 (alpha parameter – Lasso regularization) and L2 (lambda parameter – Ridge regularization) regularization, as well as tree-related penalties.

Another important feature of XGBR is its ability to handle missing data and categorical features, through a process called *split finding*, meaning that it uses a novel algorithm for finding the best split points for each node in the decision tree, which is based on a greedy search[6] that evaluates all possible split points in a computationally efficient manner.

XGBR also supports parallel processing on multiple cores and multiple machines, which allows it to scale to very large datasets. The algorithm has been shown to perform well on a wide range of tasks and has won several machine learning competitions (Chen & Guestrin, 2016).

[6] Greedy search is the process optimization by choosing the most suitable feature and split point at each node of the tree. It focusses on making locally optimal choices without taking into account the overall structure of the tree. This process is not related to hyperparameter tuning (Chen & Guestrin, 2016).

2.8. Artificial Neural Networks

Feedforward neural networks are the most common type of artificial neural networks (ANN) used in statistical learning. These networks consist of a set of input nodes, one or more hidden layers of computational nodes, and an output layer of nodes that produce the final predictions. The connections between nodes are weighted, and each hidden and output node applies an activation function to the weighted sum of its inputs. A representation of an ANN is presented in the next figure:

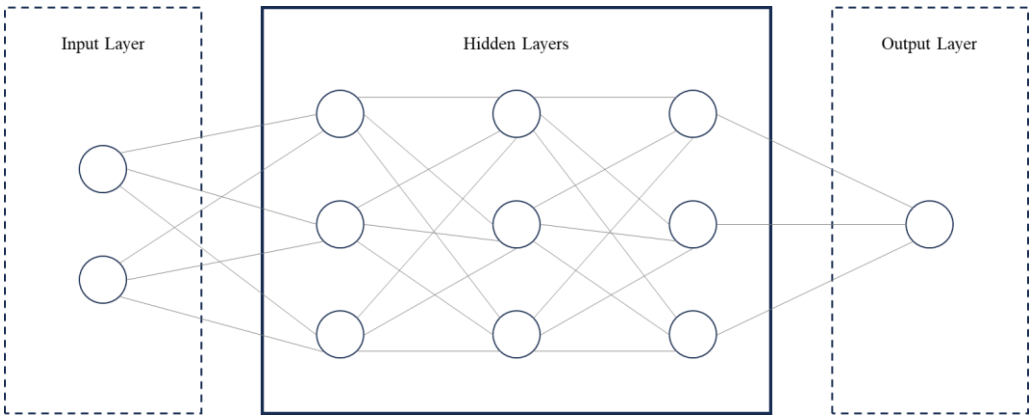

Figure 4. Layered Structure Diagram of a Neural Network[7]

The input nodes receive the features of a training example and pass them through to the first hidden layer, where the weighted sum of the inputs is computed and passed through an activation function, such as the sigmoid, hyperbolic tangent, or rectified linear unit (ReLU) function. The outputs of the first hidden layer then become inputs to the next hidden layer, and this process continues until the output layer produces the final predictions.

To train a feedforward neural network, we typically use backpropagation, which involves computing the gradient of the loss function with respect to the weights in the network and updating the weights accordingly. This process is iterated multiple times until the weights converge to their optimal values. Feedforward neural networks are powerful models that can learn complex relationships between inputs and outputs, making them useful for a variety of prediction tasks.

[7] This figure is a simplification of the ANN utilized in this research, and that the neurons within each layer are intentionally omitted for illustration purposes. This deliberate simplification is made to ensure that the visualization remains clear, concise, and understandable.

2.9. Multi- Layer perceptron

The Multilayer Perceptron (MLP) is a type of Artificial Neural Network (ANN) that consists of multiple layers of neurons, each connected to the next layer. These layers are known as the input layer, one or more hidden layers, and the output layer. The architecture of the MLP allows it to learn complex relationships in the data, making it particularly suitable for more intricate nonlinear problems that cannot be efficiently solved by simpler models.

The multilayer perceptron algorithm (MLP) is also an ANN algorithm, with the particularity of using an interesting library of the Scikit-Learn. In this dissertation, the MLP was only used because it's an easier way to implement neural networks in just a few lines of code. Nonetheless, according to Aurélien Géron (2022), although Scikit-Learn is convenient, the neural net features are limited, therefore many prefer to use the Keras library (i.e., the common ANN above).

3. State of the art

As stated in the previous chapter, the detection of damage caused by earthquakes in buildings immediately after their occurrence using ML algorithms is a growing field of interest, although research has been mainly concentrated in the recent years. Xie et al. (2020) presented one of the most extensive state of the art reviews on ML within the field of earthquake engineering.

The importance of Xie et al. (2020) study is to highlight the potential of the role of ML in a wider variety of studies regarding seismic events. At the date of the publication of the paper, the authors stated that nearly 200 pertinent publications were available combining the two fields (Earthquake engineering and Machine Learning). Moreover, the paper covers several areas related to the study of earthquakes, namely (i) *seismic hazard analysis*, which predicts ground shaking and associated uncertainties, (2) *system identification and damage detection* that spans a broad spectrum of studies, from classifying component failures in controlled experiments to detecting extensive structural damage on a larger scale, (3) *seismic fragility assessment*, involving the estimation of structural failure probabilities under seismic conditions, and (4) *structural control methods* for earthquake mitigation using active and semi-active techniques to reduce vibrations induced by earthquakes.

The area of earthquake engineering of Xie et al. (2020) review that most closely aligns with the area of the work described in this dissertation is the one labelled as *System identification and damage detection*[8], more precisely using numerical[9] data. Overall, in Xie et al. (2020) review it was concluded that Artificial Neural Networks, Support Vector Machine, Response Surface Model, Logistic Regression, Decision Trees, Random Forest and hybrid methods emerge as the predominantly employed techniques in this research field (Table 1). Additionally, and on a different perspective, there is also the literature review of Harirchian et al. (2021) and Mangalathu et al. (2020)[10].

[8] Please note that many areas defined by Xie et al. (2020) may, or not, overlap between themselves.
[9] Numerical data consists in quantitative values that can be measured or counted. An alternative is photographic data that is also used within the earthquake engineering field using ML algorithms for damage detection.
[10] Harirchian et al. (2021) reviews the assessment of visual safety and classification of damage in existing buildings. Mangalathu et al. (2020) analyses the applications of machine learning in structural building design

Table 1. A survey of the ML methods' applications across four areas within the field of earthquake engineering

	ANN	SVM	RSM	LogR	DT&RF	Hybrid	Other	**Total**
Seismic Hazard Analysis	21	6	1	1	5	9	17	**60**
System Identification and Damage Detection	29	9	2	3	5	14	20	**82**
Seismic Fragility Assessment	10	6	30	13	2	-	6	**67**
Structural Control for Earthquake Mitigation	19	-	-	-	-	7	-	**26**
Total	**79**	**21**	**33**	**17**	**12**	**30**	**43**	**235**

ANN - Artificial Neural Networks; SVM - Support Vector Machine; RSM - Response surface model; LogR - Logistic Regression; DT&RF - Decision Tree and Random Forest.

Source: *Xie et al. (2020, p.4)*

As technology continues its rapid progression, marked by a significant evolution in computational capabilities over the past 10-20 years (for instance, Goh (1994), De Stefano et al. (1999) and Nakamura et al. (1998) conducted pioneering research on machine learning algorithms in the earthquake science field in the 90's), we conducted an extensive review, searching by keywords in the B-On portal, of the current state of the art. The goal was to assemble a relevant body of contemporary research for comparative analysis alongside the proposed objective within this dissertation. Specifically, the aim within this dissertation was to review papers published from 2020 onwards that exclusively addressed the topic of damage detection.

Therefore, starting with the analysis of Xu et al. (2020) research where the authors proposed an accurate and efficient real-time seismic damage prediction method based on ML algorithms and multiple (48) intensity measures. Using three algorithms, Logistic Regression, Support Vector Machine, and Decision Tree, a seismic damage prediction was performed for 12 different building classes on a total of 435 buildings in the same region. The conclusions were that that the Logistic Regression algorithm has a high accuracy and efficiency, the Support Vector Machine algorithm is accurate but has a relatively lower efficiency. Accordingly, the Decision Tree algorithm is efficient and explainable, but has relatively

and performance. Both articles contribute to the advancement of building assessment and design techniques, offering valuable information for civil engineering and building safety.

lower prediction accuracy. Nevertheless, Xu et al. (2020) resorted to a categorical output variable, hence the results are bounded to a classification problem.

In addition, Hwang et al. (2021) proposed both regression-based and classification-based methods to infer the damage status of reinforced concrete structure buildings. Within the regression-based ML techniques the authors used six methods (Multiple Linear Regression, RR, DTR, RFR, Adaptive Boosting and XGBR) to predict the maximum story drift[11].The authors concluded that XGBR has a superior performance compared to the other algorithms and the Multiple Linear Regression and RR models were not able to obtain good results.

Kalakonas & Silva (2022) suggest the use of artificial neural networks (ANNs) to predict the seismic vulnerability of building classes. The authors point out the weaknesses of existing vulnerability models, which are based on empirical relationships between building attributes and damage and suggest that ANN might improve the accuracy and efficiency of seismic risk assessments. For that purpose, they trained an ANN on a dataset of different building types with different features in order to predict the seismic susceptibility of building classes, as opposed to the majority of the research that focus either on a single structure type or component. The results of the paper also show that the suggested technique is useful in predicting seismic vulnerability (the capacity of structures, such as buildings, to withstand damage from earthquakes), with higher accuracy than other traditional models. According to the authors, the ANN method is computationally efficient and may be used for large-scale building classes for quick seismic risk assessment.

Moreover, Demertzis et al., (2023) studied and compared the performance of an extensive list of 15 ML algorithms (Light GBR, GBR, RFR, Extra tree regressor, k-Nearest Neighbours Regressor, Linear Regression, Bayesian Ridge, RR, DTR, AdaBoost Regressor, LR, among others) to predict the damage after an earthquake in 30 buildings, resorting to 14 ground motion parameters (value range corresponding to 65 earthquakes) and 4 structural parameters. In terms of performance, the Light GBR showed the best results when compared to the other algorithms. One interesting option of this paper is that although it encompasses a

[11] Maximum story drift differs from Maximum displacement. The former refers to the greatest relative horizontal movement between two consecutive floors of a building during a seismic event, demonstrating the extent to which floors shift or drift horizontally in relation to each other. The latter pertains to the maximum movement experienced by a specific point, such as the top of a building, during an event like an earthquake, encompassing horizontal, vertical, or rotational motion.

multitude of algorithms it does not study artificial neural networks, which were already proven to be the most popular method applied to ML in earthquake engineering. According to Demertzis et al., (2023), their intent was to assess damage response with adequate trustworthiness by not adopting to black box ML methods that are not possible to be understood.

Alcantara & Saito (2023) study also focuses on using ML methods to predict the damage condition of reinforced concrete of 600 buildings after an earthquake, using 30 variables of input with inter-story drift ratio as the output. A particularity of this research is the reassortment to sensor data responses, both ground or/and roof of the building. The authors used Linear Regression, DTR, RFR, AdaBoost Regressor, XGBR, MLP, and GBR, being the latter the method that achieved the highest performance (i.e., R^2) among the other ML methods.

It is possible to concluded that, although the state-of-the-art review highlighted that ANN has been widely used for various applications, it is fair to say that other algorithms, particularly Gradient Boost, are becoming increasingly popular due to their strong performance and simplicity when compared to ANN.

An overview of the previous studies, with more details on variable selection and other related key findings, can be seen below in Table 2.

Table 2. Overview of the recent State of the Art in Machine Learning applications for earthquake damage detection

Authors	ML Task	ML Algorithms	Number of Inputs	Dataset Division	Hyper parameterization
Xu et al. (2020)	Classification	Sector Vector Machine; Logistic Regression; and Decision Tree.	48	Hold out: 70/30	Grid Search Cross - Validation
Key findings: This study emphasizes the importance of carefully selecting intensity measures (IMs) for precise seismic-damage prediction. Machine learning algorithms, such as the Logistic Regression algorithm has high accuracy and efficiency, meanwhile Support Vector Machine is accurate but is relatively less efficient. The decision tree algorithm easier to understand, although it has lower prediction accuracy. Moreover, while nonlinear time-history analysis (NLTHA) offers accuracy, it comes at a computational cost, whereas fragility analysis (FA) and capacity spectrum method (CSM) sacrifice accuracy for efficiency. The suitability of peak ground acceleration (PGA) as an IM is questionable, and this study's proposed method facilitates IM performance comparisons across diverse building types and regions.					
Hwang et al. (2021)	Regression and Classification	**Classification**: Naive Bayes; k-Nearest Neighbours; Decision Tree; Random Forest; AdaBoost and; Extreme Gradient Boosting. **Regression**: Multiple Linear Regression; Ridge Regression; Decision Tree Regression; Random Forest Regression; Adaptive Boosting Regression; and Extreme Gradient Boosting Regression.	16	Hold out: 70/30	Hand Tunning
Key findings: Overall, the paper concludes that achieving a robust seismic vulnerability assessment for modern reinforced concrete frame buildings requires considering modelling uncertainties at both component and system levels. Notably, boosting algorithms, such as adaptive boosting and extreme gradient boosting, outperform other techniques in both response prediction and collapse status classification. Additionally, the study highlights the significant impact of uncertain modelling parameters, particularly those related to reinforced concrete beam properties, on seismic response predictions based on capacity design principles.					
Demertzis et al. (2023)	Regression	Light Gradient Boosting Regressor; Gradient Boosting Regressor; Random Forest Regressor; Extra Tree Regressor; k-Nearest Neighbours Regressor; Linear Regression; Bayesian Ridge; Ridge Regression; Decision Tree Regressor; AdaBoost Regressor; Elastic Net; Lasso Regression; Orthogonal Matching Pursuit; Huber Regressor; and Least Angle Regression.	18	5-fold Cross - Validation	Hand Tunning
Key findings: Some of the authors most important key findings are that the most important seismic and structural parameters for predicting the seismic response of reinforced concrete (R/C) buildings are the height of the building, the number of stories, the structural eccentricity, and the ratio of base shear received by R/C walls. Moreover, the Shapley Additive Explanations (SHAP) method can be used to explain the contribution of each individual feature to the final prediction, and to investigate the general relationship between feature value and impact on the prediction. From a vast list of algorithms, the Light Gradient Boosting Regressor demonstrated its robust applicability, stability, and effective noise handling through the Gradient One Side Sampling technique and predictive enhancement via tree segmentation, making it a cost-effective solution for complex spatial-temporal problems in machine learning					
Kalakonas & Silva (2023)	Regression	Artificial Neural Networks	14	10-fold Cross - Validation	Hand Tunning
Key findings: The authors' main conclusions highlight that Artificial Neural Networks demonstrate superior performance compared to conventional methods, offering potential improvements in reliability and accuracy for scenario and probabilistic seismic risk assessments. Additionally, they underscore the limitations of Single Degree of Freedom (SDOF) oscillator models in capturing complex features like higher modes of vibration and torsional effects. The study also emphasizes the utility of damage and consequence models for estimating the likelihood of exceeding specific damage states and conditional Loss Ratios (LRs) based on Expected Displacement Performances (EDPs). Furthermore, the research underscores the significance of accounting for uncertainty in Intensity Measures (IMs) and Expected Displacement Performances (EDPs) when evaluating the seismic vulnerability of building portfolios.					
Alcantara & Saito (2023)	Regression	Linear Regression; Decision Tree Regressor; Random Forest Regressor; Gradient Boosting Regressor; AdaBoost Regressor, Extreme Gradient Boosting Regression, and Multi-Layer Perceptron.	30	Hold out: 80/20	Hand Tunning
Key findings: The main conclusions were that the Random Forest and Gradient Boosting proved to be most accurate algorithm for inter-story drift ratio prediction. Moreover, the significance of intensity measure (IM) levels surpassed structural features, highlighting record variability's role in model accuracy. The key IMs to predict building damage came from ground and roof sensors and were based in acceleration and velocity. The others stated that future studies should increase earthquake record diversity to encompass more earthquake features.					

4. Methodology

This section describes the methodology used and the sample dataset used. The sample dataset selection is an adaptation of the sample of Kalakonas & Silva (2022) and the methodology is in line with various works such as Alcantara & Saito (2023), Demertzis et al. (2023) and Hwang et al. (2021) because they too make a comparison between algorithms. Moreover, in this research the algorithms are programmed resorting to very well know libraries, namely Tensor Flow[12] for artificial neural networks, and scikit-learn[13] for all the other algorithms.

4.1. Materials

In this dissertation it used a computer with the following characteristics:

- Processor: *Intel Core i9 10900K 10-Core (3.7GHz-5.3GHz);*
- RAM: *4x Kingston 32GB DDR4 3200MHz HyperX Fury Black;*
- Graphics Card: *ZOTAC GAMING GeForce RTX 3090 Trinity OC 24gb.*

4.2. Dataset Description

Firstly, it must be borne in mind that this research does not intend to select and collect primary IMs, pre-process ground movements and gathering and modelling of class structures, but only to process an initial dataset and subsequently apply and test various machine learning algorithms in order to compare results, as it is the motivation of this research. Therefore, the following data was provided by the work of Martins & Silva (2021) and Kalakonas & Silva (2022).

[12] https://www.tensorflow.org/
[13] https://scikit-learn.org/

4.2.1. Building Classes

In the previous chapter 3. State of the Art, it was possible to conclude that although recent research has been taking into account the heterogeneity of buildings, some research (e.g., Alcantara & Saito (2023)) does not. More precisely, Kalakonas & Silva (2022) stated that the majority of the studies only focus on either a single structure or a component of a structure, neglecting the wide range of aleatory and epistemic uncertainties concerning each building. Therefore, it was decided that in this research, predicting damage to a building after a seismic event is conducted considering different building classes as well.

The categorization of each building type is conducted according to a set of structural attributes, namely: **(i)** the construction material, **(ii)** the lateral load resisting system[14], **(iii)** the number of storeys, and **(iv)** the seismic design code level[15].

(i) Regarding the material of construction in this sample there are 3 categories: reinforced concrete with infilled frames, confined and unreinforced masonry.

(ii) Lateral load resisting system can be classified in wall systems or infilled frames. Wall systems refer to building structures in which walls serve as the primary load-bearing element (vertically and laterally). Infilled frames refer to buildings that utilize a skeletal frame structure as the main load-bearing component, with non-load-bearing infill materials for the exterior walls.

(iii) Concerning the number of stories, they vary from 1, 2 and 4 stories.

(iv) Seismic design code in this dataset can either be (i) low ductility (when there is limited seismic provisions (for instance, in informal construction or structures built prior to 1960), (ii) non-ductile (for the unreinforced masonry structures, which are expected to perform poorly), and lastly (iii) no seismic provision (does not have specific design features or measures in place to withstand or resist seismic forces)

[14] As the name indicates are systems or components within a building or structure designed to withstand and counteract lateral forces. Those forces can be attributed to wind, seismic activity (earthquakes) or human activities.

[15] Seismic design code level, or just seismic design, states the degree on which structures where designed based on the compliance of seismic provisions.

Ultimately this categorisation lead to a set of 7 building types capturing the most common construction styles in the Balkan region, used in Kalakonas & Silva (2022), where the structural and dynamic properties of them were obtained through numerical modelling in Martins & Silva (2021). Below is a table summarizing the classes (Table 3).

Table 3. Building Class description

Class Name	Material	LLRS[1]	Storeys	Seismic Design
CR_LFINF-CDN-0_H2	Reinforced concrete	Infilled frames	2	No seismic provisions
CR_LFINF-CDN-0_H4	Reinforced concrete	Infilled frames	4	No seismic provisions
MCF_LWAL-DUL_H1	Confined masonry	Wall	1	Low ductility
MCF_LWAL-DUL_H2	Confined masonry	Wall	2	Low ductility
MUR_LWAL-DNO_H1	Unreinforced masonry	Wall	1	Non-ductile
MUR_LWAL-DNO_H2	Unreinforced masonry	Wall	2	Non-ductile
MUR_LWAL-DNO_H4	Unreinforced masonry	Wall	4	Non-ductile

4.2.2. Input and Output Variables

According to Xu et al. (2020) it is crucial to not overly rely on structural characteristics of buildings as inputs, as it can compromise the applicability and universality of the algorithm.

To unlock the full potential of machine learning algorithms and achieve robust regression outcomes, a large dataset encompassing information from multiple seismic events is indispensable. In the scope of this dissertation, the dataset provided by Kalakonas & Silva (2022) is sourced from multiple repositories. It encompasses data from the Pan-European Engineering Strong-Motion Database (ESM), which aggregates ground motion data from Europe and the Middle East. Additionally, it incorporates information from the NGA-West2 database, which captures ground motions recorded of worldwide shallow crustal[16] earthquakes in active tectonic regimes, aligning with the predominant tectonic regime in the Balkan region and majority of ESM database. Moreover, it also includes ground motion data from the K-NET (ground motion data from Japanese earthquakes), specifically filtered for

[16]Shallow crustal earthquakes are seismic occurrences situated at a relatively shallow depth within the earth's crust, usually less than 70 kilometers. Owing to their closeness to the surface, these earthquakes can exert more direct and potentially harmful effects when contrasted with deeper seismic events.

shallow crustal earthquakes. In summary (prior to pre-processing the data), 1688, 488, and 1070 entries, corresponding to different earthquakes (seismic events) were extracted from, ESM, NGA-West 2 and K-NET databases, respectively. A detailed view of the importance of each repository is seen in Table 4.

Table 4. Seismic events per Building Class and respective Repository Sources.

Class Name	ESM	NGA -West2	K-NET	Total Entries[1]
CR_LFINF-CDN-0_H2	1 678	459	1 036	**3 173**
CR_LFINF-CDN-0_H4	1 676	443	1 037	**3 156**
MCF_LWAL-DUL_H1	1 688	488	1 059	**3 235**
MCF_LWAL-DUL_H2	1 688	485	1 059	**3 232**
MUR_LWAL-DNO_H1	1 688	481	1 041	**3 210**
MUR_LWAL-DNO_H2	1 685	473	1 044	**3 202**
MUR_LWAL-DNO_H4	1 683	468	1 041	**3 192**

[1]Each entry corresponds to a different seismic event (after the pre-processing of data – explained in the next sub-chapter).

The initial (raw) input variables comprised of Arias Intensity (total energy imparted to the ground over the duration of an earthquake), PGA, and SA in various ranges. Regarding the output, or the predicting variable, in accordance with majority of literature it is being considered the Maximum Displacement. More precisely, the peak displacement of single-degree-of-freedom oscillators, that is the farthest distance a component of a structure or system moves from its original (equilibrium) position during a dynamic event (ground motion), concerning a single degree of movement (a 2D vector, a back-and-forth movement of the building). The maximum displacement of the SDOF oscillators per ground motion was computed using nonlinear dynamic analysis described in detail in the work of Martins & Silva (2021). Pass a certain limit of the displacement the buildings are believed to suffer damage. The limits are indicated in Table 5 below.

Table 5. Damage limit per Building Class

Class Name	Limits (cm)	MD[1] > Limit	MD < Limit	Total Entries[2]	Damage (%)
CR_LFINF-CDN-0_H2	1.4	667	2 506	3 173	21%
CR_LFINF-CDN-0_H4	3.8	356	2 800	3 156	11%
MCF_LWAL-DUL_H1	0.3	105	3 130	3 235	3%
MCF_LWAL-DUL_H2	0.7	332	2 900	3 232	10%
MUR_LWAL-DNO_H1	0.3	392	2 818	3 210	12%
MUR_LWAL-DNO_H2	0.6	654	2 548	3 202	20%
MUR_LWAL-DNO_H4	1.3	546	2 646	3 192	17%

[1]Maximum Displacement

[2] Each entry corresponds to a different seismic event (after the pre-processing of data – explained in the next sub-chapter).

4.3. Pre-processing

Prior to start the training of any algorithm, pre-processing of the data is of utmost importance, encompassing essential steps such as cleaning, normalization, and splitting.

4.3.1. Data Cleaning

The initial phase of data pre-processing involved a cleansing operation to solve issues such as blank, duplicate, and null values. Removing duplicates ensures that each entry in the dataset is unique, avoiding distortions in analysis and inference caused by repeated rows and eliminating null values ensures that the data is complete, preventing distortions in calculations and models. This was achieve using the Pandas library. Additionally, a feature (variable) selection process was undertaken, employing an iterative trial-and-error methodology.

The initial (raw) dataset consisted of 29 variables, including 1 variable for Arias Intensity, 1 variable of PGA, and 27 variables of SA ranging from [0.01 to 3.0] seconds. However, to avoid the dimensional curse and overfitting, several trial-and-error tests were conducted, leading to the exclusion of variables that did not contribute positively to the model's performance. This resulted in an intermediate version, denoted as S-19V, which comprises 19 variables: 1 PGA and 18 SA variables for periods in the range of [0.25 to 2.0] seconds. Consequently, following multiple iterations of trial-and-error, the dataset was ultimately condensed into a more concise form (S-10V), which includes only 10 variables (1 PGA and 9 SA variables spanning [0.3 to 1.6] seconds). The choice of 10 variables stems from an

optimization process, wherein various combinations were systematically tested. This number represents the optimal balance, maximizing efficiency without compromising the model's predictive capacity. Further reduction would significantly jeopardize the accurate representation of variable relationships, adversely impacting the model's quality.

This streamlined dataset facilitates more accurate modelling, eliminates excessive variables, and enables more efficient and fast computational operations. By going through these steps, the data cleaning process has contributed to the creation of a more practical dataset, optimizing the efficiency of subsequent algorithmic operations.

4.3.2. Data Transformation

Starting with normalization, according to Aurélien Géron (2022), it is one of the most important transformations needed on a dataset. Shalabi et al. (2006) even suggests that the *min-max normalization* consistently had the highest accuracy. Note that the normalization of the data is a common process in data science and statistical analysis that consists of transforming numerical values into a common scale, in this way it is ensured that all features have the same scale. The *min-max normalization* is a data preprocessing technique that transforms the variables for a specific interval normally between 0 and 1.

Furthermore, the maximum displacement variable was logarithmically transformed, so that the variance of the variable can be stabilised. This technique reduces outliers' impact, especially in skewed distributions, by compressing values on higher scales. The user must be aware that interpreting the post-transformation values requires considering the introduced scale changes (Curran-Everett, 2018).

4.3.3. Data Splitting

Géron (2022) mentions that it is common to use an 80/20 split for training and testing data. This technique involves dividing the available data into two parts, where 80% is used for training the machine learning model, and the remaining 20% is used for testing. If the model performs well on the testing set (also known as holdout set), it is an indication that the model has generalized well and can make accurate predictions on new data. On the other hand, if the model performs poorly on the testing set, it is an indication that the model needs to be improved. This method is known as holdout validation.

4.4. Hyperparameter tunning

Many authors agree that the most important task to obtain great results is the optimization process and the chosen hyperparameters for each algorithm (Demertzis et al., 2023). Moreover, Yu & Zhu (2020) mention that grid search is the most straightforward search algorithm that leads to the most accurate predictions as long as sufficient computational resources are given.

As show in chapter 4.1 Materials, the necessary computational resources of sufficient capacity to expediently undertake grid search were available to apply this technique. More precisely for grid search cross-validation that entails a substantial computational cost.

In detail, grid search cross-validation is a hyperparameter optimization technique used in machine learning to systematically search for the best combination of hyperparameters for a given model (Géron, 2022). Grid search cross-validation works by defining a grid of possible values for each hyperparameter and then evaluating the model's performance on each combination of hyperparameters using cross-validation. Meanwhile, cross-validation involves splitting the data into training and validation sets multiple times and evaluating the model's performance on each split. For that, in this dissertation only the 80% of training data previously split was used, so that the model would not be biased with data that had already been seen.

4.5. Nonlinear dynamic analysis

As shown in Table 6 the amount of data pertaining to non-damaged structures significantly surpasses that of structures exhibiting damage. To mitigate potential biases in the outcomes, since input data for non-damaged cases exhibit a linear relationship with the output, whereas this linearity disappears beyond the threshold of damage, it was decided to balance data by only selecting the same number of linear (non-damage) and nonlinear (damage) data (Kalakonas & Silva, 2022). This method represents a different approach, as the literature review did not unequivocally establish whether previous authors had undertaken nonlinear dynamic analyses into their results.

It is to be noted that the 50/50 composition of the dataset (50% linear and 50% nonlinear) will only be performed in the dataset of 10 input variables (S-10V) as it is only justifiable due to computational constraints. Also, based on the observation of nonlinear data, algorithms based on linear regressions are upfront not expected to perform the best. More details regarding this will be explained in chapter 5.

Table 6. Damage limit per Building Class and corresponding Total Entries vs. Balance Entries

Class Name	Limits (cm)	MD[1] > Limit	MD < Limit	Total Entries	Balanced Entries	Damaged (%)
CR_LFINF-CDN-0_H2	1.4	667	2 506	3 173	1 334	50%
CR_LFINF-CDN-0_H4	3.8	356	2 800	3 156	712	50%
MCF_LWAL-DUL_H1	0.3	105	3 130	3 235	210	50%
MCF_LWAL-DUL_H2	0.7	332	2 900	3 232	664	50%
MUR_LWAL-DNO_H1	0.3	392	2 818	3 210	784	50%
MUR_LWAL-DNO_H2	0.6	654	2 548	3 202	1 308	50%
MUR_LWAL-DNO_H4	1.3	546	2 646	3 192	1 092	50%
Total		**3 052**	**19 348**	**22 400**	**6 104**	**50%**

[1]Maximum Displacement

[2] Each entry corresponds to a different seismic event (after the pre-processing of data).

4.6. Evaluation metrics

Regression tasks involve predicting continuous or numerical values. In this section, evaluation metrics commonly used to assess the performance of regression models are explored. Several researchers have made significant contributions in this area, and their work is referenced to provide a comprehensive understanding of the evaluation metrics for regression tasks. The most common evaluation metrics are:

- **Mean Squared Error (MSE)** is one of the most widely used evaluation metrics for regression tasks. It measures the average squared difference between the predicted values and the actual values. The lower the MSE, the better the model's performance. MSE is calculated by taking the average of the squared residuals.

- **Root Mean Squared Error (RMSE)** is a variant of MSE that takes the square root of the average squared difference between the predicted and actual values. RMSE is preferred over MSE when the objective is to interpret the error metric in the same units as the target variable.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2}$$

$\widehat{Y}_i$ = *predicted value*
Y_i = *actual value*
n = *number of observations*

- **Mean Absolute Error (MAE)** is another commonly used metric for regression evaluation. It calculates the average absolute difference between the predicted values and the actual values. Therefore, MAE provides a measure of the average magnitude of the errors. Lower values of MAE indicate better model performance.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{Y}_i - Y_i\right|$$

$\widehat{Y}_i$ = *predicted value*
Y_i = *actual value*
n = *number of observations*

- **R-squared (R^2)** is a metric that represents the proportion of variance in the dependent variable that is predictable from the independent variables. It measures the goodness-of-fit of the regression model. R^2 ranges from 0 to 1, with 1 indicating a perfect fit. However, R^2 should be interpreted cautiously as it can be influenced by the number of predictors and may not reflect the model's predictive performance on unseen data.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}(Y_i - \overline{Y}_i)^2}$$

$\widehat{Y}_i$ = *predicted value*

$\overline{Y}_i$ = *arithmetic mean of observed values*

Y_i = *actual value*

n = *number of observations*

- In addition to the aforementioned evaluation metrics for regression tasks, **the standard deviation (STD)** is a statistical measure that provides insights into the dispersion or variability of the predicted values around the mean. While not a specific evaluation metric, the STD can be useful in understanding the spread of the predictions and assessing the model's reliability. A higher standard deviation indicates a greater degree of variability in the predictions, suggesting potential uncertainty or inconsistency in the model's performance. The STD can be calculated by taking the square root of the variance, where the variance represents the average of the squared differences between each predicted value and the mean prediction.

$$SD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y}_i)^2}$$

$\overline{Y}_i$ = *arithmetic mean of observed values*

Y_i = *actual value*

n = *number of observations*

Several researchers have contributed to the development and analysis of these evaluation metrics for regression tasks. For instance, Hastie et al. (2021) discuss the importance of MSE, MAE, and standard deviation (SD) as key metrics in evaluating regression models. They emphasize the interpretation of these metrics in the context of the specific problem domain. In another study, Hastie et al. (2009) introduce the concept of RMSE and its significance in measuring prediction accuracy in regression analysis. They highlight the usefulness of RMSE when the magnitude of errors is important in the evaluation process. Furthermore, Draper & Smith (1998) provide a detailed discussion on R^2 and its limitations as an evaluation metric. They emphasize the need to consider additional metrics alongside R^2 to obtain a comprehensive understanding of the model's performance.

To summarize, the tasks performed throughout the training of the aforementioned algorithms encompassed data cleaning (i.e., removing blank, duplicate, and null values) and transformation (by applying the *min-max normalization*, and logarithmically transformation to avoid the impact of outliers). This process was followed by the creation of various datasets (S-19V, S-10V and S-10V-NL/L) through variable reduction and data adjustment, with the latter dataset focusing on ensuring fairness between linear and nonlinear sets. An essential part of training relates to the definition of the split between training and testing data (80/20), as well as the definition and appliance of the hyperparameter optimization technique (Grid Search Cross-Validation). Furthermore, the metrics chosen for evaluating model performance were explicitly stated, offering an objective foundation for critical analysis of the results obtained.

5. Results

Following the completion of hyperparameter optimization, the nine algorithms mentioned earlier were trained, and then used to make predictions. To assess the performance of the models, a range of error metrics, including RMSE, MAE, SD, and R^2 were computed. Moreover, to facilitate visualization and provide a comprehensive overview, although all algorithms were trained for each building class separately, each algorithm valuation metric is the result of the average metric across the seven building classes. This averaging process allows for a consolidated assessment of the models' performance across different building classes. By considering the aggregated results, any class-specific variations in performance can be accounted for, providing a more generalized evaluation.

The evaluation of the models using the aforementioned error metrics enables a comprehensive comparison of their predictive capabilities. Lower values of RMSE and MAE indicate a better fit of the model to the data, with smaller deviations between predicted and actual values. A lower SD suggests greater consistency and reliability in the model's predictions. The R^2 value provides an indication of how well the model explains the variance in the data, with higher values indicating a better fit. Considering these error metrics and their average values across the seven building class datasets, it is possible to draw conclusions regarding the overall performance of the nine algorithms.

5.1. Dataset with 19 input variables (S-19V)

The results obtained from the 19 input variables reveal valuable insights into the performance of various algorithms in relation to evaluation measures. The findings indicate that the ANN algorithm consistently outperformed the other algorithms across all evaluation measures (Table 7, Figure 5 And Figure 6) This suggests that ANN is particularly effective in addressing the complexities associated with nonlinear problems, such as those encountered in the study.

Additionally, the regularization techniques, Lasso and Ridge, commonly employed in linear regression problems, exhibited comparatively lower performance in the study. This can be

attributed to their inherent limitations when dealing with complex nonlinear problems. Specifically when buildings experience damage, their behaviour deviates from linearity, rendering Lasso and Ridge to be less effective in such scenarios. Hwang et al. (2021) highlighted the same issues in their research, where they also concluded that the R^2 of the linear-relationship models were smaller compared to other models such as decision tree, random forest, and boosting methods, indicating a nonlinear relationship between the input and output variables.

Following ANN, three algorithms emerged as notable contenders with closely matched results: RFR, GBR, and XGBR (by this particular order). These algorithms demonstrated strong performance, albeit slightly below that of ANN. The remaining algorithms in the study also exhibited satisfactory results, although their performance was comparatively lower than the top four algorithms.

Table 7. A Summary of results from the S-19V sample

Algorithms	**Evaluation metrics**				
	RMSE	**MAE**	**STD**	**R^2**	**Training Time (s)**
RR	0.504	0.396	0.504	0.785	0.008
LR	0.504	0.396	0.505	0.785	0.006
SVR	0.251	0.190	0.251	0.943	0.428
DTR	0.258	0.200	0.257	0.940	0.072
RFR	0.222	0.174	0.222	0.954	17.572
GBR	0.224	0.176	0.224	0.953	0.888
XGBR	0.227	0.177	0.227	0.952	0.915
ANN	0.221	0.172	0.221	0.955	42.05
MLP	0.256	0.200	0.256	0.942	8.275

RR - Ridge Regressor; **LR** - Lasso Regressor; **SVR** - Support Vector Regressor; **DTR** - Decision Tree Regressor; **RFR** - Random Forest Regressor; **GBR** - Gradient Boost Regressor; **XGBR** - Extreme Gradient Boosting Regressor; **ANN** - Artificial Neural Networks; **MLP** - Multi-Layer Perceptron.

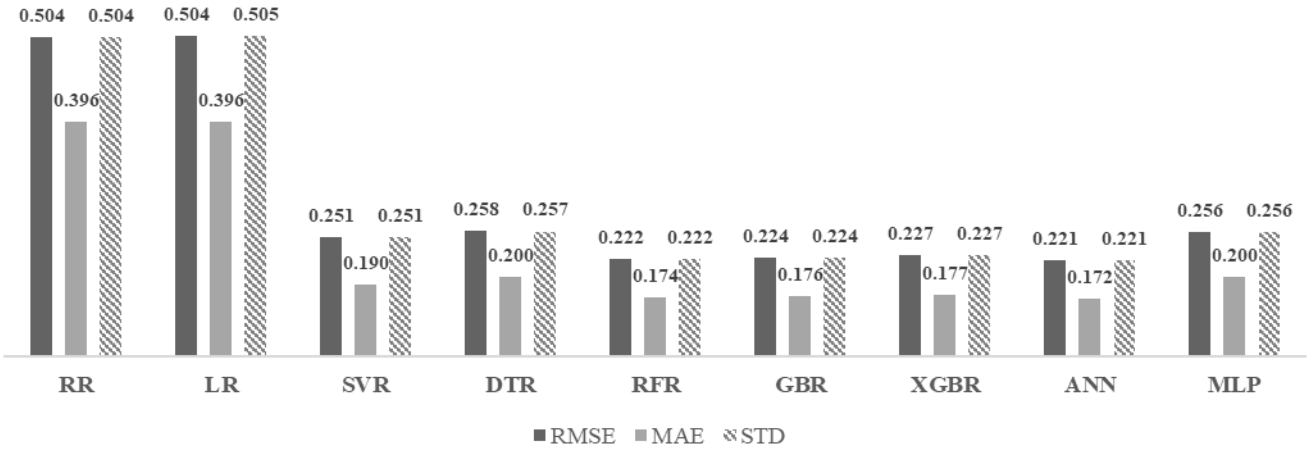

Figure 5. Evaluation metrics from the S-19V sample – RSME, MAE and STD

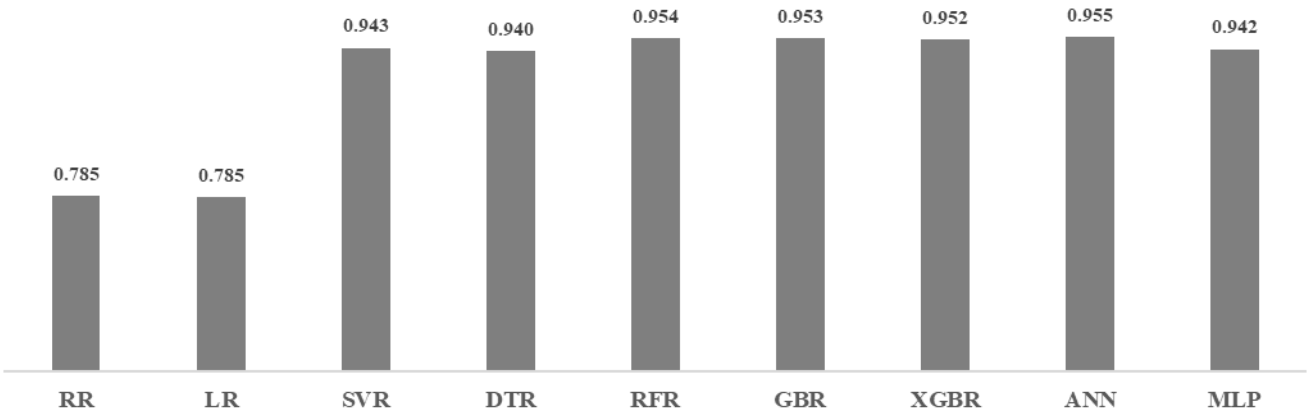

Figure 6. Evaluation metrics from the S-19V sample – R^2

Considering the practical implications of the research, it is worth noting that collecting data on all 19 variables used in the study may be a challenge in real-world scenarios due to time constraints (for instance, ANN and RFR are top performers; however, they require more than 10x and 2x the training time of others, respectively). Consequently, a subset of 10 variables was identified and utilized in the analysis. This reduction in the number of variables aims to streamline data collection processes and facilitate the implementation of the studied algorithms in practical applications.

5.2. Dataset with 10 input variables (S-10V)

Upon comparing the results obtained with the 10 variable (S-10V) subset to those achieved with the previous 19 variable (S-19V) dataset, it can be concluded that despite the significant reduction in variables, the performance remains commendable. This suggests that utilizing only 10 variables is a feasible approach that reduces the data collection requirements for making accurate forecasts. Consistent with the previous analysis, the evaluation of metrics reveals that RR and LR continue to underperform. This finding reinforces the understanding that the problem at hand is not strictly linear, emphasizing the limitations of these regularization techniques in nonlinear scenarios. Conversely, the ANN algorithm once again demonstrates its superiority, outperforming all other algorithms across all evaluation metrics, a result yet again very much expected. Following ANN, the RFR, GBR, and XGBR (by this particular order) consistently exhibit strong performance. These algorithms demonstrate their potential as reliable alternatives, offering good results in various prediction tasks. It is worth noting that the overall performance of the remaining algorithms remains satisfactory, and still slightly lower in comparison to the top-performing algorithms. More details on the metrics are provided in Table 8, Figure 7 and Figure 8.

Table 8. Summary of results from the S-10V sample

Algorithms	Evaluation metrics				
	RMSE	**MAE**	**STD**	**R^2**	**Training Time (s)**
RR	0.505	0.398	0.505	0.784	0.007
LR	0.506	0.398	0.507	0.783	0.004
SVR	0.260	0.198	0.260	0.939	0.355
DTR	0.270	0.210	0.270	0.933	0.056
RFR	0.234	0.182	0.233	0.949	10.292
GBR	0.235	0.184	0.235	0.948	0.675
XGBR	0.237	0.185	0.237	0.948	0.511
ANN	0.231	0.182	0.231	0.950	35.689
MLP	0.253	0.196	0.253	0.943	6.382

RR - Ridge Regressor; **LR** - Lasso Regressor; **SVR** - Support Vector Regressor; **DTR** - Decision Tree Regressor; **RFR** - Random Forest Regressor; **GBR** - Gradient Boost Regressor; **XGBR** - Extreme Gradient Boosting Regressor; **ANN** - Artificial Neural Networks; **MLP** - Multi-Layer Perceptron.

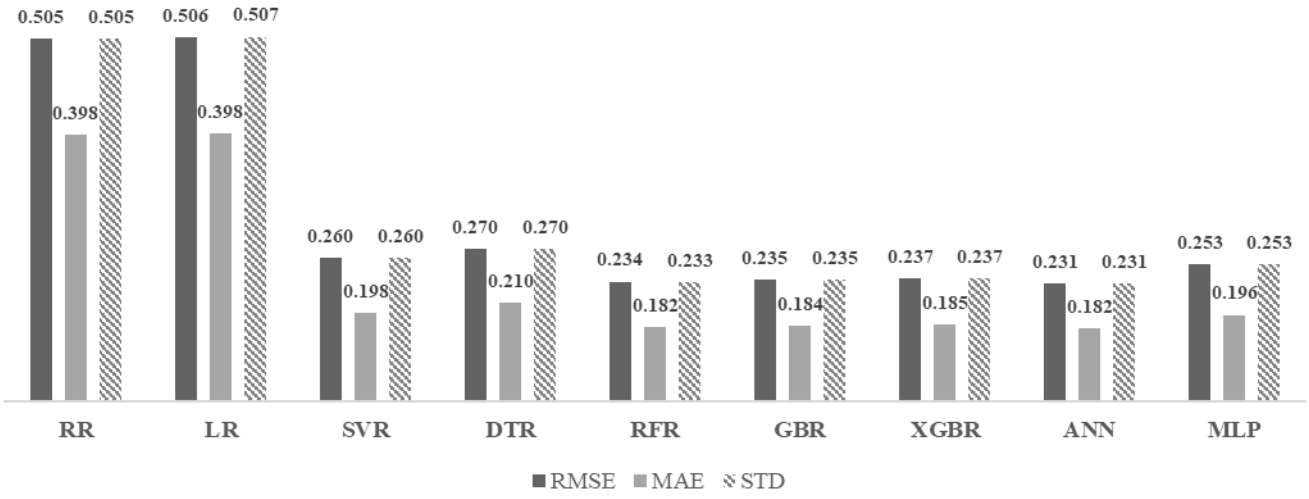

Figure 7. Evaluation metrics from the S-10V sample – RSME, MAE and STD

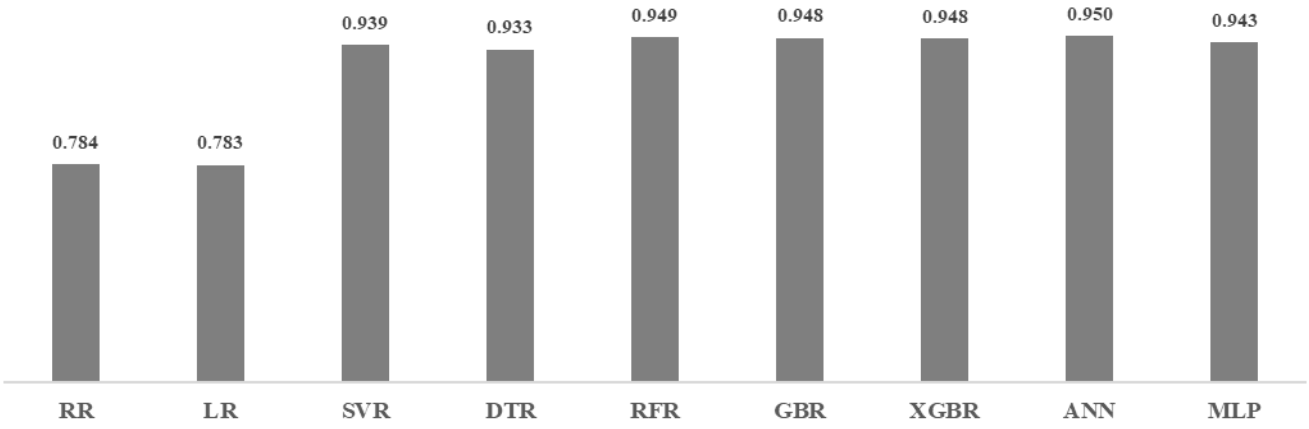

Figure 8. Evaluation metrics from the S-10V sample – R^2

Additionally, as anticipated, ANN and RFR remained the most time-consuming algorithms to train, having decreased their time by 15% and 41%, respectively. Moreover, the average training time of all algorithms reduced by 23% compared to the previous sample, and when excluding algorithms like ANN and RFR, it decreased by approximately 25%.

5.3. Dataset with 10 variables corrected for linearity (S-10V-NL/L)

In this analysis the dataset with 10 variables was used with a 50% of linear data and 50% of nonlinear data, as explained in chapter 4.2. With this last assessment was evident that the ANN algorithm consistently outperforms the other algorithms, across all evaluation measures. This reaffirms the effectiveness of ANN in handling complex nonlinear problems, as observed in the previous analyses. Following ANN, XGBR and RFR demonstrated strong performance throughout the evaluation metrics, with XGBR now leaving the 4th best to 2nd best algorithm. The GBR also exhibits competitive results, albeit slightly lower than XGBR and RF. These findings are consistent with the previous analysis, where these algorithms consistently emerged as notable contenders with strong performance. On the other hand, LR and RR continue to show comparatively lower performance. It's worth noting that the evaluation metrics of the MLP, followed by SVR, and then DTR also show satisfactory results, although not as strong as the top-performing algorithms. Overall, the findings from Table 9 are in line the previous analysis.

Table 9. Summary of results from the S-10V-NL/L sample

Algorithms	Evaluation metrics				
	RMSE	**MAE**	**STD**	**R^2**	**Training Time (s)**
RR	0.591	0.462	0.589	0.795	0.003
LR	0.593	0.463	0.590	0.792	0.001
SVR	0.291	0.218	0.291	0.951	0.042
DTR	0.333	0.251	0.333	0.937	0.010
RFR	0.262	0.201	0.263	0.961	2.605
GBR	0.267	0.207	0.268	0.960	0.341
XGBR	0.259	0.202	0.259	0.962	0.251
ANN	0.243	0.188	0.244	0.966	13.98
MLP	0.283	0.224	0.281	0.955	1.733

RR - Ridge Regressor; **LR** - Lasso Regressor; **SVR** - Support Vector Regressor; **DTR** - Decision Tree Regressor; **RFR** - Random Forest Regressor; **GBR** - Gradient Boost Regressor; **XGBR** - Extreme Gradient Boosting Regressor; **ANN** - Artificial Neural Networks; **MLP** - Multi-Layer Perceptron.

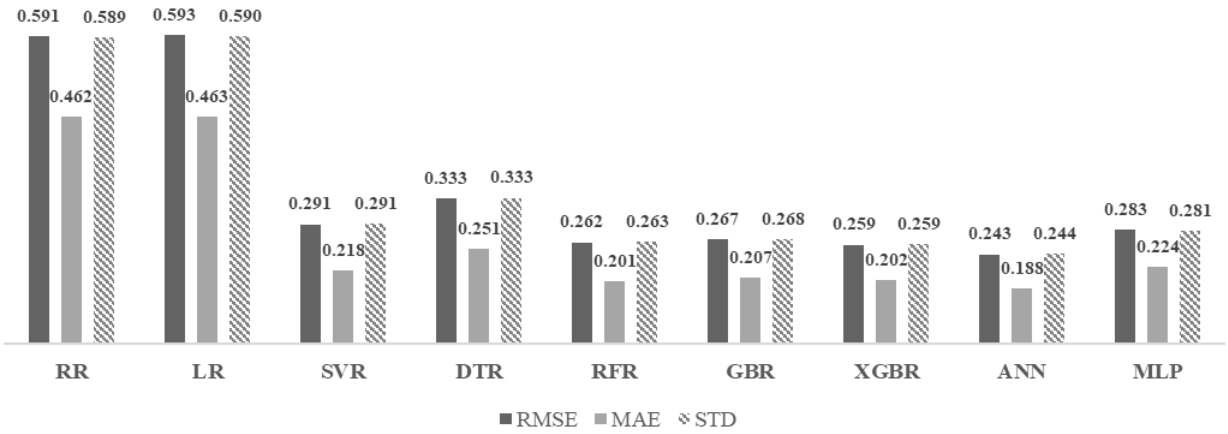

Figure 9. Evaluation metrics from the S-10V-NL/L sample – RSME, MAE and STD

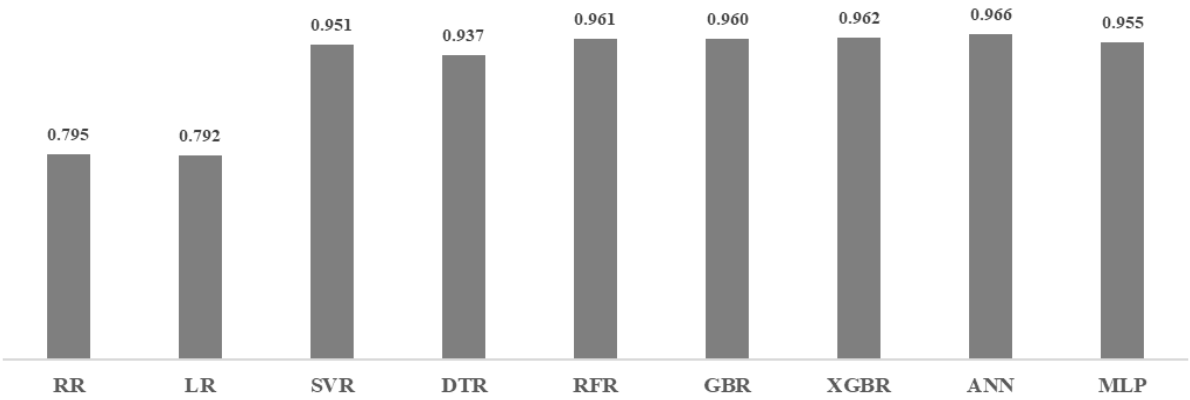

Figure 10. Evaluation metrics from the S-10V-NL/L sample – R^2

Furthermore, the conclusions regarding time consumption for training the algorithms remains the same, with ANN being overall the most time-consuming. Nonetheless, since there was a decrease of 73% of entry data (Table 6 - total entries are 22,400 and balanced entries are 6,104), it led to an average decrease of training time of 65%.

In summary, the initial dataset contained 29 variables. Through a process of trial-and-error the number of variables in the dataset was progressively reduced until there were 19 variables left (S-19V), with minimal impact on the performance of results as shown in Table 8. Subsequently, knowing that 19 variables still posed practical challenges for real-world application, the dataset was further reduced in half through iterative experimentation, ultimately arriving at 10 variables (S-10V). Remarkably, this reduction had relatively small impact (although it was a negative one of 0.37%) on the final model performance, considering the significant reduction in the number of variables (Table 10).

To correct potential bias in the data, an analysis was conducted where the dataset was composed into 50% of linear and 50% of nonlinear segments (S-10V- NL/L). Importantly, it is worth highlighting that despite the efforts to balance the data between linear and nonlinear segments, it was only achieved slightly improved results compared to the previous dataset configurations. This may potentially suggest that the initial dataset did not exhibit substantial bias.

Overall, the process of variable reduction and data validation underscores the rigor and robustness of the methodology used, ensuring that the dataset remains representative and suitable for rigorous scientific analysis. In model building, it becomes clear that models should not only focus on performance but also in practicality. Ideally, they should be combined with efficiency and usefulness in mind, so that the solutions created are not only accurate but also easily applied in real scenarios. For instance, the study conducted by Xu et al. (2020), where a staggering 48 input variables were employed in their model. While this approach may yield high accuracy, it simultaneously raises practical concerns regarding real-world applicability. Thus, although collecting a significant number of variables is not an insurmountable task, it may present challenges in collecting all the required data promptly. Moreover, as the volume of data to be gathered increases, it commonly necessitates additional resource allocation, hence making the process less efficient. Additionally, it's essential to recognize that the applicability of a model with an abundance of variables can be constrained in less developed regions lacking comprehensive earthquake engineering resources.

Concerning the most effective algorithms, the ANN algorithm consistently outperforms other algorithms in terms of predictive capabilities. The GBR, XGBR, and RFR algorithms also

demonstrate strong performance and can be considered reliable alternatives. Meanwhile, the LR and RR exhibit comparatively lower performance, reinforcing their limitations in handling complex nonlinear problems. MLP, SVR and DTR also show satisfactory results but are slightly outperformed by the top algorithms. As stated in chapter 2., although MLP is also an ANN algorithm, as seen in Table 10, it is placed between the 6th and 5th best algorithm across the different samples, once again proving Aurélien Géron (2022) point that while the Scikit-Learn library is more convenient and faster to train, its limited neural net features often lead users to prefer the more versatile Keras library for building artificial neural networks.

In terms of training time, as visible in Table 9, the ANN algorithm requires the longest training time compared to other algorithms (more than 13 times the training duration of others). This can be attributed to the inherent complexity and computational requirements of training neural networks. However, the superior performance of ANN justifies the longer training time, as it provides more accurate predictions.

Table 10. Summary of results across all samples - R^2

Algorithms	**Perfomance Metric: R^2**		
	S-19V	**S-10V**	**S-10V-NL/L**
RR	0.785	0.784	0.795
LR	0.785	0.783	0.792
SVR	0.943	0.939	0.951
DTR	0.940	0.933	0.937
RFR	0.954	0.949	0.961
GBR	0.953	0.948	0.960
XGBR	0.952	0.948	0.962
ANN	0.955	0.950	0.966
MLP	0.942	0.943	0.955

RR - Ridge Regressor; **LR** - Lasso Regressor; **SVR** - Support Vector Regressor; **DTR** - Decision Tree Regressor; **RFR** - Random Forest Regressor; **GBR** - Gradient Boost Regressor; **XGBR** - Extreme Gradient Boosting Regressor; **ANN** - Artificial Neural Networks; **MLP** - Multi-Layer Perceptron.

5.4. Comparison of Results with the Current State of the Art

Apart from the aforementioned analysis it is important to compare the previous results with other scientific work, namely the ones discussed in chapter 3. A table summarizing the results based only on R^2 is presented below.

Table 11. Comparative Evaluation Metric Analysis against State of the Art – R^2

Algorithms	This work			State of the Art			
	Perfomance Metric: R^2						
	S-19V	S-10V	S-10V-NL/L	Hwang et al. (2021)	Kalakonas & Silva (2022)	Demertzis et al. (2023)	Alcantara & Saito (2023)
RR	0.785	0.784	0.795	0.536[(a)] / 0.455[(b)]		0.744	
LR	0.785	0.783	0.792			0.682	
SVR	0.943	0.939	0.951				
DTR	0.940	0.933	0.937	0.592[(a)] / 00.604[(b)]		0.742	0.857[(d)] / 0.776[(e)]
RFR	0.954	0.949	0.961	0.805[(a)] / 0.723[(b)]		0.719	0.867[(d)] / 0.893[(e)]
GBR	0.953	0.948	0.960			0.754 / **0.862**[(c)]	**0.870[(d)] / 0.902[(e)]**
XGBR	0.952	0.948	0.962	**0.861[(a)] / 0.829[(b)]**			0.818[(d)] / 0.862[(e)]
ANN	**0.955**	**0.950**	**0.966**		**0.966**		
MLP	0.942	0.943	0.955				0.820[(d)] / 0.881[(e)]

RR- Ridge Regressor; LR - Lasso Regressor; SVR - Support Vector Regressor; DTR - Decision Tree Regressor; RFR - Random Forest Regressor; GBR - Gradient Boost Regressor; XGBR - Extreme Gradient Boosting Regressor; ANN - Artificial Neural Networks; MLP - Multi-Layer Perceptron.

(a) four-story case; (b) eight-story case; (c) Light Gradient Boost algorithm; (d) floor case; (e) floor and roof case.

Note: The outcomes presented in bold font indicate the superior algorithm.

First and foremost, when it comes to assessing structural damage in buildings post-earthquake, it is crucial to emphasize the importance of achieving an R^2 value exceeding 90% (0.9) in the predictive models, since human lives are at stake, and in the worst-case scenario, undermine something as bad as human life losses.

Another important note to be made before moving forward, is that the algorithms' results presented in this study are not directly comparable to the ones presented in the state of the art due to the utilization of different datasets, dataset treatment, chosen algorithm idiosyncrasies, different types of input variables and how it was collected, as well as the fact that algorithms may be predicting different aspects or outcomes (different output variable). Any comparison between the algorithms should be made with caution, considering these differences in data and prediction objectives.

With the above mentioned in mind, it's worth noting that Hwang et al. (2021) made predictions for maximum story drift, and the top-performing algorithm was XGBR. Their results indicated an R^2 value of 0.861 for the four-story case and 0.829 for the eight-story case. If we consider the results of the other algorithms used also in this study, namely the RFR, DTR, and RR, the performance is inferior and as previously discussed, it is important to emphasize that having an R^2 value above 0.9 would be highly recommended. It's notable that when comparing these results with the findings from this dissertation, our samples and model achieved superior performance, even with a smaller set of variables (Table 11).

In the study of Kalakonas & Silva (2022), ANN was employed to predict the maximum displacement (same as in this work) utilizing a dataset comprising 14 variables. Notably, the study achieved an impressive average (26 building classes) R^2 value of 0.966, which is nearly equivalent to the performance observed in S-10V-NLL/L. It is noteworthy to mention that in this study sample, specifically S-10V and S-10V-NL/L, a more constrained set of variables comprising only 10 factors, was used, and that despite this reduction in the number of variables (greater practicality), both samples exhibited noteworthy predictive capabilities. Though it is important to recognize that they may not be directly comparable to the study conducted by Kalakonas & Silva (2022), these results are an interesting indicator that “less can be more”. Since the initial data set in this study comes from the work of Kalakonas & Silva (2022), this discrepancy arises from the fact that different building classes were considered, potentially introducing variations in the predictive outcomes, among other factors such as hyperparameterization techniques (Grid Search Cross-Validation vs. hand tuning). Nonetheless, these results highlight the efficacy of parsimonious feature selection and model generalization strategies within the realm of building displacement prediction.

Demertzis et al. (2023) also analysed the same problem though a lens of comparison of various algorithms. In their study, with a set of 18 variables, the maximum inter-story drift ratio was the variable to be predicted from a sample of 30 reinforced concrete buildings (three hypotheses regarding the distribution of masonry infills were examined for each of the 30 buildings, i.e., three different subsets: (i) 30 buildings lacking masonry infills, (ii) 30 buildings with masonry infills evenly distributed along their height, and (iii) 30 buildings featuring an absence of infills in the first story and infills in the upper stories). The conclusions were that the Light GBR proved to be the best, achieving an R^2 performance of 0.862 (average of the three subsets) or, at best 0.907 for only the subset of buildings lacking masonry infills, which in comparison is again slightly below this study results with only 10 variables. Even if we don't include ANN, and only compared to the other models in common such as RFR, GBR, RR, LR, and DTR that are not black box algorithms, this dissertation results still have better results in comparison (Table 11).

Alcantara & Saito (2023) collected data using sensors on the ground and on the roof of the buildings, dividing the research in 2 case-studies: the first case only using the ground sensor data, and in the second using ground and roof sensor data. In the first case, the best algorithm for predicting maximum story drift (output variable) was GBR with an R^2 of 0.870. In the second case the best algorithm was again GBR with an R^2 of 0.902, which slightly in line with this research results, as GBR also makes the top performing algorithms. Also, given that Alcantara & Saito (2023) does not use ANN, but instead they use MLP, and MLP is the algorithm exhibiting the poorest performance, this again comes to show that MLP might not be the suitable alternative for ANN, for reasons already explained. Overall, the results of all the same algorithms in different studies used are below the results of this study, even with a reduced number of variables (27 variables in Alcantara & Saito (2023) vs., 10 variables in this work).

The better performance of the algorithms used in this dissertation in comparison to those employed by other researchers can be attributed to a series of steps taken in this study. First and foremost, the quality of the dataset can deviate significantly due to the focus on ensuring that it was meticulously curated and devoid of anomalies. Furthermore, the implementation of a comprehensive data preprocessing pipeline, which, in this study, included the meticulous application of hyperparameterization techniques and data transformations. This approach not only addressed data linearity but also leveraged the use of logarithmic transformations to

enhance the dataset's suitability for modelling, something not as evident in the work described in the state of the art. The selection of the input and output variables can also be a critical differentiator aspect, where choosing the most relevant predictors but also having the best calculation methods for certain variables, may have resulted in a more accurate representation of the underlying phenomena. Lastly, the consideration of different building types allows tailoring the models to specific scenarios that may have been overlooked by the previous studies. Our comprehensive approach, including data quality enhancement, meticulous data processing, variable selection, and nuanced building type considerations, may also have led to the better results of the algorithms than the ones used by other researchers.

6. Conclusion & Further Research

Earthquakes are an extremely important natural phenomenon that must be studied as their impact can be devastating, physically as well as economically and socially. Therefore, this research project was motivated by a profound interest in contributing to the critical field of earthquake engineering, due to the paramount importance of promptly detecting building damage after seismic events, showcasing the potential of Machine Learning (ML) algorithms for this purpose. By harnessing ML's adaptability and efficiency, this work sought to improve the state of the art in earthquake damage prediction, with focus on algorithm comparison. This research underscores the vital need for innovative approaches to mitigate post-earthquake catastrophes and enhance public safety.

The comprehensive examination of the state of the art revealed a growing interest in detecting post-earthquake building damage using ML. While this research field has seen recent development, there remains a notable scarcity of literature comprehensively addressing the intersection of ML and earthquake engineering. Notably, even though the state of the art highlighted the widespread use of ANN, it is fair to say that other algorithms, particularly Gradient Boost and its variants, are gaining increasing popularity due to their strong performance and simplicity when compared to ANN methodologies.

In terms of data set used, this study takes into account the heterogeneity of buildings by exploring seven diverse building classes representing the most common styles seen in the Balkan region, determined by key structural attributes like construction material, lateral load resistance, story count, and seismic code. These structures were simulated by employing numerical modelling derived from Martins & Silva (2021) and used in Kalakonas & Silva (2022). Moreover, the initial dataset included 29 input variables, but to avoid the dimensional curse and overfitting, it was downsized to 19 variables by concentrating on specific variable ranges. Eventually, this dataset was further reduced to just 10 input variables. The output variable considered is the Maximum Displacement, representing the peak movement of a building component during seismic events, conventionally indicating damage beyond a specific threshold.

Additionally, to address the data imbalance issue, a 50/50 data split between linear (non-damage) and nonlinear (damage) cases was adopted and applied to the dataset of 10 input variables, aiming to mitigate biases and improve performance in machine learning algorithms. To ensure data quality, it was latter performed meticulous data cleansing and transformation (normalization and logarithmic transformation were applied to ensure consistent scales and stabilize variable variance). As necessary in ML, the data was split into training and testing sets using an 80/20 split ratio and Grid search cross-validation was employed for hyperparameter optimization to enhance model performance.

One of main conclusion that could be drawn from this research is that the gradual reduction of variables within the dataset, progressing from an initial 29 variables to a final set of 19 (S-19V), and ultimately consolidating to 10 variables (S-10V), yielded only a modest impact of -0.37% on the model's overall performance. This observation underscores the balance between computational complexity and practical applicability, showing that utilizing a subset of fewer variables (a more pragmatic approach) is a feasible approach that still yields acceptable performance. This reduction in variables can simplify data collection processes and facilitate practical implementation of the studied algorithms. Also, an attempt to balance linear and nonlinear segments resulted in modest performance improvement (S-10V- NL/L) of +1.23%, suggesting that, although the initial dataset had bias, it was not highly significant.

Regarding algorithm performance, it was possible to conclude from all samples used in this study that the ANN algorithm consistently outperformed others, followed closely by GBR, XGBR, and RFR. And that LR and RR exhibited lower performance, indicating challenges in handling complex, nonlinear problems. MLP, SVR, and DTR also showed satisfactory results but were slightly outperformed by the top algorithms. Based on the MLP results compared to other algorithms, it was also reaffirmed that it is preferred to use ANN, despite the MLP advantages in convenience and training time.

In line with the last sentence, despite its longer training time, the superior performance of ANN justifies the computational investment, making it the preferred choice for accurate predictions.

During this research work several significant points have emerged, opening doors to potential areas of further investigation and exploration in earthquake engineering.

As indicated in the state of the art section, particularly in the work by Xie et al. (2020), the domain of earthquake engineering presents multiple avenues for further exploration. These encompass seismic hazard analysis, which predicts ground shaking and associated uncertainties to inform risk assessment and building design. The realm of system identification and damage detection spans a broad spectrum, from classifying component failures in controlled experiments to detecting extensive structural damage on a larger scale. Seismic fragility assessment, involving the estimation of structural failure probabilities under seismic conditions, is integral to implement effective risk mitigation strategies. Additionally, structural control methods for earthquake mitigation employ active and semi-active techniques to reduce vibrations induced by earthquakes. Therefore, the selection of appropriate algorithms may vary depending on the specific problem, emphasizing the continued need for research in diverse facets of this field.

As stated, there are three main problems associated with the usage of ML, (i) the availability of high-quality training data, (ii) the complexity of the relationships between the factors involved, (iii) the potential for uncertainty and error. Regarding the first point, future research should prioritize increasing the diversity and quality of available data. Meaning, the incorporation of a broader spectrum of building or component classes (taking into consideration the heterogeneity of types of buildings), and the quest for high-quality data across different geographic zones should also be pursued, potentially utilizing complementary technologies like image recognition for data augmentation.

Additionally, there is still a lack of more studies comparing algorithms, similar to this work and Demertzis et al. (2023), as well as exploring lesser known algorithms and the impact of hyperparameterization. Moreover, experimentation with alternative variables and variable calculation methods, particularly those involving new technologies such as sensors, can offer valuable insights into improving predictive models.

Furthermore, as highlighted in Kalakonas & Silva (2022), this study, along with many existing ones in the field, grapples with inherent limitation when it comes to the output variable (maximum displacement) assessment. These limitations stem from the use of single-

degree-of-freedom (SDOF) oscillators for vulnerability function development, that comprehends the SDOF oscillators' inability to comprehensively address other impact factors such as higher modes of vibration and torsional effects (and not only the simplification of the reality with the only back-and-forth movement of the structure). Although much more complex, Multi degree-of-freedom (MDOF) systems offer a means to address these limitations, thus emphasizing the potential for future research in this domain.

Finally, although the main conclusion in this study is the acknowledgement of ANN better performance compared to other algorithms for seismic damage assessment, it is imperative to recognize the associated trade-off between the considerably longer training times. The decision to employ ANN or other algorithms ultimately hinges upon the experts assessments and research of whether time constraints outweigh the model's efficiency.

7. Bibliography

Alcantara, E. A. M., & Saito, T. (2023). Machine Learning-Based Rapid Post-Earthquake Damage Detection of RC Resisting-Moment Frame Buildings. *Sensors*, *23*(10). https://doi.org/10.3390/s23104694

Aurélien Géron. (2022). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems. In *O'Reilly Media* (3rd Editio).

Calvi, G. M., Pinho, R., Magenes, G., Bommer, J. J., Restrepo-Vélez, L. F., & Crowley, H. (2006). Development of seismic vulnerability assessment methodologies over the past 30 years. *ISET Journal of Earthquake Technology*, *43*(3).

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*. https://doi.org/10.1145/2939672.2939785

Curran-Everett, D. (2018). Explorations in statistics: The log transformation. *Advances in Physiology Education*, *42*(2). https://doi.org/10.1152/ADVAN.00018.2018

De Stefano, A., Sabia, D., & Sabia, L. (1999). Probabilistic neural networks for seismic damage mechanisms prediction. *Earthquake Engineering and Structural Dynamics*, *28*(7–8). https://doi.org/10.1002/(sici)1096-9845(199908)28:8<807::aid-eqe838>3.0.co;2-#

Demertzis, K., Kostinakis, K., Morfidis, K., & Iliadis, L. (2023). An interpretable machine learning method for the prediction of R/C buildings' seismic response. *Journal of Building Engineering*, *63*, 105493. https://doi.org/10.1016/J.JOBE.2022.105493

Draper, N. R., & Smith, H. (1998). Applied Regression Analysis, 3rd Edition. In *John Wiley & Sons, Inc.*

Goh, A. T. C. (1994). Seismic liquefaction potential assessed by neural networks. *Journal of Geotechnical Engineering*, *120*(9). https://doi.org/10.1061/(ASCE)0733-

9410(1994)120:9(1467)

Harirchian, E., Aghakouchaki Hosseini, S. E., Jadhav, K., Kumari, V., Rasulzade, S., Işık, E., Wasif, M., & Lahmer, T. (2021). A review on application of soft computing techniques for the rapid visual safety evaluation and damage classification of existing buildings. In *Journal of Building Engineering* (Vol. 43). https://doi.org/10.1016/j.jobe.2021.102536

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Springer series in statistics: The elements of statistical learning: Data mining, inference and prediction. In *EAS Publications Series*.

Hastie, T., Tibshirani, R., James, G., & Witten, D. (2021). An introduction to Statistical Learning with Applications in R (2nd Edition). *Springer Texts*, *102*.

Hwang, S. H., Mangalathu, S., Shin, J., & Jeon, J. S. (2021). Machine learning-based approaches for seismic demand and collapse of ductile reinforced concrete building frames. *Journal of Building Engineering*, *34*, 101905. https://doi.org/10.1016/J.JOBE.2020.101905

Kalakonas, P., & Silva, V. (2022). Seismic vulnerability modelling of building portfolios using artificial neural networks. *Earthquake Engineering and Structural Dynamics*, *51*(2). https://doi.org/10.1002/eqe.3567

Mangalathu, S., Sun, H., Nweke, C. C., Yi, Z., & Burton, H. V. (2020). Classifying earthquake damage to buildings using machine learning. *Earthquake Spectra*, *36*(1). https://doi.org/10.1177/8755293019878137

Martins, L., & Silva, V. (2021). Development of a fragility and vulnerability model for global seismic risk analyses. *Bulletin of Earthquake Engineering*, *19*(15). https://doi.org/10.1007/s10518-020-00885-1

Nakamura, M., Masri, S. F., Chassiakos, A. G., & Caughey, T. K. (1998). A method for non-parametric damage detection through the use of neural networks. *Earthquake Engineering and Structural Dynamics*, *27*(9). https://doi.org/10.1002/(SICI)1096-9845(199809)27:9<997::AID-EQE771>3.0.CO;2-7

Shalabi, L. Al, Shaaban, Z., & Kasasbeh, B. (2006). Data Mining: A Preprocessing Engine. *Journal of Computer Science*, *2*(9). https://doi.org/10.3844/jcssp.2006.735.739

Xie, Y., Ebad Sichani, M., Padgett, J. E., & DesRoches, R. (2020). The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra*, *36*(4). https://doi.org/10.1177/8755293020919419

Xu, Y., Lu, X., Tian, Y., & Huang, Y. (2020). Real-Time Seismic Damage Prediction and Comparison of Various Ground Motion Intensity Measures Based on Machine Learning. *Journal of Earthquake Engineering*. https://doi.org/10.1080/13632469.2020.1826371

Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, *1168*(2). https://doi.org/10.1088/1742-6596/1168/2/022022

Yu, T., & Zhu, H. (2020). *Hyper-Parameter Optimization: A Review of Algorithms and Applications*. 1–56. http://arxiv.org/abs/2003.05689